

DIPLOMARBEIT

Gesamtprojekt
Windpark-Monitoring Simonsfeld

Windpark Icinga-Monitoring Basisfunktionen

Osman Mujkanovic 5BHEL

Betreuer: Dipl. -Ing. Gerald Stoll

Monitoring auf Basis von InfluxDB und Grafana

Stephan Schrenk 5BHEL

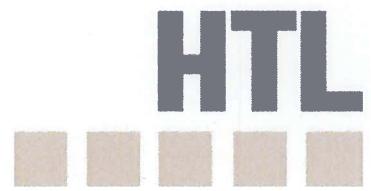
Betreuer: Dipl. -Ing. Gerald Stoll

ausgeführt im Schuljahr 2016/17

Abgabevermerk:

Datum: 04.04.2017

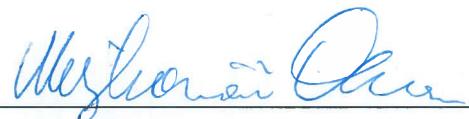
übernommen von:



Höhere Technische Bundeslehranstalt Hollabrunn
Höhere Lehranstalt für Elektronik und Technische Informatik

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe.

Osman Mujkanovic

A handwritten signature in blue ink that reads 'Osman Mujkanovic'.

Stephan Schrenk

A handwritten signature in blue ink that reads 'Stephan Schrenk'.

HINWEISE

Die vorliegende Diplomarbeit wurde in Zusammenarbeit mit der Firma Windkraft Simonsfeld AG ausgeführt.

Die in dieser Diplomarbeit entwickelten Prototypen und Software-Produkte dürfen ganz oder auch in Teilen von Privatpersonen oder Firmen nur dann in Verkehr gebracht werden, wenn sie diese selbst geprüft und für den vorgesehenen Verwendungszweck für geeignet befunden haben.

Es wird keinerlei Haftung übernommen für irgendwelche Schäden, die aus der Nutzung der hier entwickelten oder beschriebenen Bestandteile des Projekts resultieren.

Für alle Entwicklungen gilt die GNU General Public License

[<http://www.gnu.org/licenses/gpl.html>] der Free Software Foundation, Boston, USA in der Version 3.

Die Diplomarbeit erfüllt die "Standards für Ingenieur- und Technikerprojekte" entsprechend dem Rundschreiben Nr. 60 aus 1999 des BMBWK (GZ.17.600/101-II/2b/99). [https://www.bmb.gv.at/ministerium/rs/1999_60.html]

SCHLÜSSELBEGRIFFE

Network Monitoring, SNMP, Windparks, Icinga2, Grafana, Check_MK, Net-SNMP, EasySNMP, Python, Graphite, Icingaweb2, InfluxDB, Master, Satellite, Client, Server, Ubuntu, Windows, Ping, MIB, OID, Remote-Desktop, SSH

DANKSAGUNGEN

Wir bedanken uns sehr herzlich bei der Windkraft Simonsfeld AG für die Bereitstellung der Ressourcen und der Infrastruktur.

Ein weiterer Dank gilt insbesondere Herrn Michael Krell der Windkraft Simonsfeld AG für seine unterstützenden Tätigkeiten in Bezug auf vor Ort zu erledigende Tätigkeiten.

Zu guter Letzt möchten wir uns bei Herrn Dipl. -Ing. Gerald Stoll für die Betreuung unserer Diplomarbeit sehr herzlich bedanken.

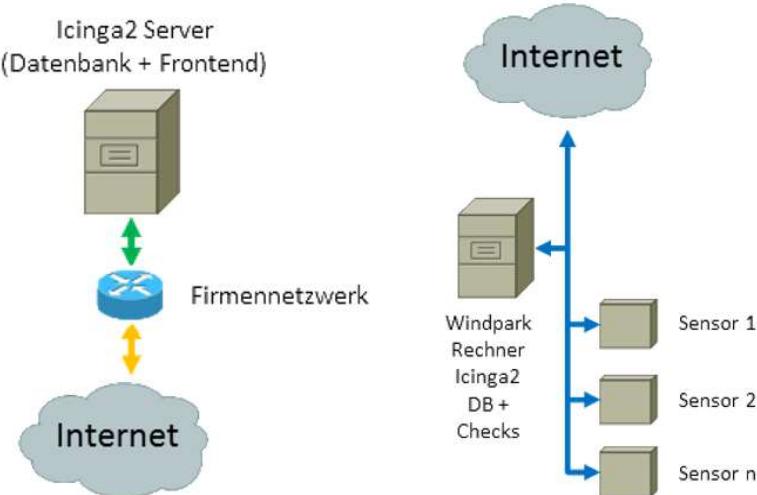
**DIPLOMARBEIT
DOKUMENTATION**

Namen der Verfasser/innen	Osman Mujkanovic, Stephan Schrenk
Jahrgang Schuljahr	5BHEL 2016/17
Thema der Diplomarbeit	Windpark-Monitoring Simonsfeld
Kooperationspartner	Windkraft Simonsfeld AG

Aufgabenstellung	Die Aufgabe der Diplomarbeit ist es auf Basis der Open-Source-Software Icinga2 ein System zur Überwachung der Windparks der Firma Windkraft Simonsfeld aufzubauen. Hierbei sollen die Windparks, welche aus Windows- und Linux-Servern sowie diversen Sensoren zur Messung unterschiedlichster Parameter bestehen, überwacht werden. Weiters soll der Aufbau einer Time-Based-Datenbank auf Basis von InfluxDB und die Anzeige der Daten mit Hilfe von Grafana getestet werden. Besonderes Augenmerk soll auf die Ausfallsicherheit des Systems gelegt werden, speziell bei der Verbindung zwischen den Servern.
------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Realisierung	Zu Beginn wurde die Software Icinga2 und Icingaweb2 lokal installiert um ein Gefühl für die zukünftige Arbeitsumgebung zu bekommen. Diese Tätigkeiten wurden durch das Buch über Icinga2 erläutert und anhand dieses Buchs durchgeführt. Weiters wurden InfluxDB und Grafana lokal installiert um einen Umstieg auf die Umgebung der Firma möglichst einfach zu gestalten. Diese Migration auf die reale Umgebung beinhaltet die Überwachung von zwei Rechnern, einem Temperatursensor, einer Network Management Card und einem Switch.
--------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Ergebnisse	Es wurde eine Überwachung der zur Verfügung gestellten Testumgebung realisiert. Hierbei handelt es sich um jeweils einen Windows- und einen Linux-Server, den Temperatursensor, den Switch und eine Netzwerkkarte. Um eine Abhängigkeit der Software Icinga2 zu verringern wurde mittels Python ein Script zur Überwachung dieser Sensoren geschrieben. Mit Grafana werden die Daten, welche vom Script in der Datenbank abgespeichert werden, grafisch dargestellt. Weiters wurden alternative Systeme zu den verwendeten analysiert und getestet.
------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Typische Grafik, Foto etc. (mit Erläuterung)	 <p>Das Diagramm zeigt die Architektur eines Monitorsystems. In der Mitte befindet sich ein blauer Kreis mit einem weißen Kreuz, der als Router oder Switch dient. Von diesem führen Pfeile zu zwei grauen Kästen: oben zu einem vertikalen Server-Kasten (Icinga2 Server) und unten zu einem vertikalen Kasten mit einem blauen Kreis (Windpark Rechner). Beide Kästen sind über einen horizontalen Pfeil miteinander verbunden. Von jedem Kasten führt ein vertikaler Pfeil zu einem blauen Kreis, der als 'Internet' markiert ist. Von jedem Internet-Kreis führt ein vertikaler Pfeil zu einem weiteren grauen Kasten, der als 'Sensor 1', 'Sensor 2' oder 'Sensor n' beschriftet ist. Ein großer blauer Pfeil führt von den Sensoren über das Internet zum Icinga2 Server.</p> <p>Der Rechner im Windpark sammelt die Daten der einzelnen Sensoren und sendet diese über das Internet an den „Icinga2 Server“, welcher der Hauptserver für die Verarbeitung der Daten ist. Diese Konstellation gewährleistet die Ausfallsicherheit des Systems, auch bei einer Unterbrechung der Internetverbindung, da der Server im Windpark die Daten auch lokal speichert und diese, sobald eine Verbindung verfügbar ist, an den Hauptserver sendet.</p>
-------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Teilnahme an Wettbewerben, Auszeichnungen	--
-------------------------------------------	----

Möglichkeiten der Einsichtnahme in die Arbeit	HTBL Hollabrunn Anton Ehrenfriedstraße 10 2020 Hollabrunn
-----------------------------------------------	-----------------------------------------------------------------

Approbation (Datum / Unterschrift)	Prüfer/Prüferin	Direktor/Direktorin Abteilungsvorstand/Abteilungsvorständin
---------------------------------------	-----------------	----------------------------------------------------------------

DIPLOMA THESIS

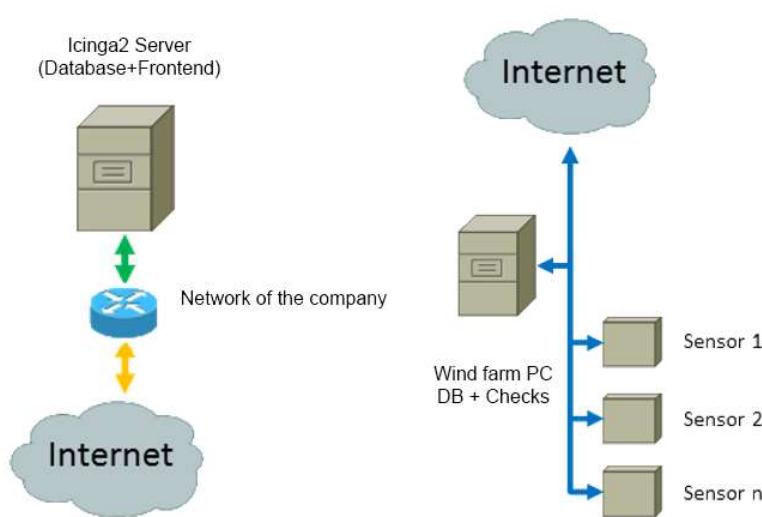
Documentation

Author(s)	Osman Mujkanovic, Stephan Schrenk
Form Academic year	5BHEL 2016/17
Topic	Windpark-Monitoring Simonsfeld
Co-operation Partners	Windkraft Simonsfeld AG

Assignment of Tasks	<p>The task of the diploma thesis is to build up a system for the monitoring of the wind farms of the company Windkraft Simonsfeld based on the open source software Icinga2. The wind park networks, which consist of Windows and Linux servers as well as various sensors for measuring various parameters, are to be monitored. In addition, the establishment of a time-based database based on InfluxDB and the display of the data should be tested using Grafana. Special attention should be paid to the failure safety of the system, especially when connecting the servers.</p>
---------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Realisation	<p>At the beginning the Icinga2 and Icingaweb2 software was installed locally to get a feeling for the future work environment. These activities were explained by the book about Icinga2 and carried out using this book. Furthermore, InfluxDB and Grafana were installed locally in order to make a change to the company's environment as easy as possible. This migration to the real environment involves the monitoring of two computers, a temperature sensor, a network management card and a switch.</p>
-------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Results	<p>Monitoring of the provided test environment was realized. These are each a Windows and a Linux server, the temperature sensor, the switch and a network card. To reduce the dependency of the Icinga2 software, a script for monitoring these sensors was written using Python. With Grafana, the data stored by the script in the database are displayed graphically. Furthermore, alternative systems were analysed and tested.</p>
---------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>Illustrative Graph, Photo (incl. explanation)</p>	 <p>The diagram shows a network architecture. At the top, a cloud labeled "Internet" is connected to a "Wind farm PC DB + Checks" (represented by a server icon). This computer is connected to a "Network of the company" (represented by a switch icon). The network then connects to an "Icinga2 Server (Database+Frontend)" (represented by a server icon). A double-headed arrow connects the "Wind farm PC" and the "Icinga2 Server". Below the "Wind farm PC", three sensors are shown: "Sensor 1", "Sensor 2", and "Sensor n", each represented by a small server icon. Arrows point from each sensor to the "Wind farm PC". The "Wind farm PC" has a blue arrow pointing upwards towards the "Internet".</p> <p>The computer in the wind park collects the data from the individual sensors and sends them over the Internet to the "Icinga2 Server", which is the main server for processing the data.</p> <p>This constellation ensures that the system is fail-safe, even when the Internet connection is interrupted because the server in the wind farm stores the data locally and sends it to the main server as soon as a connection is available.</p>
------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>Participation in Competitions Awards</p>	<p>--</p>
-------------------------------------------------	-----------

<p>Accessibility of Diploma Thesis</p>	<p>HTBL Hollabrunn Anton Ehrenfriedstraße 10 2020 Hollabrunn</p>
--------------------------------------------	--------------------------------------------------------------------------

<p>Approval (Date / Sign)</p>	<p>Examiner</p>	<p>Head of College / Department</p>
-----------------------------------	-----------------	-------------------------------------

Windpark-Monitoring Simonsfeld

Verlauf

- 23.09.2016 um 13:37 Die Themenstellung "Windpark-Monitoring Simonsfeld" (Osman Mujkanovic, Stephan Schrenk) wurde eingereicht.
 - 26.09.2016 um 13:22 Die Themenstellung "Windpark-Monitoring Simonsfeld" (Osman Mujkanovic) wurde vom Betreuer / von der Betreuerin akzeptiert.
 - 26.09.2016 um 18:57 Die Themenstellung "Windpark-Monitoring Simonsfeld" (Osman Mujkanovic) wurde vom zuständigen Abteilungsvorstand akzeptiert.
 - 28.09.2016 um 22:19 Die Themenstellung "Windpark-Monitoring Simonsfeld" (Osman Mujkanovic) wurde vom Direktor / von der Direktorin akzeptiert.
 - 05.10.2016 um 08:07 Die Themenstellung "Windpark-Monitoring Simonsfeld" (Osman Mujkanovic) wurde vom Landesschulinspektor / von der Landesschulinspektorin genehmigt.
-

Schule

Höhere technische Bundeslehranstalt HOLLABRUNN

Abteilung(en)

Hauptverantwortlich: Elektronik und Technische Informatik

AV

Hauptverantwortlich: Wilfried Trollmann

Abschließende Prüfung

2017

Betreuer/innen

Hauptverantwortlich: Gerald Stoll

Ausgangslage

Das WKS-Netzwerk besteht aus der Infrastruktur im Bürogebäude Ernstbrunn, sowie den Windpark-Netzwerken.

Ein typisches Windpark-Netzwerk besteht aus:

Mindestens 1 x Rechner (Windows 7 bzw. Ubuntu 16.04)

und mehreren Netzwerkkomponenten (Router, Switches, Temperatur-, Eissensoren, Webcams, usw...)

Projektteam (Arbeitsaufwand)

Name	Individuelle Themenstellung	Klasse	Arbeitsaufwand
Osman Mujkanovic (Hauptverantwortlich)	Windpark Icinga-Monitoring Basisfunktionen	5BHEL_17	180 Stunden
Stephan Schrenk	Monitoring auf Basis von InfluxDB und Grafana	5BHEL_17	180 Stunden

Projektpartner

Windkraft Simonsfeld (E-Mail: office@wksimonsfeld.at, <http://www.wksimonsfeld.at/>)
Simonsfeld 57a, 2115 Ernstbrunn, Michael Krell, 02576 3324

Untersuchungsanliegen der individuellen Themenstellungen

Die Aufgabe dieser Diplomarbeit ist die Ablöse des alten Monitoringsystems für die Windparkstandorte des Windparkbetreibers Simonsfeld. Das derzeitige Monitoringsystem soll auf ein mandantenfähiges System auf Basis der Open-Source-Software Icinga umgestellt und erweitert werden. Die einzelnen Windparkstandorte bestehen aus Windows- oder Linux-Servern und diversen Sensoren für die Messung unterschiedlichster Parameter. Besonderes Augenmerk ist auf die Ausfallssicherheit und da speziell auf den Ausfall einer IP-Leitung, zu legen.

Zusätzlich sollen der Aufbau einer Time-Based-Datenbank auf Basis von InfluxDB oder einer ähnlichen Datenbank und die Anzeige diverser Sensordaten mit Hilfe von Grafana oder Grafite oder ähnlichen Produkten getestet werden.

Zielsetzung

Ziel ist ein funktionsfähiges Monitoringsystem für alle Windparkstandorte auf Basis der OpenSource-Software Icinga. Das System soll auf den von Simonsfeld bereitgestellten Servern laufen und in deren derzeitige Überwachungsumgebung integriert werden. Zusätzlich sollen auf Basis einer Time-Based-Datenbank und einer modernen Grafikaufbereitung einzelne Parameter angezeigt werden.

Geplantes Ergebnis der Prüfungskandidatin/des Prüfungskandidaten

Geplante Ergebnisse sind die Installation und Konfiguration des Monitoringservers beim Kunden und die Anbindung der einzelnen Windparkstandorte in dieses System. Die wichtigsten Parameter der Standorte sollen aufgezeichnet und angezeigt werden und ein Ausfallstest durchgeführt werden. Die Integration in das Gesamtsystem von Simonsfeld wäre wünschenswert.

Meilensteine

18.05.2016 Projektstart

09.09.2016 Interne Abstimmung

30.09.2016 Installation des zentralen Icinga2 Servers (Testüberwachung)

30.11.2016 Installation der Satellite Nodes (Testüberwachung)

30.01.2017 Migration auf reale Umgebung

15.03.2017 Gesamtübernahme und Dokumentation



Inhalt

1 Einführung	6
2 Aufgabenstellung.....	6
3 Theoretische Kenntnisse	8
3.1 Ping.....	8
3.1.1 Beschreibung.....	8
3.1.2 Beispiel	8
3.1.3 Funktionsweise.....	10
3.1.4 Round-Trip-Delay	11
3.1.5 Verwandte Programme und Erweiterungen.....	12
3.1.6 Ping in Icinga2.....	12
3.2 SNMP	13
3.2.1 Allgemeines	13
3.2.2 GET-Request	13
3.2.3 SET-Request.....	14
3.2.4 TRAP	14
3.2.5 Version 1.....	14
3.2.6 Version 2 und Secure SNMP.....	14
3.2.7 Version 3.....	15
3.2.8 Paketaufbau	15
3.3 MIB	17
3.4 OID.....	19
3.4.1 Beschreibung.....	19
3.4.2 Notation	19
3.4.3 Wurzel	20
3.4.4 Beispiel	20



4 Softwarewahl.....	21
4.1 Python	21
4.1.1 Allgemeines	21
4.1.2 Vorteile	22
4.1.3 Nachteile	22
4.2 InfluxDB	23
4.2.1 Allgemeines	23
4.2.2 Vorteile	24
4.2.3 Nachteile	25
4.3 Grafana	25
4.3.1 Allgemeines	25
4.3.2 Vorteile	26
4.3.3 Nachteile	26
4.4 Net-SNMP	26
4.4.1 Allgemeines	26
4.5 EasySNMP.....	27
4.5.1 Allgemeines	27
4.5.2 Verwendung	27
4.6 PuTTY	27
4.6.1 Allgemeines	27
4.6.2 Oberfläche	28
4.7 Icinga2	29
4.7.1 Kurzbeschreibung.....	29
4.7.2 Funktionalität	30
4.7.3 Hosts und Services.....	33
4.7.4 Host- und Servicedefinitionen.....	36



4.7.5 Host- und Servicezustände.....	38
4.7.6 Benachrichtigungssystem.....	38
4.7.7 Verteiltes Netzwerk aufbauen	40
4.7.8 Icingaweb2	41
4.7.9 Graphite.....	42
4.8 Check_MK.....	44
4.8.1 Kurzbeschreibung.....	44
4.8.2 Funktionalität	45
4.8.3 Icinga2 oder Check_MK?.....	53
4.9 TeamViewer	54
4.9.1 Verbindungsaufbau	54
4.9.2 Datenübertragung	55
5 Testumgebung.....	57
5.1 Sensorüberwachung.....	57
5.1.1 Aufgabenstellung	57
5.1.2 Einrichten der Virtuellen Maschine.....	58
5.1.3 Testsensor	59
5.1.4 Einbinden des Testsensors in Icinga2.....	59
5.1.5 Erster Test.....	60
5.1.6 Fehlermeldung	60
5.1.7 Überwachung	61
5.1.8 Graphite.....	62
5.1.9 E-Mail-Benachrichtigungen.....	62
5.2 Verteiltes Netzwerk.....	65
5.2.1 Aufgabenstellung	65
5.2.2 Vorprojekt	65



5.2.3 Inbetriebnahme der PCs.....	67
5.2.4 Ubuntu Setup	67
5.2.5 Windows Setup	67
5.2.6 Einbinden ins verteilte Netzwerk	68
5.2.7 Einbinden der Clients	68
6 Zeitaufwand.....	73
6.1 Mujkanovic	73
6.2 Schrenk	97
7 Schlusswort.....	103
7.1 Icinga2	103
7.2 InfluxDB und Grafana	103
8 Anhang.....	104
8.1 Icinga2 installieren	104
8.2 Icingaweb2 installieren	106
8.3 Graphite installieren.....	107
8.4 Verteiltes Netzwerk aufbauen	111
8.4.1 Allgemeines	111
8.4.2 Master Setup	112
8.4.3 Satellite Setup	115
8.5 Installation Net-SNMP	120
8.5.1 Allgemeines	120
8.5.2 Anleitung	120
8.6 Installation Easy-SNMP.....	122
8.6.1 Allgemeines	122
8.6.2 Anleitung	123
8.6.3 Verwendung	123



8.7 Installation InfluxDB	124
8.7.1 Allgemeines	124
8.7.2 Anleitung	125
8.7.3 Verwendung	126
8.8 Installation Grafana	132
8.8.1 Allgemeines	132
8.8.2 Anleitung	132
8.8.3 Verwendung	134
8.9 Python-Script	141
8.9.1 Allgemeines	141
8.9.2 Code	142
9 Abbildungsverzeichnis	146
10 Quellenverzeichnis	150



1 Einführung

Die Diplomarbeit wurde von der Windkraft Simonsfeld AG in Auftrag gegeben. Auftraggeber war Michael Krell, Mitarbeiter der Monitoring-/IT-Abteilung. Das Unternehmen beschäftigt sich mit der Produktion von erneuerbarer Energie, der Projektierung und der Betriebsführung von alternativen Energieerzeugungsanlagen.

Windkraft Simonsfeld besitzt bereits eine Netzwerkuüberwachung über das Programm Icinga. Diese Überwachung war ebenfalls eine Diplomarbeit im Jahr 2011/2012 (von Florian Hannak und Patrick Huber).

2 Aufgabenstellung

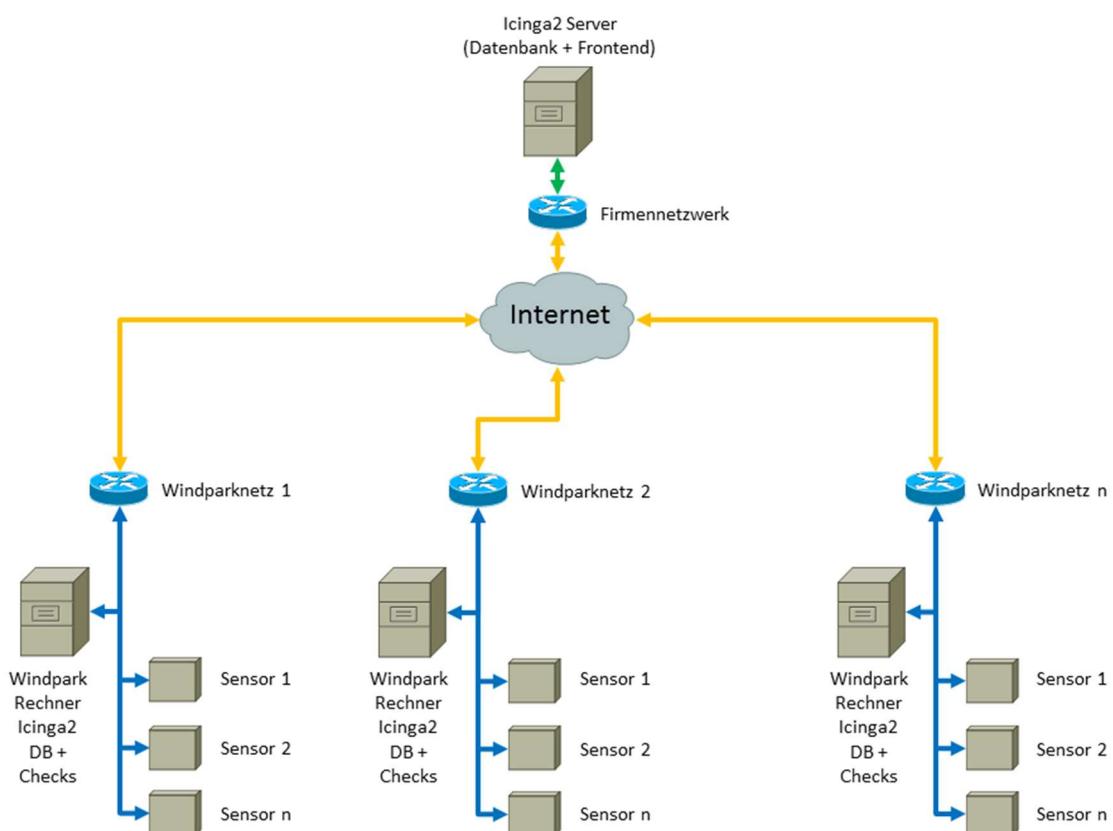


Abbildung 1 - prinzipielle Struktur Netzwerk



Ziel der Diplomarbeit ist es über einen zentralen Server das gesamte Netzwerk, inklusive der Netzwerke in den einzelnen Windparks, zu überwachen. Diese Überwachung wird einerseits mit dem Tool Icinga2, andererseits mit einem eigenen Script und Anzeige mittels SNMP durchgeführt.

Der Hauptserver, welcher in der obigen Abbildung „Icinga2 Server“ genannt wird, ist die zentrale Einheit, auf der sämtliche Daten gespeichert und verarbeitet werden. Die einzelnen Rechner in den Windparks sammeln lokal die Daten der im dortigen Netzwerk vorhandenen Sensoren und speichert sie zum einen lokal, sendet sie aber bei vorhandener Internetverbindung auch an den Icinga2-Server weiter.

Diese Überwachung wird mittels Bottom-Up-Prinzip durchgeführt. Dies bedeutet, dass die sogenannten „Slaves“, wie die Rechner in den Windparks bezeichnet werden, selbstständig die Daten an den Hauptserver senden. Dies verringert die Belastung des Servers, da er sich nicht um die Abfrage an jeden einzelnen Rechner kümmern muss und die Daten automatisch empfängt.

Weiters ermöglicht dieses Prinzip die Gewährleistung der Überwachung trotz eines Ausfalls des Internets, da die Slaves auch solange sie offline sind die Daten aus dem Netzwerk sammeln und dazu keine Anfrage über das Internet des zentralen Servers benötigen. Sollte die Verbindung also unterbrochen werden bedeutet dies keinen Datenverlust, sondern nur eine Verzögerung in der Übertragung für die Dauer des Ausfalls.



3 Theoretische Kenntnisse

3.1 Ping

3.1.1 Beschreibung

Mit einem sogenannten Ping kann überprüft werden ob ein bestimmtes IP-fähiges Gerät (z.B. ein Router, ein Switch, ein PC usw.) erreichbar ist. Zusätzlich zur Erreichbarkeit eines Hosts wird meistens auch die Zeit zwischen dem Aussenden eines Paketes zu einem Host und dem Empfangen eines darauf zurückgeschickten Antwortpaketes angezeigt (= Round-Trip-Delay(RTD) oder Round-Trip-Time(RTT)). Grundsätzlich kann man mit einem Ping überprüfen, ob das Routing zur Gegenstelle funktioniert, deren TCP/IP-Stack funktionsfähig ist und, wie schon beschrieben, mit welcher Verzögerung bei einer Verbindung zu rechnen ist. [WIK17i]

Der Ping wird prinzipiell bei jeder Überwachung verwendet da durch diesen geklärt werden kann, ob ein Sensor Down ist oder er aus einem anderen Grund seine Arbeit nicht vollrichten kann.

3.1.2 Beispiel

Der Befehl in der Kommandozeile sieht in Windows und Ubuntu gleich aus. Der Unterschied ist jedoch, dass Windows automatisch nach **4-mal** den ping abbricht, während Ubuntu den Befehl solange durchführt bis er manuell abgebrochen wird.

Als Beispiel wurde unter Ubuntu ein Sensor angepingt welcher eine interne IP-Adresse besitzt:

```
root@icinga:~# ping 10.48.1.31
PING 10.48.1.31 (10.48.1.31) 56(84) bytes of data.
64 bytes from 10.48.1.31: icmp_seq=1 ttl=254 time=54.6 ms
64 bytes from 10.48.1.31: icmp_seq=2 ttl=254 time=1.24 ms
64 bytes from 10.48.1.31: icmp_seq=3 ttl=254 time=1.30 ms
64 bytes from 10.48.1.31: icmp_seq=4 ttl=254 time=1.26 ms
64 bytes from 10.48.1.31: icmp_seq=5 ttl=254 time=1.57 ms
64 bytes from 10.48.1.31: icmp_seq=6 ttl=254 time=1.29 ms
```

Abbildung 2 – Ping unter Ubuntu

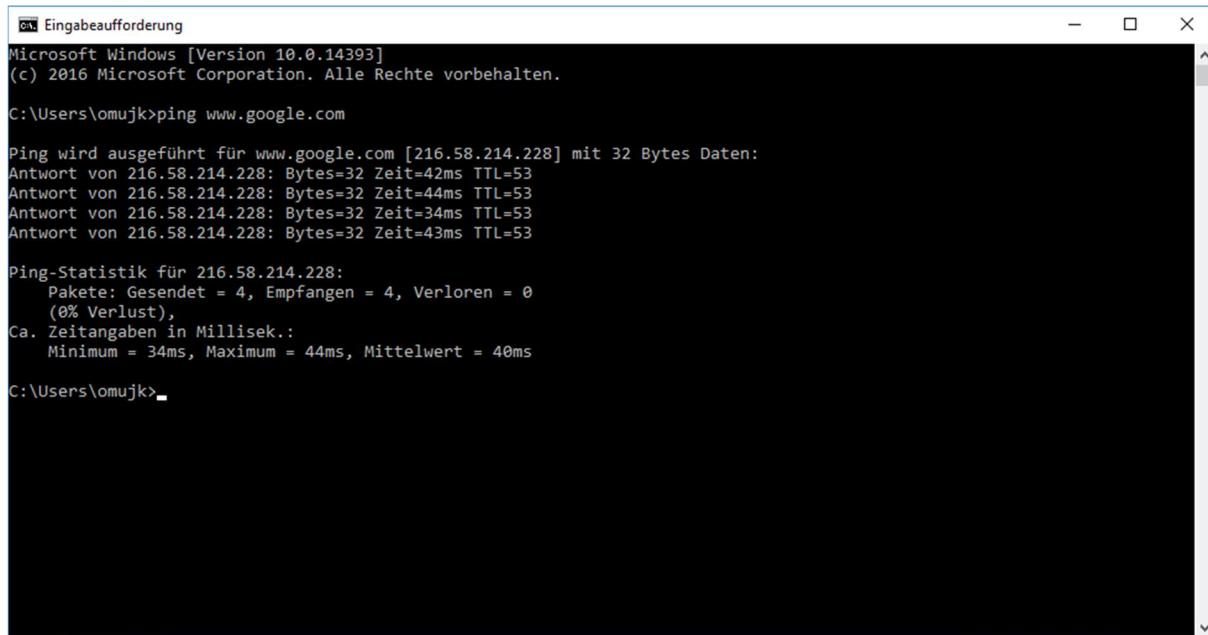


Wie auf der Abbildung zu sehen ist, wurde der ping öfter als 4-mal durchgeführt. Nach 6-mal wurde er mit der Tastenkombination **Strg+C** abgebrochen. Der ping kann aber auch direkt in der Kommandozeile – mit dem Zusatz -c4 - begrenzt werden:

```
root@icinga:~# ping 10.48.1.31 -c4
PING 10.48.1.31 (10.48.1.31) 56(84) bytes of data.
64 bytes from 10.48.1.31: icmp_seq=1 ttl=254 time=19.9 ms
64 bytes from 10.48.1.31: icmp_seq=2 ttl=254 time=1.28 ms
64 bytes from 10.48.1.31: icmp_seq=3 ttl=254 time=1.31 ms
64 bytes from 10.48.1.31: icmp_seq=4 ttl=254 time=1.32 ms
```

Abbildung 3 – Begrenzter Ping unter Ubuntu

Ein ping kann auch auf Webseiten durchgeführt werden da diese ebenfalls eine IP-Adresse besitzen. Um dies darzustellen wurde diese Art von ping unter Windows durchgeführt:



```
C:\> Eingabeaufforderung
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\omujk>ping www.google.com

Ping wird ausgeführt für www.google.com [216.58.214.228] mit 32 Bytes Daten:
Antwort von 216.58.214.228: Bytes=32 Zeit=42ms TTL=53
Antwort von 216.58.214.228: Bytes=32 Zeit=44ms TTL=53
Antwort von 216.58.214.228: Bytes=32 Zeit=34ms TTL=53
Antwort von 216.58.214.228: Bytes=32 Zeit=43ms TTL=53

Ping-Statistik für 216.58.214.228:
Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
(0% Verlust),
Ca. Zeitangaben in Millisek.:
Minimum = 34ms, Maximum = 44ms, Mittelwert = 40ms

C:\Users\omujk>_
```

Abbildung 4 – Ping unter Windows

Im Endeffekt gibt es keinen Unterschied zwischen einem ping auf eine Website oder ein physikalisches Gerät. Der ping unter Ubuntu und unter Windows funktioniert ebenfalls gleich und man muss sich auf keine Unterschiede gefasst machen.



3.1.3 Funktionsweise

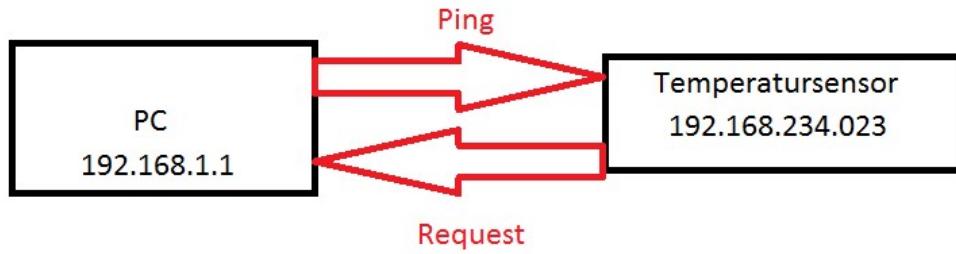
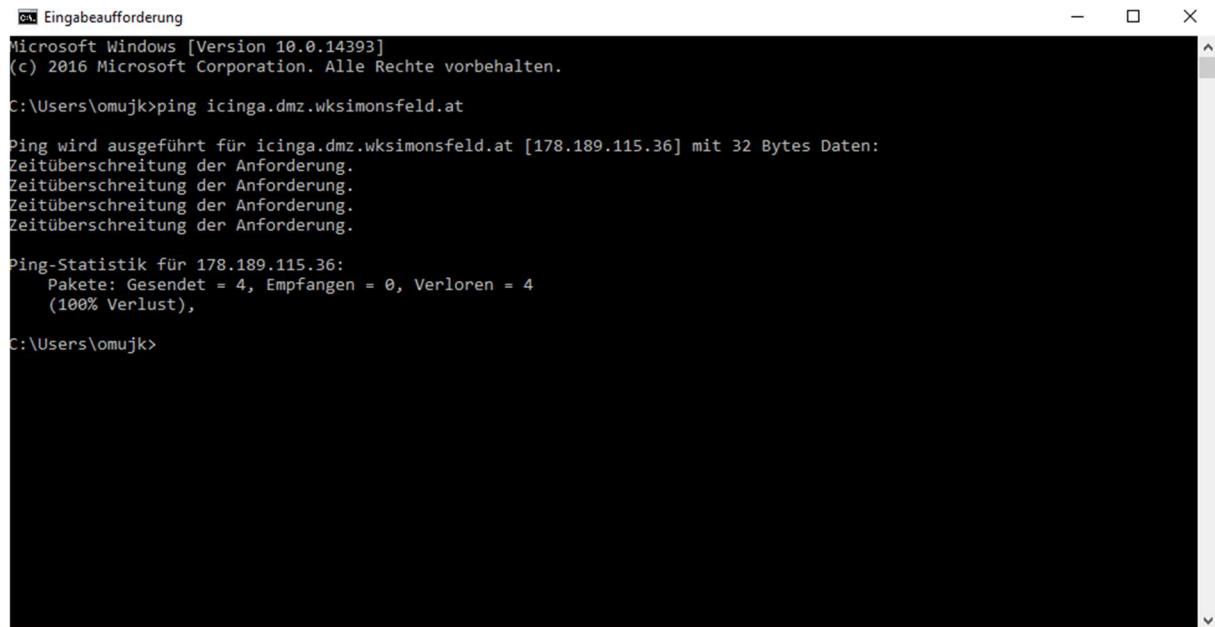


Abbildung 5 – Ping Funktionsweise

Es werden ICMP Pakete an einen Server, einen PC, einen Sensor usw. gesandt und es wird die Erreichbarkeit überprüft. Zusätzlich wird die Zeit gemessen bis die Antwort des Hosts eintrifft (hin und zurück). In unserem Beispiel ca. 42ms (response time average). Mit der Angabe TTL (Time to Live) kann man grob abschätzen über wieviele Router die ICMP Pakete gelaufen sind. Jeder Router dekrementiert den Wert um mindestens 1 wobei der Initialwert je nach Implementierung 64, 128 etc. sein kann.

Bei einem Ping wird ein ICMP(v6)-„Echo-Request“-Paket an die Zieladresse gesendet. Der entsprechende Empfänger sendet, sofern er das Protokoll unterstützt, eine Antwort ICMP-„Echo-Reply“ zurück.

Ist der zu überprüfende Host nicht erreichbar so sendet der dafür zuständige Router „Network unreachable“ (Netzwerk nicht erreichbar) oder „Host unreachable“ (Host nicht erreichbar). Wenn die Zeitspanne, zwischen der Anfrage und Empfangen der Antwort zu groß ist wird „Zeitüberschreitung der Anforderung ausgegeben“.



```
cmd Eingabeaufforderung
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\omujk>ping icinga.dmz.wksimonsfeld.at

Ping wird ausgeführt für icinga.dmz.wksimonsfeld.at [178.189.115.36] mit 32 Bytes Daten:
Zeitüberschreitung der Anforderung.
Zeitüberschreitung der Anforderung.
Zeitüberschreitung der Anforderung.
Zeitüberschreitung der Anforderung.

Ping-Statistik für 178.189.115.36:
Pakete: Gesendet = 4, Empfangen = 0, Verloren = 4
(100% Verlust),

C:\Users\omujk>
```

Abbildung 6 – Ping Zeitüberschreitung der Anforderung

In diesem Fall ist jedoch nicht sofort die Gegenstelle offline. Durch Sicherheitsfunktionen wie einer Firewall werden Geräte geschützt und sind somit auch nicht per ping erreichbar.

3.1.4 Round-Trip-Delay

Kurz RTD wird auch round-trip-time, Paketumlaufzeit oder als Pingdauer bezeichnet. In dieser Zeit sind enthalten:

- Signallaufzeit zwischen Sender und Empfänger
- Die Verarbeitungszeit in den Zwischenstationen (Router, Switches etc.)
- Die Verarbeitungszeit in den TCP/IP-Stacks des Absende- und des Zielrechners

Auch bei Verbindungen mit hoher Auslastung erhöht sich die RTD (durch Wartezeiten in Routern usw.) [WIK17j].



3.1.5 Verwandte Programme und Erweiterungen

Folgend einige Erweiterungen vom Diagnose-Programm ping und Programme die ähnlich arbeiten:

- **bing** (misst zusätzlich die Übertragungsrate)
- **fping** (kann mehrere Rechner zugleich anpingen, Broadcast)
- **hping** (erlaubt die Manipulation von Protokollen)
- **arping** (liest im lokalen Netzwerk auch die MAC-Adresse aus)
- **mtr** (kombiniert die Funktionalität von „traceroute“ und „ping“ in einem Netzwerkdiagnoseprogramm)
- **httping** (ermittelt die Antwortzeit für einen URL)
- **ping6** (unter Linux für IPv6, in Windows bedient ping beide Protokolle(IPv4/IPv6), einstellbar durch die Optionen -4 -6)

3.1.6 Ping in Icinga2

Der Ping wird von Icinga2 automatisch ausgeführt, wenn dieser für den Host als Service angelegt wurde.

Ausgabe des Plugins

```
PING OK - Packet loss = 0%, RTA = 0.06 ms
```

Abbildung 7 – Ping in Icinga2

Das Tool führt den Ping selbstständig aus und gibt auf der GUI den Packet Loss an (Pakete, die nicht zurückgeschickt wurden) sowie die RTA (Round Trip Average)



3.2 SNMP

3.2.1 Allgemeines

Das **Simple Network Management Protocol**, oder kurz SNMP, wurde von der IETF, einer Organisation, die sich mit der technischen Weiterentwicklung des Internets befasst, entwickelt. Es dient um Netzwerkelemente wie zum Beispiel Router, Server, Switches, Drucker usw. von einer zentralen Station aus zu überwachen und zu steuern. Das Protokoll ist hierbei für die Kommunikation zwischen dieser Station und der zu überwachenden Station verantwortlich. Es wurde so ausgeführt, dass sämtliche netzwerkfähige Geräte in die Überwachung einbezogen werden können. [WIK17a]

Zur Kommunikation verwendet das in der "RFC 1157 - SIMPLE NETWORK MANAGEMENT PROTOCOL (SNMP)" beschriebene Protokoll meist TCP/IP, aber auch IPX und diverse andere Protokolle. Um die Belastung des Netzwerks gering zu halten dient der UDP-Port 161 dazu die Anfragen des Masters am Agent zu empfangen und der UDP-Port 162 dazu die TRAP-Meldungen der Clients am Manager zu empfangen.

Hauptsächlich werden die folgenden drei Befehle genutzt:

- GET
- SET
- TRAP

3.2.2 GET-Request

Bei diesem Befehl wird der Client vom Server aktiv abgefragt. Dies bedeutet, dass das System entweder einmal, oder bei Erstellung einer Grafik auch regelmäßig, die Daten des Clients abfragt



3.2.3 SET-Request

Bei diesem Request sendet der Manager eine Anfrage an den Agent um den Wert einer der Variablen zu verändern. Diese Änderungen sind als atomare Operationen zu behandeln, was bedeutet, dass sie nicht unterbrochen werden dürfen. Der Agent liefert als Antwort die neuen Werte der Variablen zurück. [WIK17b]

3.2.4 TRAP

In diesem Fall sendet der Agent ohne Aufforderung des Managers eine Nachricht an diesen. Dies findet häufig in der Meldung von Fehlfunktionen Anwendung, kann aber auch eine Rückmeldung auf einen SET-Request sein, falls die beschriebenen Datensätze nicht durchgeführt werden konnten. Ein TRAP wird jedoch nicht vom Manager bestätigt, was es dem Client nicht ermöglicht zu kontrollieren ob sein TRAP vom Manager erhalten wurde oder nicht.

3.2.5 Version 1

Die erste Version von SNMP wurde 1988 in den RFCs 1155, 1156 und 1157 definiert. Bei dieser Version war jedoch die Sicherheit ein sehr großes Problem, da das Passwort unverschlüsselt übertragen wird und dadurch sehr anfällig für Abhörungen ist.

3.2.6 Version 2 und Secure SNMP

Das Secure SNMP wurde aus dem steigenden Bedürfnis nach Sicherheit 1992 definiert, jedoch nie veröffentlicht, da es durch Version 2 ersetzt wurde.

Die zweite Version besteht grundlegend aus drei Teilen, es gibt „Party-Based“, „User-Based“ und „Community-Based“.

3.2.6.1 Party-Based

Diese Variante wurde 1993 von der IETF veröffentlicht und verbesserte die Sicherheit und Vertraulichkeit des Protokolls. Hierbei wird der Community-String verschlüsselt, die Kommunikation zwischen Managern ermöglicht und der „GetBulk“-Befehl eingefügt. Dieser erleichtert die Abfrage von Tabellen zwar erheblich, dennoch wird diese Version heute nicht mehr verwendet.



3.2.6.2 User-Based

Dieser Typ wurde in den RFCs 1909 und 1910 definiert und man fügte Benutzernamen um die Sicherheit zu verbessern, dennoch wird auch diese Version heute nicht mehr verwendet.

3.2.6.3 Community-Based

Die Version v2c wurde gegenüber SNMPv1 lediglich um die Funktionen von v2p erweitert. GetBulk und die Kommunikation zwischen den Managern wurden eingeführt, die Sicherheit jedoch blieb auf dem Stand der ersten Version.

Definiert wurde diese Variante in den RFCs 1901, 1905 und 1906.

3.2.7 Version 3

In der dritten Version wurde auf die Sicherheit des Protokolls viel Wert gelegt, wodurch auch die Komplexität stark anstieg. Dadurch bedingt ist SNMPv3 noch nicht so stark verbreitet wie der Vorgänger SNMPv2.

Definiert wurde diese Version in den RFCs 3410 bis 3418.

3.2.8 Paketaufbau

SNMP-Paket-Header	
Versionsnummer	Community Name

Abbildung 8 – SNMP-Paket-Header

Der erste Teil des SNMP-Pakets ist der sogenannte „SNMP-Paket-Header“, welcher die Versionsnummer des Protokolls und den Community Name der Anfrage enthält.



PDU-Header			
Pakettyp (Get, GetNext, ...)	RequestID	Fehlerstatus	Fehler-Index

Abbildung 9 – SNMP-PDU-Header

Der zweite Teil ist der „PDU-Header“, was für Protocol Data Units steht. In diesem Paketeil steht der Pakettyp, welcher zum Beispiel Get oder Set sein kann und eine RequestID, welche für die Zuordnung der Antwort zur Anfrage verwendet wird. Diese ID ist bei beiden Paketen gleich um diese Zuordnung möglich zu machen. Weiters gibt es einen Fehlerstatus und einen Fehlerindex, welche im Fehlerfall mitteilen welcher Fehler aufgetreten ist. Ein Fehler könnte zum Beispiel sein, dass eine Set-Methode ausgeführt werden soll obwohl nur Leserechte vorhanden sind. Tritt kein Fehler auf sind diese beiden Teile mit dem Wert Null befüllt.

PDU-Body			
Variable Binding 1	Variable Binding 2	...	Variable Binding n

Abbildung 10 – SNMP-PDU-Body

Der letzte Teil ist der „PDU-Body“, in welchem die eigentlichen Werte übertragen werden. Jede „Variable Binding“ enthält die gesendete OID und den zugehörigen Wert.

Eine genaue Größe eines SNMP-Pakets kann nicht angegeben werden, da die Größe des PDU-Bodys abhängig von der Anzahl der abgefragten Informationen ist.



3.3 MIB

Die **Management Information Base** ist eine hierarchische Ansammlung von Informationen über ein Gerät. Jedes administrierbare Gerät enthält eine MIB, welche die Informationen als Objekte mit eigenen Objektcodes verwaltet.[WIK17c]

Sie wird von Protokollen wie zum Beispiel SNMP dazu verwendet, um auf verschiedene Informationen zuzugreifen.

Es gibt skalare und tabellarische MIBs, welche sich dadurch unterscheiden, dass skalare ein einziges Objekt definieren und tabellarische mehrere verwandte Objekte in MIB Tabellen zusammenfassen.

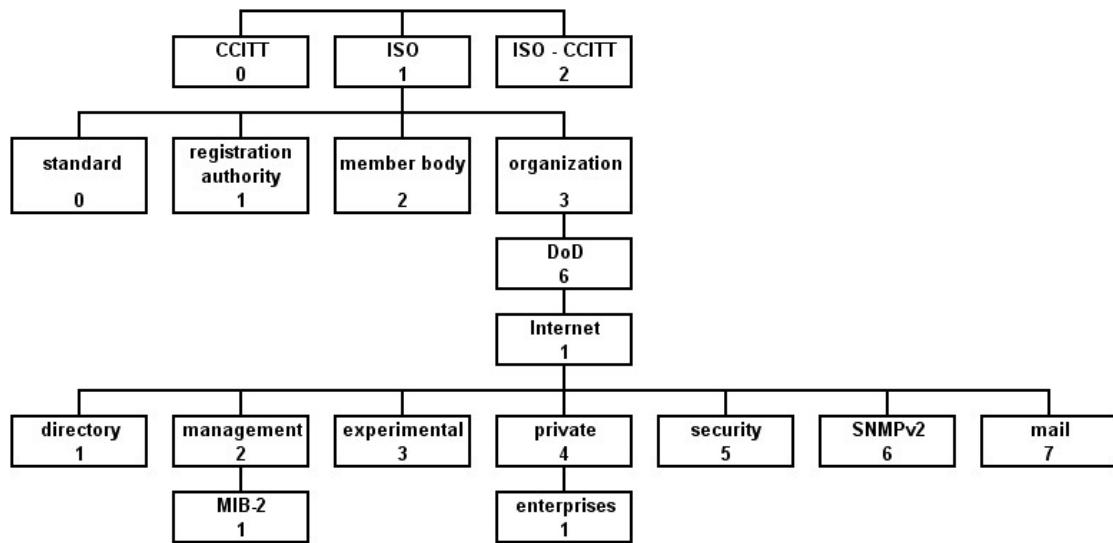


Abbildung 11 – MIB-Baum

In der obigen Grafik ist der Aufbau des MIB-Baumes zu sehen, der von uns verwendete Pfad lautet „1.3.6.1.4.1“, da dort Firmen ihre privaten MIBs anlegen können. Dazu müssen sie eine sogenannte „Enterprise ID“ bei der IANA, welche für die Zuordnung von Nummern und vor Allem IP-Adressen im Internet zuständig ist, beantragen. [WIK17d]

Eine MIB wird mittels SMI-Syntax als ASCII-Datei geschrieben, in welcher Verzweigungen festgelegt, Angaben über Datentyp, Zugriffsrechte, Status und so weiter gemacht werden. Zur Überprüfung der Syntax gibt es sogenannte „MIB-Compiler“, welche den Code auf seine Richtigkeit überprüft und formatiert.



Eine MIB in SMI-Syntax könnte beispielsweise folgendermaßen aussehen:

```
system    OBJECT IDENTIFIER ::= { mib-2 1 }
sysDescr OBJECT-TYPE
    SYNTAX  DisplayString (SIZE (0..255))
    ACCESS  read-only
    STATUS   mandatory
    DESCRIPTION
        "A textual description of the entity. This value
         should include the full name and version
         identification of the system's hardware type,
         software operating-system, and networking
         software. It is mandatory that this only contain
         printable ASCII characters."
    ::= { system 1 }
```

Es wird zuerst die OID definiert und anschließend das Objekt als „OBJECT-TYPE“, dieses enthält zum Beispiel Informationen über den Zugriff und eine Beschreibung.



3.4 OID

3.4.1 Beschreibung

Ein **Object Identifier** ist ein weltweit eindeutiger Bezeichner, der benutzt wird um ein Informationsobjekt zu benennen. Die OID stellt einen Knoten in einem hierarchisch zugewiesenen Namensraum dar, der durch den ASN-1 Standard definiert ist. Jeder Knoten ist durch eine Folge von Nummern gekennzeichnet, die seine Position beginnend an der Wurzel des Baums angibt.

Die Regeln für die Vergabe von OIDs sind festgelegt in den Normen **ISO/IEC 9834 und DIN 66334**. Die Verwaltung des OID-Baumes und die Sicherstellung der Eindeutigkeit von OIDs beruhen auf der Übertragung der Zuständigkeit für die untergeordneten Knoten an den Besitzer einer OID.

3.4.2 Notation

Im internationalen OID-Baum werden alle Kanten von einem Knoten zu einem Kindknoten durch ganze Zahlen, sogenannte **primary integer values**, eindeutig identifiziert. Jede Kante ist gleichermaßen mit einer eindeutigen **Unicode-Bezeichnung (Unicode label)** versehen.

Jede Kante kann eine oder mehrere sekundäre Bezeichnungen – **secondary identifiers** – haben. Für die sekundären Bezeichnungen gelten die Zeichenrestriktionen für ASN-1 Bezeichnungen. Die sekundären Bezeichnungen müssen nicht eindeutig sein.

Jedes Objekt ist genau einem Knoten zugeordnet. Für ein solches Objekt wird der ASN-1 Object Identifier im engen Sinne als Folge der Kantennummer auf dem Pfad von der Wurzel zu dem entsprechenden Knoten definiert.

```
{1 3 6 1 4 1} gemäß ASN.1-Notation,  
1.3.6.1.4.1 gemäß Punkt-Notation,  
urn:oid:1.3.6.1.4.1 gemäß URN-Notation
```

Abbildung 12 – OID-Notation



Um die Lesbarkeit von OIDs zu verbessern kann der Wert einer ASN1 OID unter Verwendung der zugehörigen sekundären Bezeichnungen referenziert werden:

```
{iso(1) org(3) dod(6) internet(1) private(4) enterprise(1)} gemäß ASN.1-Notation
```

Abbildung 13 – OID-Referenzen

3.4.3 Wurzel

Die Kanten von der Wurzel des OID-Baumes wurden wie folgt definiert:

1. ITU-T
2. ISO
3. Joint-iso-itu-t

Eine Erweiterung durch eine Standardisierungsorganisation wäre nicht möglich, da in der Binärkodierung von OIDs zugunsten einer Komprimierung kein weiterer Platz reserviert wurde.

3.4.4 Beispiel

Eine MIB kann wie folgt aussehen:

1.3.6.1.4.1.343

Der Aufbau der einzelnen OIDs sieht dann wie folgt aus:

```
1 ISO
  1.3 identified-organization,
    1.3.6 dod,
      1.3.6.1 internet,
        1.3.6.1.4 private,
          1.3.6.1.4.1 IANA enterprise numbers,
            1.3.6.1.4.1.343 Intel Corporation
```

Abbildung 14 – Beispiel OIDs

Die MIB kann in Icinga2 bei Serviceabfragen verwendet werden um einen bestimmten SNMP-Wert zu erhalten [WIK17k].



4 Softwarewahl

Um ein Monitoring, wie in unserem Fall, selbst zu programmieren sind eine Reihe von Tools beziehungsweise Plugins nötig. Zu wählen war eine Programmiersprache, in der das Script für die Abfrage geschrieben wird, eine passende SNMP-Library für diese Sprache, eine Datenbank mit zugehöriger Schnittstelle um eine Kombination mit dem Script zu ermöglichen und ein Tool zur Anzeige, das bevorzugt das gewählte Datenbankmanagementsystem unterstützt.

All diese Komponenten müssen aufeinander abgestimmt werden, weshalb folgende Tools gewählt wurden.

4.1 Python

4.1.1 Allgemeines



Abbildung 15 – Python-Logo

Als Programmiersprache wurde Python gewählt, weil es eine interpretierte, objektorientierte höhere Programmiersprache mit einfacher Syntax ist. Dies bedeutet, dass ein Programm nicht beim Start durch einen Compiler übersetzt, sondern während der Laufzeit von einem Interpreter interpretiert wird, was eine hohe Plattformunabhängigkeit ermöglicht. [PYT17]

Weiters fördert Python die Lesbarkeit der Codes enorm, da anstatt der klassischen Klammern oder Schlüsselwörter zur Abgrenzung der einzelnen Codeblöcke ein strukturierter Programmierstil verwendet wird. Die Blöcke werden also durch die Unterschiede in der Einrückung unterschieden.



4.1.2 Vorteile

Python ist gratis und einfach zu bekommen, wodurch sich eine große Community gebildet hat, welche eine Vielzahl an Open-Source Erweiterungen programmierte. Diese können einfach in Foren gefunden und heruntergeladen werden. [MIH17]

Ein wesentlicher Vorteil ist die bereits angesprochene Lesbarkeit des Codes. Python funktioniert nur dann, wenn die Formatierung eingehalten wird, da es anders nicht möglich ist, Codeblöcke zu definieren.

Dank des Python-Interpreters, welcher auf sehr vielen Plattformen zur Verfügung steht, ist die Sprache sehr plattformunabhängig und Programme können ohne viel Aufwand auch auf anderen Betriebssystemen verwendet werden.

4.1.3 Nachteile

Die Syntax ist zwar sehr einfach und ermöglicht ein leichtes Lernen der Sprache, jedoch ist ein Code durch die Formatierung ziemlich fehleranfällig und diese Fehler sind dann häufig schwer zu finden, da es sich nicht zwingend um einen Programmierfehler handeln muss, sondern simpel ein Fehler in der Einrückung vorhanden sein kann.

Da Python erst während der Laufzeit interpretiert wird sind die Programme im Verhältnis zu kompilierten Codes relativ langsam und Fehler fallen nicht vor dem Programmstart auf, sondern erst, wenn dieser Code-Teil verarbeitet werden soll. Dadurch können Fehler in Programmteilen, welche nur selten abgearbeitet werden, für längere Zeit nicht auffallen.



4.2 InfluxDB

4.2.1 Allgemeines



Abbildung 16 – InfluxDB-Logo

InfluxDB ist eine Zeitreihen Datenbank, das heißt es wird standardmäßig zu jedem Eintrag ein Zeitstempel gespeichert. Dieser ist nur für die Maschine und nicht für den Menschen lesbar, da sie in Nanosekunden angegeben ist. Der Zeitstempel ist gleichzeitig ein einmaliger Wert, der jeden Eintrag eindeutig macht, ein „Primary Key“ also. [DIN17a]

Eine Zeitreihen Datenbank ist eine Datenbank, die speziell für Anwendungen ausgelegt ist, die eine Zeiterfassung oder Echtzeitzugriffe benötigen. Dies ist zum Beispiel bei vielen Sensordaten der Fall, da man häufig beispielsweise einen Temperaturverlauf über die Zeit anzeigen möchte. [WIK17e]

Es wird eine SQL-ähnliche Sprache namens InfluxQL verwendet, welche den Umstieg von SQL-Datenbanken zu der NoSQL-Datenbank InfluxDB erleichtern soll.

Der „select“-Befehl wurde von SQL übernommen und so erweitert, dass es möglich ist einen Zeitraum, in dem die Daten aufgetreten sein sollen, anzugeben.

Der folgende Befehl gibt alle Daten ab dem 28. März 2017 nach 14:40:46 Uhr aus.

```
select * from temp_lager where time > '2017-03-28T12:40:56.000000000Z'
```

In der Syntax ist es jedoch nötig die GMT (Greenwich Mean Time) zu verwenden, da der Zeitstempel der Datenbank in diesem Format eingetragen wird.

Weiters wird in InfluxDB mehr Wert darauf gelegt Daten zu erstellen und zu lesen als diese zu verändern und zu löschen, es handelt sich also nicht um eine „CRUD“-Datenbank, sondern eher um eine „CRud“-Datenbank. [WIK17f]

CRUD steht für **C**reate **R**ead **U**update **D**elete und gibt an wofür eine Datenbank hauptsächlich verwendet wird. Da es bei InfluxDB aber eher um Create und Read geht als um Update und Delete wird sie als “CRud“ bezeichnet.



4.2.2 Vorteile

Ein wesentlicher Vorteil von InfluxDB ist die Schemafreiheit, welche das Organisieren von sehr großen Datenmengen im Vergleich zu klassischen Datenbanken mit Schema sehr stark vereinfacht. [DBE17]

Weiters bietet InfluxDB, im Gegensatz zu manchen anderen NoSQL-Datenbanken, eine sehr stark an SQL angelehnte Sprache namens InfluxQL. Diese macht den Umstieg von der Welt der klassischen DBMS in die der NoSQL-DBMS sehr einfach und es ist kaum Umgewöhnung nötig.

Darüber hinaus unterstützt InfluxDB viele bekannte Programmiersprachen wie Java, JavaScript, PHP, Python und so weiter und bietet ebenso zwei unterschiedliche Arten des Zugriffs. Dieser kann entweder über eine HTTP API oder per JSON-Format erfolgen.

Das verwendete JSON-Format sieht folgendermaßen aus:

```
„measurement“: „name“,
„tags“: {
    „host“: „name“,
    „region“: „name“
},
„fields“: {
    „SysTime“: variable1,
    „value“: variable2
}
```

Ein Measurement ist vergleichbar mit einem Table bei SQL Datenbanken, ein Tag ist zum Beispiel eine Information über den Standort und ein Field ist ein zu erfassender Wert.

[IPD17]



4.2.3 Nachteile

Einer der Nachteile von InfluxDB ist die kleine Community, da diese Datenbank noch nicht sehr weit verbreitet ist und angewandt wird. Dadurch ist die zu findende Hilfe zu spezifischen Problemen sehr stark begrenzt, was es nötig macht den Fehler selbst zu finden und zu beheben.

4.3 Grafana

4.3.1 Allgemeines



Abbildung 17 – Grafana-Logo

Grafana ist ein Tool zur Visualisierung von unterschiedlichsten Daten, welches nach dem Vorbild von Graphite entwickelt wurde. Es soll Schwierigkeiten, die der Entwickler von Grafana mit Graphite hatte, lösen und die Bedienung vereinfachen. [YTB17]

Grafana stellt Daten aus einer Datenbank in meist mehreren Graphen dar, welche zu einem sogenannten „Dashboard“ zusammengefasst werden. [DGR17]



Abbildung 18 – Graphen Überwachung



4.3.2 Vorteile

Grafana unterstützt standardmäßig diverse Datenbanksysteme wie zum Beispiel auch InfluxDB, was eine gemeinsame Verwendung dieser beiden Tools ohne viel Aufwand ermöglicht. Weiters werden Graphite, OpenTSDB, CloudWatch und einige mehr unterstützt.

Die grafische Oberfläche ist sehr modern und übersichtlich gestaltet, was ein schnelles Einarbeiten ermöglicht. Im Gegensatz zu anderen vergleichbaren Tools sind vor allem die Einstellungen der einzelnen Graphen, wie zum Beispiel das Auswählen der Daten, die Einstellung der Achsen oder die Anzeige der Kurve sowie der verwendeten Werte, sehr einfach und verständlich realisiert.

4.3.3 Nachteile

Ein Nachteil von Grafana ist die Reichweite, welche bisher noch sehr begrenzt ist. Bei Fehlern ist es schwierig jemanden zu finden, der das gleiche Problem hatte und eine Lösung kennt.

4.4 Net-SNMP

4.4.1 Allgemeines



Abbildung 19 – Net-SNMP-Logo

Net-SNMP ist eine Sammlung von zusammenspielenden Applikationen, welche die Implementierung von SNMP v1, v2c und v3 sowohl unter IPv4 als auch IPv6 ermöglicht. In diesem Paket, zu welchen die Entwicklungen zusammengefasst wurden, befinden sich zum Beispiel Command-Line Applikationen, ein MIB-Browser, einen erweiterbaren Agenten zur Beantwortung von SNMP-Abfragen und einiges mehr. [NET17]

Verfügbar ist Net-SNMP sowohl für viele Unix-Betriebssysteme, also auch für Microsoft Windows.



4.5 EasySNMP

4.5.1 Allgemeines

EasySNMP ist ein SNMP-Client API, welche speziell für Python entwickelt wurde und das Problem der Leistung der PySNMP-Library beheben soll. Da sie auf Net-SNMP basiert ist es nötig dies zu installieren, bevor EasySNMP installiert wird. [EAS17a]

4.5.2 Verwendung

Die Syntax von EasySNMP sieht bei einer Abfrage zum Beispiel folgendermaßen aus:

```
inst = snmp_get('OID', hostname='name or IP', community='communityname',  
version=1)
```

„inst“ wird in diesem Fall als Name verwendet, weil der Befehl „snmp_get“ der Aufruf einer Klasse ist, welche eine Instanz mit sämtlichen Informationen über das Gerät zurückliefert. Die einzelnen Werte sind in dieser Klasse als Attribute vorhanden.

4.6 PuTTY

4.6.1 Allgemeines



Abbildung 20 – PuTTY-Logo

PuTTY ist ein kostenloser Terminal emulator für Windows und Linux zur Herstellung von SSH-, Telnet-, oder Remote-login-Verbindungen. Dieser SSH-Client wird benötigt, da Windows standardmäßig keinen Client installiert hat und eine Remote-Verbindung zum Server nötig ist. [WIK17g]



4.6.2 Oberfläche

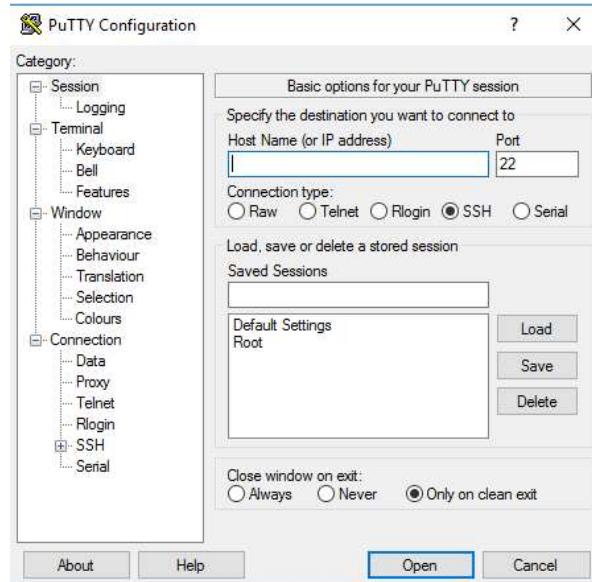


Abbildung 21 – Oberfläche PuTTY

Um eine SSH-Verbindung herzustellen ist es nur nötig entweder den Host-Namen oder die IP-Adresse des Servers, zu dem die Verbindung aufgebaut werden soll, anzugeben. Der Port wird von PuTTY standardmäßig je nach Auswahl des Verbindungstypen korrekt hinzugefügt.

Wird eine Verbindung häufiger benötigt ist es ratsam unter „Saved Sessions“ einen Namen für diese einzugeben und anschließend mit „Save“ die Einstellungen zu speichern.



4.7 Icinga2



Abbildung 22 – Icinga2

Icinga2 ist ein Wunsch des Auftraggebers da die bisherige Überwachung mit Icinga realisiert wurde und man sehr zufrieden mit diesem Tool sei. Darüber hinaus ist das Programm eine Open Source Software, somit sind alle Inhalte gratis zur Verfügung gestellt [EPE17a].

4.7.1 Kurzbeschreibung

Icinga2 ist der Nachfolger des Tools Icinga. Die Anwendung wurde am 16. Juni 2014 veröffentlicht und wird seitdem stetig weiterentwickelt. Das neue Monitoring-Programm der Firma Netways baut dabei nicht auf dem Vorgänger auf, sondern wurde komplett neu geschrieben um alte Einschränkungen aufzulösen und viele Neuerungen zu bringen. Diese Neuerungen sind:

- Umstieg von C auf C++
- Neue Konfigurationssprache
- Integrierter Cluster-Stack (Icinga2 Instanzen können miteinander kommunizieren)
- Verteilen einer Konfiguration über das Clusterprotokoll an alle Icinga2 Instanzen
- Besseres Ausnutzen der verfügbaren Ressourcen durch Multithreading
- Einsatz als Agent
- API, mit der zu überwachende Objekte zur Laufzeit hinzugefügt werden können

Die neue regelbasierte Sprache erlaubt eine deutlich flexiblere Konfiguration: Viele Objekte, die früher einzeln definiert worden sind können nun automatisch durch Auswerten von Regeln und Funktionen angegeben werden.



4.7.2 Funktionalität

4.7.2.1 Aufbau von Icinga2

Icinga2 besteht aus dem Icinga2-Core und der GUI Icingaweb2. Icingaweb2 kann auf einem Webserver ausgeführt werden wie z.B.: Apache2.

Unter Ubuntu sind die Konfigurations-Files unter **/etc/icinga2** zu finden:

```
root@icinga:/etc/icinga2# ls -la
total 60
drwxr-x---  9 nagios  nagios  4096 Feb  3 09:06 .
drwxr-xr-x 109 root   root   4096 Feb  7 11:00 ..
drwxr-xr-x  2 root   root   4096 Feb  6 16:37 conf.d
-rw-r--r--  1 root   root   1025 Feb  1 15:51 constants.conf
-rw-r--r--  1 root   root   998 Feb  1 15:51 constants.conf.orig
drwxr-xr-x  2 root   root   4096 Feb  1 15:51 features-available
drwxr-xr-x  2 root   root   4096 Feb  1 15:51 features-enabled
-rw-r--r--  1 root   root  1881 Feb  1 00:00 icinga2.conf
-rw-r--r--  1 root   root  164 Apr  5 2016 init.conf
drwxr-x---  2 nagios nagios  4096 Feb  1 15:51 pki
drwxr-xr-x  5 root   root  4096 Feb  1 16:00 repository.d
drwxr-xr-x  2 root   root  4096 Feb  1 15:43 scripts
-rw-r--r--  1 root   root  168 Feb  1 15:51 zones.conf
-rw-r--r--  1 root   root  692 Feb  1 15:51 zones.conf.orig
drwxr-xr-x  2 root   root  4096 Feb  1 15:43 zones.d
```

Abbildung 23 – Icinga2 unter Ubuntu

Unter Windows sind die Konfigurations-Files unter **C:\Program Files (x86)\ICINGA2\etc\icinga2** zu finden:

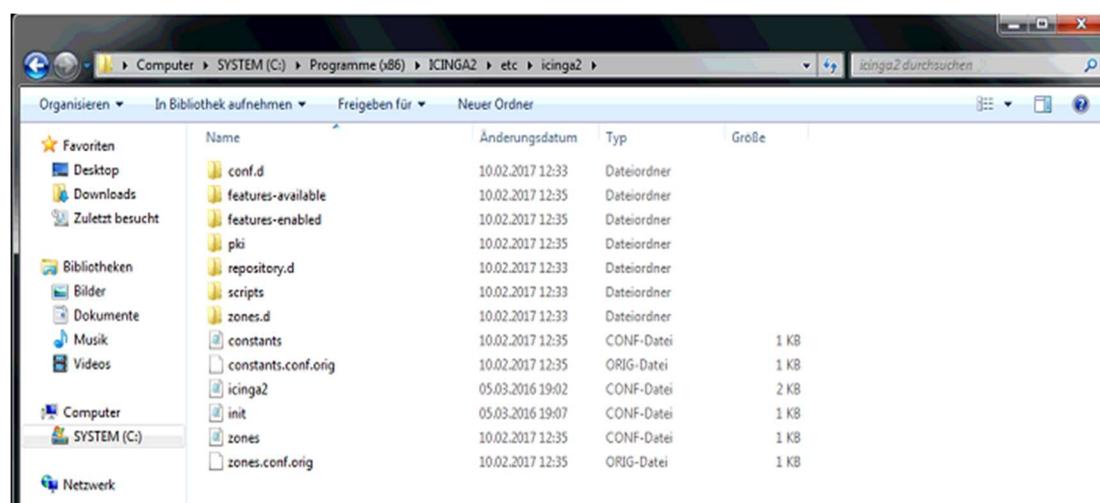


Abbildung 24 – Icinga2 unter Windows



Icinga2 wird daher entweder über die Kommandozeile oder über einen Windows Editor konfiguriert. Für die Konfiguration wurde eine eigene Konfigurationssprache entwickelt (stark an C/C++ angelehnt) welche es ermöglicht Icinga2 auf den beiden Betriebssystemen gleich konfigurieren zu können.

Anmerkung: Da somit alles gleich ist unter beiden Betriebssystem wurden die folgenden Screenshots nur unter Ubuntu 16.04 aufgenommen!

In der Datei conf.d werden alle Hosts, Services, User usw. konfiguriert:

```
root@icinga:/etc/icinga2/conf.d# ls -la
total 64
drwxr-xr-x 2 root root 4096 Feb  6 16:37 .
drwxr-x--- 9 nagios nagios 4096 Feb  3 09:06 ..
-rw-r--r-- 1 root root 225 Feb  1 15:52 api-users.conf
-rw-r--r-- 1 root root 35 Apr  5 2016 app.conf
-rw-r--r-- 1 root root 114 Feb  4 2016 apt.conf
-rw-r--r-- 1 root root 1220 Feb  1 00:00 commands.conf
-rw-r--r-- 1 root root 542 Apr  5 2016 downtimes.conf
-rw-r--r-- 1 root root 638 Apr  5 2016 groups.conf
-rw-r--r-- 1 root root 1501 Apr  5 2016 hosts.conf
-rw-r--r-- 1 root root 674 Apr  5 2016 notifications.conf
-rw-r--r-- 1 root root 801 Apr  5 2016 satellite.conf
-rw-r--r-- 1 root root 2132 Apr  5 2016 services.conf
-rw-r--r-- 1 root root 1024 Feb  6 15:25 .services.conf.swp
-rw-r--r-- 1 root root 1654 Apr  5 2016 templates.conf
-rw-r--r-- 1 root root 906 Apr  5 2016 timeperiods.conf
-rw-r--r-- 1 root root 308 Apr  5 2016 users.conf
```

Abbildung 25 – Datei conf.d

In der Datei features-enabled sind alle aktivierten Features zu finden:

```
root@icinga:/etc/icinga2/features-enabled# ls -la
total 8
drwxr-xr-x 2 root root 4096 Feb  1 15:51 .
drwxr-x--- 9 nagios nagios 4096 Feb  7 12:47 ..
lrwxrwxrwx 1 root root 30 Feb  1 15:51 api.conf -> ../features-available/api.conf
lrwxrwxrwx 1 root root 34 Jan 31 16:55 checker.conf -> ../features-available/checker.conf
lrwxrwxrwx 1 root root 34 Feb  1 15:38 command.conf -> ../features-available/command.conf
lrwxrwxrwx 1 root root 36 Feb  1 15:39 ido-mysql.conf -> ../features-available/ido-mysql.conf
lrwxrwxrwx 1 root root 34 Jan 31 16:55 mainlog.conf -> ../features-available/mainlog.conf
lrwxrwxrwx 1 root root 39 Jan 31 16:55 notification.conf -> ../features-available/notification.conf
```

Abbildung 26 – Datei features-enabled



Dementsprechend befinden sich in der Datei features-available alle Features welche noch nicht aktiviert sind und aktiviert werden können:

```
root@icinga:/etc/icinga2/features-available# ls -la
total 80
drwxr-xr-x 2 root root 4096 Feb 1 15:51 .
drwxr-x--- 9 nagios nagios 4096 Feb 7 12:47 ..
-rw-r--r-- 1 root root 300 Feb 1 15:51 api.conf
-rw-r--r-- 1 root root 301 Feb 1 15:51 api.conf.orig
-rw-r--r-- 1 root root 131 Apr 5 2016 checker.conf
-rw-r--r-- 1 root root 158 Apr 5 2016 command.conf
-rw-r--r-- 1 root root 177 Apr 5 2016 compatlog.conf
-rw-r--r-- 1 root root 255 Apr 5 2016 debuglog.conf
-rw-r--r-- 1 root root 226 Feb 1 00:00 gelf.conf
-rw-r--r-- 1 root root 210 Apr 5 2016 graphite.conf
-rw-r--r-- 1 root root 311 Apr 5 2016 icingastatus.conf
-rw-r--r-- 1 nagios nagios 237 Feb 1 15:39 ido-mysql.conf
-rw-r--r-- 1 root root 528 Feb 1 00:00 influxdb.conf
-rw-r--r-- 1 root root 144 Apr 5 2016 livestatus.conf
-rw-r--r-- 1 root root 179 Apr 5 2016 mainlog.conf
-rw-r--r-- 1 root root 151 Apr 5 2016 notification.conf
-rw-r--r-- 1 root root 210 Apr 5 2016 opensdb.conf
-rw-r--r-- 1 root root 166 Apr 5 2016 perfdata.conf
-rw-r--r-- 1 root root 243 Apr 5 2016 statusdata.conf
-rw-r--r-- 1 root root 127 Apr 5 2016 syslog.conf
```

Abbildung 27 – Datei-features available

In der Datei repository.d sind alle Endpoints, Hosts und Zonen definiert:

```
root@icinga:/etc/icinga2/repository.d# ls -la
total 24
drwxr-xr-x 5 root root 4096 Feb 1 16:00 .
drwxr-x--- 9 nagios nagios 4096 Feb 7 12:47 ..
drwxr-x--- 2 nagios nagios 4096 Feb 3 08:53 endpoints
drwxr-x--- 4 nagios nagios 4096 Feb 3 09:08 hosts
-rw-r--r-- 1 root root 159 Apr 5 2016 README
drwxr-x--- 2 nagios nagios 4096 Feb 3 08:53 zones
```

Abbildung 28 – Datei repository.d

In der Datei Scripts sind die Scripts für die E-Mail-Benachrichtigung vorhanden:

```
root@icinga:/etc/icinga2/scripts# ls -la
total 16
drwxr-xr-x 2 root root 4096 Feb 1 15:43 .
drwxr-x--- 9 nagios nagios 4096 Feb 7 12:47 ..
-rwxr-xr-x 1 root root 384 Apr 5 2016 mail-host-notification.sh
-rwxr-xr-x 1 root root 437 Apr 5 2016 mail-service-notification.sh
```

Abbildung 29 – Datei scripts



Die vorher genannte Konfigurationssprache wird in jedem File verwendet. Der Vorteil daran ist das kein Vorwissen über irgendwelche Programmiersprachen benötigt wird. Darüber hinaus ist diese Sprache sehr einfach gestaltet was den Einstieg in die Konfiguration erleichtert.

Beispiel: Service „ping6“ unter Ubuntu

```
apply Service "ping6" {
    import "generic-service"

    check_command = "ping6"
    assign where host.address6
}
```

Abbildung 30 – Service "ping6" unter Ubuntu

Beispiel: Service „ping6“ unter Windows

```
apply service "ping6" {
    import "generic-service"

    check_command = "ping6-windows"
    assign where host.address6
}
```

Abbildung 31 – Service "ping6" unter Windows

Wie an diesem Beispiel zu erkennen ist funktioniert die Konfiguration unter den beiden Betriebssystemen gleich.

4.7.3 Hosts und Services

In Icinga2 werden **Hosts** angelegt welche überwacht werden sollen. Als Hosts zählen alle Geräte die im Netzwerk vorhanden sind (PC, Sensor, Router, Switch usw.).

Die Überwachungen werden **Service** genannt. Ein Service kann der Zustand des PCs (CPU-Auslastung, RAM-Auslastung usw.) oder spezifische Daten von Sensoren sein (Wahrscheinlichkeit für Regen von einem Niederschlagssensor, Temperatur eines Temperatursensors usw.). Diese Daten werden ausgewertet und von Icinga2 übernommen. Die Überwachung von spezifischen Daten erfolgt über **SNMP** und eine **MIB**.

Beispiel: PC überwacht sich selber

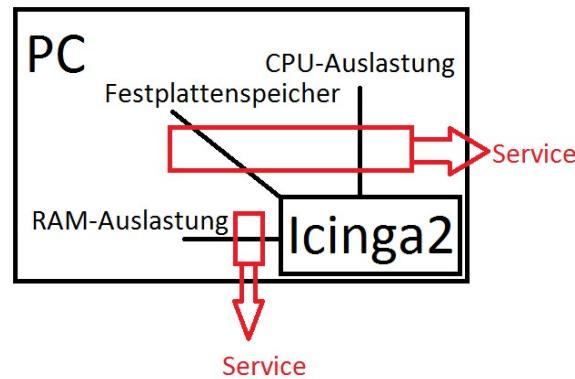


Abbildung 32 – PC überwacht sich selbst

Beispiel: PC überwacht einen Sensor

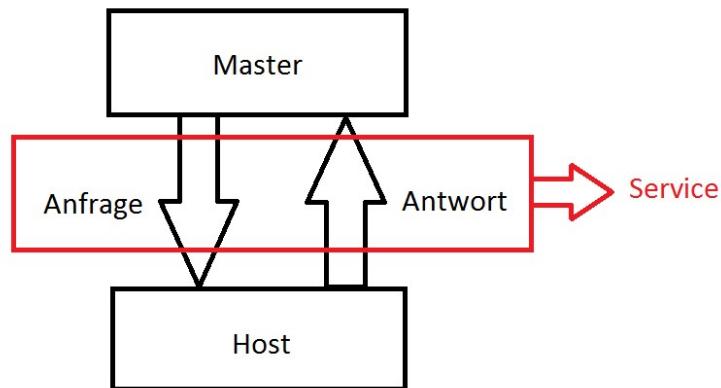


Abbildung 33 – PC überwacht Sensor

In den beiden Abbildungen sind sowohl der PC als auch der Master ebenfalls Hosts. Der Master kann entweder ein PC, ein Laptop oder auch ein Server sein. Natürlich können auch Virtuelle Maschinen als Hosts und Master verwendet werden.



Ein PC kann gleichzeitig sich selber und die Daten eines Sensors überwachen:

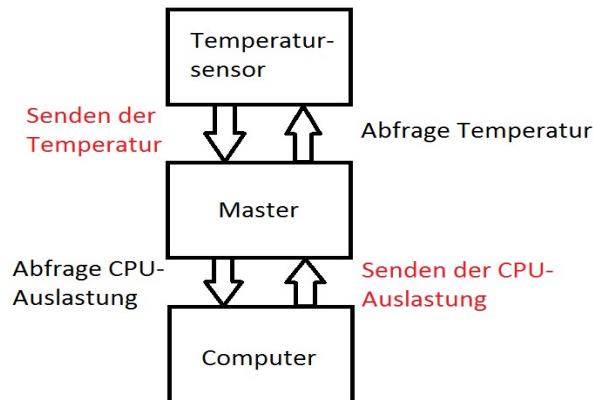


Abbildung 34 – PC überwacht sich selbst und einen Sensor

Anmerkung: Der Master und der Computer sind in diesem Fall dasselbe Gerät.

In einem System bestehend aus tausenden Hosts und Services können sogenannte **Satellite-PCs** eingesetzt werden: Sie stehen in einer **Zone** (bestehend aus Komponenten auf welche nur sie Zugriff haben), sammeln Daten und schicken diese dann zurück an den Master:

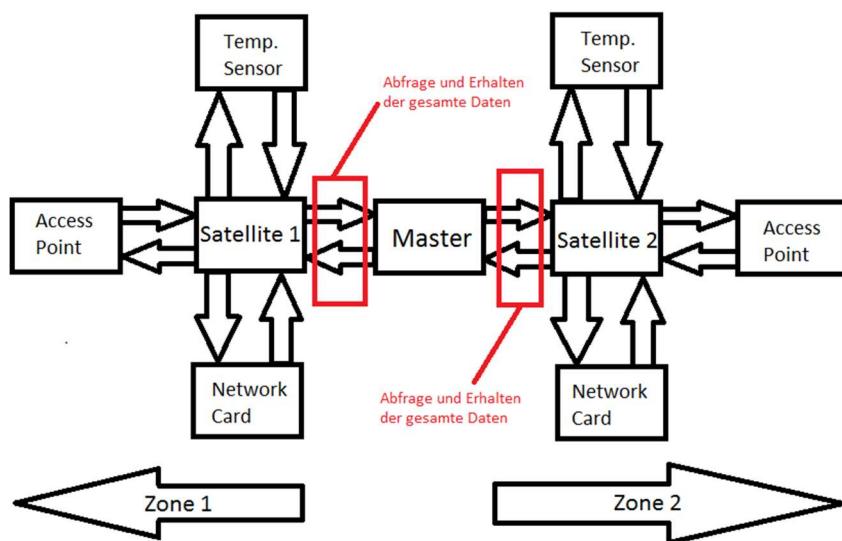


Abbildung 35 – Zonen

Die Zonen bestehen aus dem Satellite-PC und ihren dazugehörigen Komponenten welche sie überwachen sollen. Ein Satellite-PC fungiert daher wie ein Master in seiner Zone.



Der Vorteil von Satellite-PCs ist, dass der Master nicht die ganze Arbeit übernehmen muss und daher eine geringere CPU-Auslastung hat sowie nur die Satellite PCs überwachen muss. Darüber hinaus wird durch Satellite-PCs das Netzwerk weniger belastet, da der Master nicht zu allen Endgeräten kommunizieren muss sondern nur zu den Satellite-PCs.

4.7.4 Host- und Servicedefinitionen

Ein Host wird in der Datei hosts.conf im entsprechenden Dateipfad angelegt. Dafür wird unter Windows die Datei mit einem Doppelklick geöffnet (beim ersten Öffnen muss ein Editor ausgewählt werden) und unter Ubuntu wird die Datei mit einem Editor nach Wahl geöffnet, z.B.: **nano /etc/icinga2/conf.d/hosts.conf**.

Als Beispiel wurde ein Host unter einem Ubuntu PC angelegt. Dieser Host ist ein externer Sensor mit dem Namen Network-Card. Zuerst wird die Datei **generic-host** importiert um alle Standards für einen Host zu übernehmen. Das Attribut **address** steht für die IPv4-Adresse des Gerätes. Die anschließenden Zeilen sind dafür da um bei einer Änderung des Hostzustandes eine E-Mail an die Gruppe **icingaadmins** zu schicken.

```
object Host "Network-Card" {
    import "generic-host"

    address = "10.48.1.31"

    vars.notification["mail"] = {
        groups = [ "icingaadmins" ]
    }
}
```

Abbildung 36 – Beispiel Host

Das nächste Beispiel ist von einem Icinga2 Standardservice welcher vom Tool selbst angelegt und auch durchgeführt wird. Die Datei **generic-service** importiert die Standards für einen Service, das Check_Command ist der Ping4 was auf einen Ping auf IPv4-Ebene hinweist und das Attribut **assign where host.address** bedeutet dass dieser Service für jeden Host ausgeführt wird. Wird hier keine konkrete Adresse angegeben wird der Service automatisch auf jede IP-Adresse bezogen.



```
apply Service "ping4" {
    import "generic-service"

    check_command = "ping4"

    assign where host.address
}
```

Abbildung 37 – Beispiel Standardservice

Das folgende Beispiel ist die Definition eines Service über den externen Sensor vom Host-Beispiel. Hierbei kann entweder die IP-Adresse angegeben werden bei der der Service ausgeführt werden soll oder es kann der entsprechende Name des Hosts angegeben werden. Das check_command ist hierbei **SNMP**, was darauf hinweist, dass die Daten nur per SNMP vom Sensor erhalten werden können. Unter vars.snmp_oid wird eine MIB angegeben. Über diese MIB weiß der Sensor welchen Wert er nun zur Verfügung stellen muss. Die Variable vars.sla gibt an, wann der Check durchgeführt werden soll (24h, 7 Tage die Woche). Der Service nimmt nun den Zustand Warning bei 240 Volt an und den Zustand Critical bei 250 Volt. Dabei wird auch eine Änderungs-E-Mail geschickt da es sich um eine Änderung des Zustands des Service und des Hosts handelt.

```
object Service "Input Voltage" {

    import "generic-service"
    host_name = "Network-Card"
    check_command = "snmp"
    vars.snmp_oid = ".1.3.6.1.4.1.318.1.1.1.3.2.1.0"

    vars.sla = "24x7"

    vars.snmp_warn = 240
    vars.snmp_crit = 250
}
```

Abbildung 38 – Beispiel eigener Service



4.7.5 Host- und Servicezustände

Ein Host und ein Service können mehrere Zustände einnehmen:

Returncode	Servicestatus	Hoststatus
0	OK	UP
1	WARNING	UP
2	CRITICAL	DOWN
3	UNKNOWN	DOWN bzw. UNREACHABLE

Icinga2 weiß an sich nicht, ob ein Host oder ein Service überwacht wird. Dafür wird der sogenannte **Returncode** verwendet, welcher einen Wert zwischen 0 und 3 zurückliefert. Ob es sich nun um einen Service oder einen Host kann der Nutzer feststellen, indem er sich den Returncode anschaut und dem entsprechenden Host bzw. Service zuweist. In Icingaweb2 wird dies automatisch gemacht. Des Weiteren besitzt jeder Zustand eine eigene Farbe in Icingaweb2:

Farbe	Service	Host
Grün	OK	UP
Orange	WARNING	-
Rot	CRITICAL	DOWN
Violett	UNKNOWN	UNREACHABLE

4.7.6 Benachrichtigungssystem

Icinga2 besitzt ein sehr gutes Benachrichtigungssystem. Diese Benachrichtigungen können per E-Mail, SMS oder Voice-Calls durchgeführt werden. Das Prinzip dahinter ist wie folgt: Ändert sich der Returncode eines Hosts oder eines Service wird sofort eine entsprechende Benachrichtigung an die gewählten Personen geschickt (meistens an die Administratoren).

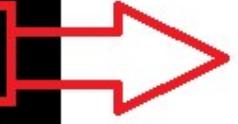
Wird in der Hostdefinition die untenstehende Konfiguration eingefügt wird eine E-Mail an die Administratorgruppe geschickt sobald sich der Zustand des Hosts oder der auf dem Host laufenden Services ändert:



```
object Host "Network-Card" {
    import "generic-host"

    address = "10.48.1.31"

    vars.notification["mail"] = {
        groups = [ "icingaadmins" ]
    }
}
```



E-Mail-Benachrichtigung

Abbildung 39 – Code für die E-Mail-Benachrichtigung

Der Icinga-Admin besitzt die E-Mail-Adresse da.wks.test@gmail.com:

```
object User "icingaadmin" {
    import "generic-user"

    display_name = "Icinga 2 Admin"
    groups = [ "icingaadmins" ]

    email = "da.wks.test@gmail.com"
}
```

Abbildung 40 – Konfiguration E-Mail-Adresse

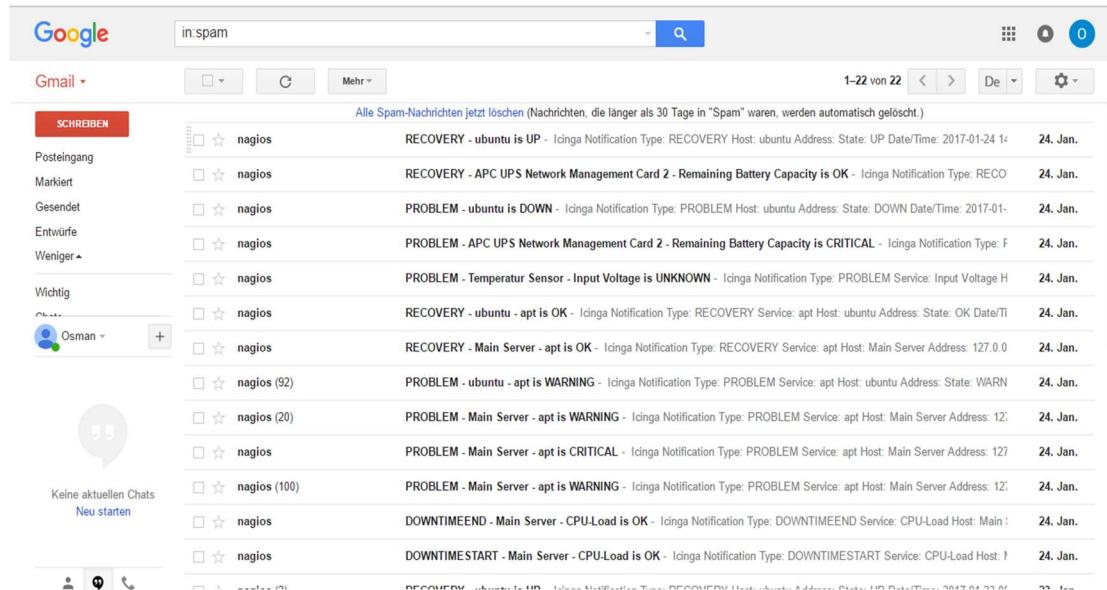


Abbildung 41 – Von Icinga2 geschickte E-Mails

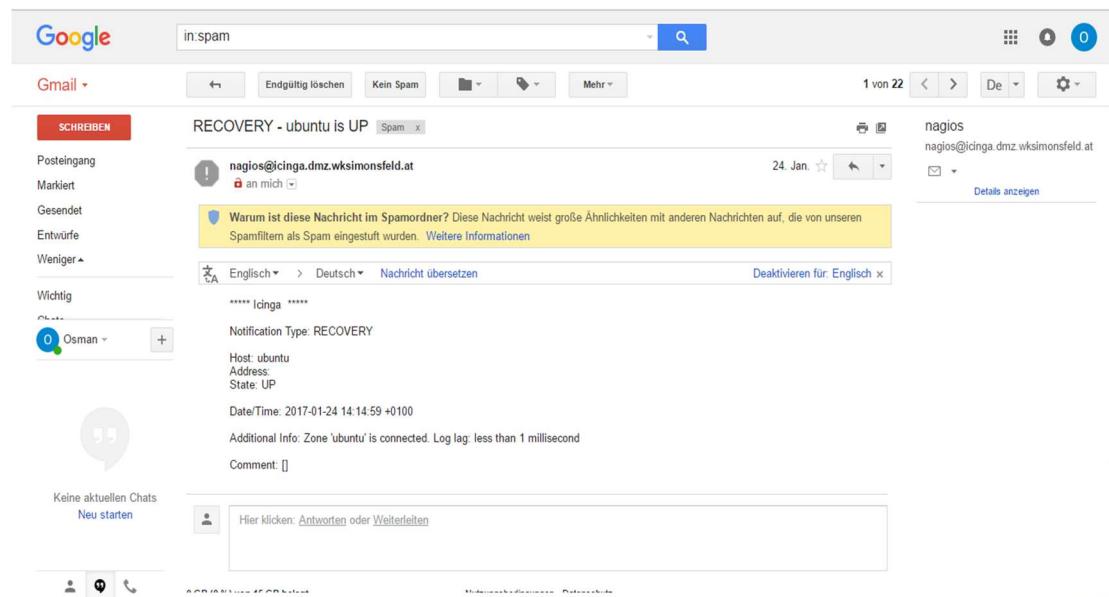


Abbildung 42 – Inhalt von Icinga2 geschickter E-Mail

Die SMS- und Voice-Call-Benachrichtigung wurde in unserer Diplomarbeit nicht benutzt da für diese Funktionen ein eigenes Modul benötigt wird.

4.7.7 Verteiltes Netzwerk aufbauen

Für die Anleitung „Verteiltes Netzwerk aufbauen“ siehe Anhang.



4.7.8 Icingaweb2

The screenshot shows the Icingaweb2 dashboard. On the left, a sidebar menu includes 'Dashboard', 'Problems', 'Overview', 'History', 'System', 'Configuration', and 'icingaweb'. The 'icingaweb' item is currently selected. The main content area has tabs for 'Current Incidents', 'Overdue', and 'Muted'. The 'Service Problems' tab is active, showing the message 'No services found matching the filter.' Below this is the 'Recently Recovered Services' section, which lists various services and their statuses. The list includes:

Time	Service	Message
6m 4s	master: apt	APT OK: 0 packages available for upgrade (0 critical updates).
16m 13s	ubuntu: apt	APT OK: 0 packages available for upgrade (0 critical updates).
Feb 3	ubuntu: disk /	DISK OK - free space: / 100064 MB (97% inode=97%).
Feb 3	ubuntu: disk	DISK OK - free space: / 100064 MB (97% inode=97%).
Feb 3	ubuntu: http	HTTP OK: HTTP/1.1 200 OK - 11595 bytes in 0.002 second response time
Feb 3	ubuntu: icinga	Icinga 2 has been running for 22 hours, 9 minutes and 58 seconds. Version: r2.6.1-1
Feb 3	ubuntu: load	OK - load average: 0.16, 0.10, 0.06
Feb 3	ubuntu: ping4	PING OK - Packet loss = 0%, RTA = 0.20 ms
Feb 3	ubuntu: ping6	PING OK - Packet loss = 0%, RTA = 0.21 ms
Feb 3	ubuntu: procs	PROCS OK: 147 processes

At the bottom right of the 'Recently Recovered Services' section is a 'Show More' link.

Abbildung 43 – Icingaweb2 Dashboard

In der obigen Abbildung ist die Startseite der GUI zu sehen – das sogenannte **Dashboard**. Dieses bietet eine Schnellübersicht über die letzten durchgeföhrten Services. Ebenfalls auf dem Dashboard zu sehen sind die **Service Problems** und die **Host Problems** welche auch links unter dem Reiter **Problems** zu finden sind.

The screenshot of the Icingaweb2 dashboard is annotated with red boxes and arrows. A red box labeled 'Problems' points to the 'Service Problems' section on the left. Another red box labeled 'Host Problems' points to the 'Host Problems' section on the left. A red box labeled 'Die letzten Services' points to the 'Recently Recovered Services' section on the right. Red arrows also point from the 'Problems' and 'Host Problems' boxes to their respective sections.

Abbildung 44 – Icingaweb2 Dashboard Erklärung



Unter **Overview** erhält man einen Überblick über alle Hosts und Services die in Icinga2 angelegt wurden:

The screenshot shows the Icingaweb2 interface for monitoring hosts. The left sidebar has a red box around the 'Overview' section, which includes links for 'Tactical Overview', 'Hosts', 'Services', 'Hostgroups', 'Servicegroups', 'Contactgroups', 'Contacts', 'Comments', and 'Downtimes'. The main content area shows a table of hosts with the following data:

Host	Status	Details
master	UP	since Feb 1 PING OK - Packet Loss = 0%, RTA = 0.05 ms
ubuntu	UP	since Feb 6 14:27 Zone 'ubuntu' is connected. Log lag: less than 1 millisecond
windows	UP	since Feb 1 Zone 'windows' is connected. Log lag: less than 1 millisecond

At the bottom of the main content area, there is a message: '0 row(s) selected' and '3 Hosts' with a green box around the number 3.

Abbildung 45 – Icingaweb2-Overview

Daneben kann noch ein Overview über die Services, Hostgroups usw. angezeigt werden.

Die anderen Reiter sind für diese Diplomarbeit von nicht so großer Bedeutung.

4.7.9 Graphite

Graphite ist ein Plugin von Icinga2 und kann zum grafischen Monitoring verwendet werden. Dabei können die Grafen entweder in Icingaweb2 (Kurve über die letzten 24 Stunden) oder im Graphite Browser betrachtet werden. Der Browser bietet dabei die Funktionen den Grafen von einem beliebigen Zeitraum wieder aufrufen zu können.

Graphite wurde mithilfe einer Installationsanleitung aus dem Internet installiert.



Der Graphite Browser – auch genannt **graphite-web** – sieht wie folgt aus:

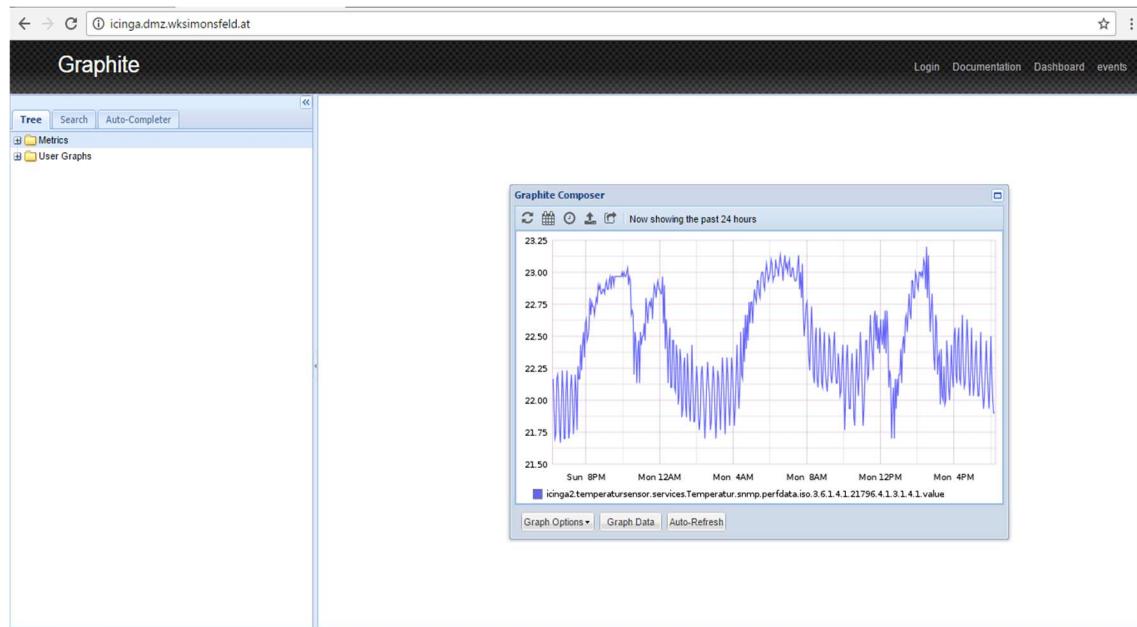


Abbildung 46 – Graphite-Web

Zu sehen ist der Verlauf eines Temperatursensors welcher in der WK Simonsfeld AG steht.

Um einen Grafen auszusuchen muss dieser unter **Metrics** (linke Spalte) ausgewählt werden.

Graphite kann aber auch direkt in Icingaweb2 eingebunden werden. In der unteren Abbildung ist derselbe Temperaturverlauf zu sehen – jedoch in Icingaweb2:

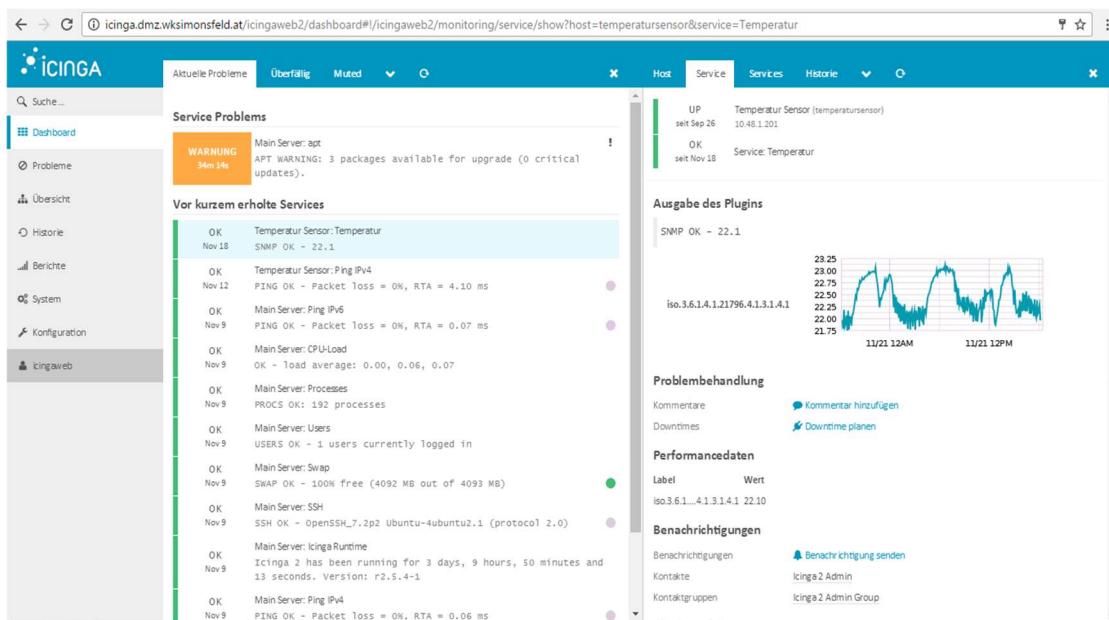


Abbildung 47 – Graphite-Plugin in Icingaweb2



Diese Anzeige läuft automatisch über die letzten 24 Stunden. Um einen präzisen Überblick über den Verlauf oder einzelne Punkte in der Grafik zu bekommen muss Graphite-Web benutzt werden.

Graphite ist jedoch nicht automatisch in Icingaweb2 eingebunden, wenn Graphite-Web installiert wird. Für die Einbindung wurde die folgende Website als Hilfestellung genommen:

<https://www.linuxfrickeln.de/visualisierung-von-perfomancedaten-in-der-icingaweb2-gui/>

4.8 Check_MK

4.8.1 Kurzbeschreibung



Abbildung 48 – Check_MK

Check_MK ist das zweite Programm welches für die Überwachung in Betracht gezogen wurde. Das Tool basiert auf **Nagios** und ist daher in vielen Dingen dazu kompatibel. In der **Raw_Edition** (Open Source Version) baut es auf dem bewährten Nagios-Kern auf, während es in der **Enterprise Edition** auf einem eigenen Kern, dem sogenannten **Check_MK Microcore** aufbaut [MKD17].



4.8.2 Funktionalität

4.8.2.1 Aufbau von Check_MK



Abbildung 49 – Check_MK Dashboard

- 1 Die Version Ihrer Check_MK-Installation. Ein Klick auf die Versionsnummer führt zu einer Aufstellung der Neuigkeiten dieser Version.
- 2 Die Gesamtübersicht (*Tactical Overview*) mit der Anzahl überwachter Hosts und Services, der jeweiligen Probleme und der Zahl derer, die noch unquittiert sind. Sie können die Ziffern anklicken und bekommen dann jeweils eine Auflistung. Probleme von Services, deren Host **DOWN** ist, werden hier nicht mitgezählt!
- 3 Ein interaktives Feld zur Suche nach Hosts und Services. Mehr dazu weiter unten...
- 4 Die Ansichten (*Views*) bieten Zugriff auf verschiedene Darstellungen von Hosts, Services und anderen Informationen aus dem aktuellen Monitoring.
- 5 Am unteren Rand der Seitenleiste finden Sie drei Knöpfe: bringt Sie zu einer Auswahl von verfügbaren Elementen für die Seitenleiste, zu Ihren persönlichen Einstellungen und mit melden Sie sich von der Check_MK-Oberfläche ab.
- 6 In der Kopfzeile jeder Seite finden Sie den Namen des angemeldeten Benutzers, seine Rolle und die Uhrzeit, zu der die Seite berechnet oder zuletzt aktualisiert wurde.
- 7 Die *Host-Statistik* zeigt die Anzahl von Hosts in den verschiedenen Zuständen - in Zahlen und in einer räumlich maßstabsgetreuen Grafik.
- 8 Die *Service-Statistik* zeigt analog die Services
- 9 Eine Auflistung aller Host-Probleme, die aktuell nicht quittiert sind. Ein Klick auf den Titel des Kastens bringt Sie zu einer vergrößerten Ansicht.
- 10 Analog die noch nicht quittierten Service-Probleme
- 11 Die Ereignisse des Monitorings der letzten vier Stunden



4.8.2.2 Hosts & Services

Genau wie Icinga2 arbeitet auch Check_MK mit Hosts und Services. Ein Host ist hier ebenfalls jedes physikalische Gerät mit einer IP-Adresse wie Server, Netzwerkgeräte, Messgeräte usw.

4.8.2.3 Hosts- und Servicedefinitionen

Zur Verwaltung der Hosts und Services wird das sogenannte **WATO** Konfigurationstool (eigenes Tool für die Konfiguration unter Check_MK) verwendet.

Abbildung 50 – Check_MK WATO

In der Übersicht wird der Reiter „New host“ ausgewählt. Anschließend muss der **Hostname**, **ein Alias und die IPv4-Adresse** angegeben werden. Für die IP-Adresse gibt es mehrere Möglichkeiten:

1. Keine IP-Adresse – der Hostname muss per DNS auflösbar sein
2. IP-Adresse angeben
3. Anstatt der IP-Adresse wird ein alternativer Hostname angegeben der per DNS auflösbar ist
4. *Host with dynamic DNS lookup during monitoring*



Darüber hinaus können noch Host-Tags gesetzt werden:

Setting	Value
Agent type	Check_MK Agent (Server) (Default value)
Criticality	Productive system (Default value)
Networking Segment	Local network (low latency) (Default value)
IP Address Family	IPv4 only (Default value)

Abbildung 51 – Check_MK Host tags

Die 3 wichtigsten Einstellungen für den **Agent Type** sind:

1. Check_MK Agent: Der Host soll über den CHECK_MK-Agenten überwacht werden
2. SNMP: Der Host soll über SNMP überwacht werden
3. No Agent: Solche Hosts werden ohne Agenten und nur mit aktiven Checks überwacht

Ein eigener Service muss in Check_MK selten selbst geschrieben werden. Das Tool besitzt eine **Serviceerkennung**, welche auf einem Host nach den interessantesten Werten Ausschau hält. Bei SNMP-Geräten werden zuerst die ersten beiden OIDs (sysDescr und sysObjectID) und je nach Wert etwa zehn weitere OIDs aufgerufen. Anhand dieser Ergebnisse und den über 600 mitgelieferten SNMP-Checkplugins entscheidet Check_MK dann, ob das Gerät dieses Plugin unterstützt. Anschließend folgt die wirkliche Erkennung, die gefundenen Plugins rufen per örtlich begrenzten SNMP-Walks genau die Daten ab die benötigt werden und ermitteln daraus die zu überwachenden Services.

Der Knopf **Full Scan** erzwingt einen SNMP-Scan und anschließendes Holen von frischen Daten via SNMP. Dadurch werden auch Services von ganz neuen Plugins gefunden, jedoch verursacht dieser Befehl meist eine Wartezeit.



Die Services können natürlich auch konfiguriert werden:

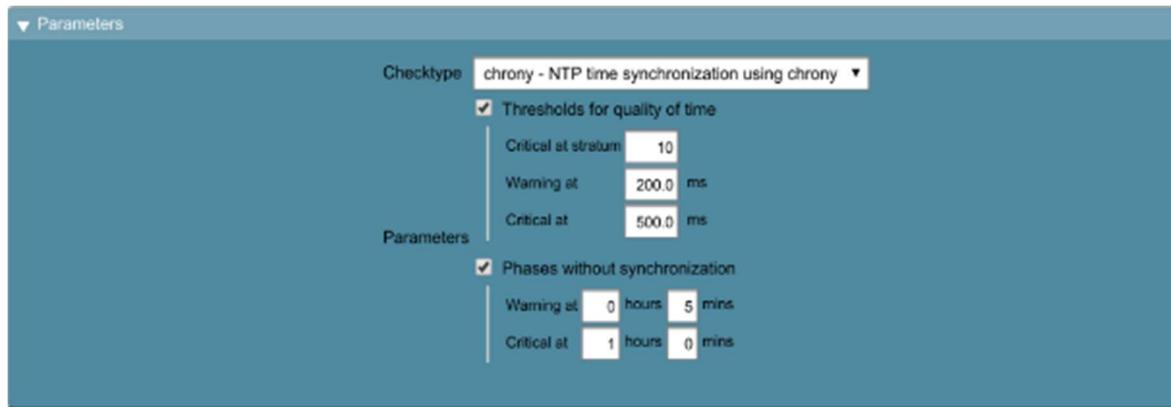


Abbildung 52 – Check_MK Manuelle Servicekonfiguration

4.8.2.4 Host- und Servicezustände

Die folgende Abbildung zeigt die Zustände welche die Hosts in Check_MK einnehmen können:

Zustand	Farbe	ID	Bedeutung
UP	grün	0	Der Host ist über das Netzwerk erreichbar (in der Regel heißt das, dass er auf PING antwortet.)
DOWN	rot	1	Der Host antwortet nicht auf Anfragen aus dem Netzwerk, ist nicht erreichbar.
UNREACH	orange	2	Der Weg zu dem Host ist aktuell für das Monitoring versperrt, weil ein Router oder Switch auf dem Weg dorthin ausgefallen ist.
PEND	grau		Der Host wurde frisch in die Überwachung aufgenommen und noch nie abgefragt. Genau genommen ist das aber kein Zustand.

Abbildung 53 – Check_MK Hostzustände

Neben dem Zustand hat ein Host noch weitere Attribute die vom Anwender konfiguriert werden:

- Name
- IP-Adresse
- Alias-Name
- ...



Darüber hinaus gibt es auch ein **Parentsyste** wie bei Icinga2. Dies heißt, dass ein Gerät von einem anderen Gerät überwacht werden kann und es somit sein „Kind“ ist. Wird diese Konfiguration verwendet kann der Zustand **UNREACH** bei einem Host vergeben werden. Wenn der Parent nicht mit seinem Kind kommunizieren kann wird der Zustand UNREACH vergeben – egal ob nur die Verbindung fehlerhaft ist oder das Kind DOWN ist.

Des Weiteren besitzt ein Host eine Menge an Services. Diese Services können (wie bei Icinga2) die Daten des PCs oder andere Werte welche über SNMP ausgetauscht werden sein. Auch hier gibt es mehrere Zustände:

Zustand	Farbe	ID	Bedeutung
OK	grün	0	Der Service ist vollständig in Ordnung. Alle Messwerte liegen im erlaubten Bereich.
WARN	gelb	1	Der Service funktioniert normal, aber seine Parameter liegen außerhalb des optimalen Bereichs.
CRIT	rot	2	Der Service ist ausgefallen, defekt
UNKNOWN	orange	3	Der Zustand des Services konnte nicht korrekt ermittelt werden. Der Monitoring-Agent hat fehlerhafte Daten geliefert oder die zu überwachende Sache ist ganz verschwunden.
PEND	grau		Der Service ist gerade in die Überwachung aufgenommen worden und es gibt noch keine Monitoring-Daten.

Abbildung 54 – Check_MK Servicezustände

4.8.2.5 Unterschied Ereignisse & Zustände

Um Check_MK verwenden zu können ist es wichtig den Unterschied zwischen Zuständen und Ereignissen zu verstehen.

Ein Ereignis ist etwas **Einmaliges** das zu **einem bestimmten Zeitpunkt** geschieht. Es ist somit nicht dauerhaft.

Ein Zustand ist eine **anhaltende Situation**, z.B. der Host hat den Zustand UP eingenommen. Um den aktuellen Zustand zu besitzen muss das Monitoring Tool diesen ständig abfragen (genau wie Icinga2).



Es gibt also die Wahl ob man mit Zuständen oder Ereignissen arbeiten möchte. Check_MK beherrscht zwar beide Disziplinen, benutzt jedoch immer dort wo die Wahl besteht das zustandsbasierte Monitoring. Der Grund liegt in den zahlreichen Vorteilen dieser Methode:

- Ein Fehler in der Überwachung wird sofort erkannt
- Das Monitoring kann selbst steuern, mit welcher Rate Zustände abgerufen werden
- Auch nach chaotischen Situationen (z.B. Stromausfällen) hat man immer einen zuverlässigen Gesamtzustand

4.8.2.6 Probleme, Alarme und Benachrichtigungen

Check_MK bezeichnet jeden Host und jeden Service der nicht den Zustand UP besitzt als ein **Problem**. Jedes Problem kann entweder **behandelt** oder **unbehandelt** sein. Jedes Problem ist also zuerst unbehandelt bis sich jemand dem Problem widmet und es anschließend bestätigt und somit in den Zustand behandelt wechseln kann (dieser Zustandswechsel ist meistens einem Wechsel in den UP Zustand gleichzustellen).

Wenn sich der Zustand eines Hosts oder einen Service ändert spricht Check_MK von einem **Alarm (Alert)**. Ein Alarm kann zu einer Benachrichtigung führen, da das Tool so voreingestellt ist, dass es im Falle eines Problems von einem Host oder einem Service jeden Kontakt dieses Objekts mit einer E-Mail benachrichtigt. Jedoch hängt diese Alarmierung von bestimmten Rahmenbedingungen ab. Check_MK benachrichtigt in den folgenden Fällen **NICHT**:

- Wenn Benachrichtigungen global in der **Master Control** deaktiviert werden
- Wenn Benachrichtigungen bei dem entsprechenden Host/Service ausgeschaltet wurden
- Wenn der jeweilige Zustand bei dem Host/Service für Benachrichtigungen deaktiviert wurde
- Wenn das Problem einen Host betrifft der DOWN oder UNREACH ist
- Wenn das Problem einen Host betrifft dessen Parent DOWN oder UNREACH ist
- Wenn der Host/Service unstetig ist
- Wenn sich der Host/Service in einer Wartungszeit befindet



Wenn diese Bedingungen erfüllt sind erzeugt der Monitoring-Kern eine Benachrichtigung, welche im zweiten Schritt jedoch eine Kette von Regeln durchläuft die ebenfalls definiert werden können.

Manchmal kommt es vor, dass sich der Zustand eines Hosts/Services ständig ändert. Um in diesem Fall viel zu viele Benachrichtigungen zu verhindern gibt es den Zustand **flapping**.

Wenn in einer bestimmten Zeit sich der Zustand nicht mehr ändert verschwindet der flapping Zustand und es kommt somit wieder zu normalen Benachrichtigungen.

Darüber hinaus können sogenannte Wartungszeiten eingeführt werden, um evtl. Kollegen mitzuteilen, dass Probleme in dieser bestimmten Zeit ignoriert werden. Diese Wartungszeiten werden verwendet um auf dem entsprechenden Host Änderungen durchführen zu können.

4.8.2.7 Timeperiods

Zeitperioden definieren regelmäßige, wöchentlich wiederkehrende Zeitbereiche, die an verschiedenen Stellen in der Konfiguration des Monitorings zum Einsatz kommen.

Standardmäßig ist die Periode **24x7** eingestellt, welche einfach **alle Zeiten** einnimmt. Es können aber z.B. **workhours** von 8:00 – 17:00 eingestellt werden.

Wichtige Bereiche in denen Timeperiods zum Einsatz kommen:

- Begrenzung der Zeiten, innerhalb derer benachrichtigt wird
- Begrenzung der Zeiten, innerhalb derer Checks durchgeführt werden
- Servicezeiten für die Berechnung von Verfügbarkeiten
- Zeiten, innerhalb derer bestimmte Regeln in der Event Console greifen

4.8.2.8 Checkintervall, Checkversuche und Checkperiode

Das Ausführen von Checks wird in Intervallen durchgeführt. Standardmäßig ist dieses Intervall auf 1 Minute gestellt → jeder Check wird 1-mal pro Minute ausgeführt. Natürlich kann diese Zeit verändert werden. Eine Änderung kann mehrere Gründe haben:

- Auf einen längeren Wert, um CPU-Ressourcen auf Server und Zielsystem zu sparen
- Auf einen kürzeren Wert, um schneller Alarne zu bekommen und Messdaten in einer höheren Auflösung zu sammeln



Durch Definition einer anderen Checkperiode als 24x7 kann das Ausführen von Checks in bestimmten Zeitfenstern unterbrochen werden. Der Zustand des Service wird dann nicht mehr aktualisiert und als **stale (veraltet)** angezeigt. Durch diese Methode kann ein großer Check definiert werden: Wird das Intervall auf 24h und die Checkperiod auf den Zeitraum 15:30 – 15:31 gesetzt wird der Check wirklich nur in diesem geringen Zeitraum durchgeführt.

Mithilfe der sogenannten Checkversuche kann definiert werden wie „wichtig“ ein Service ist. Wird die **max check attempts** auf 5 gestellt wird, wenn der Service eigentlich **CRIT** sein sollte noch keine Benachrichtigung geschickt, sondern er wird 5-mal auf seinen Zustand getestet. Ist beim 5. Mal der Zustand immer noch CRIT wird erst dann eine Benachrichtigung ausgesendet. Ein Service der sich in einem Zwischenzustand befindet, sprich **nicht OK** aber die maximalen Versuche noch nicht erreicht hat springt in einen **Soft State (weichen Zustand)**.

4.8.2.9 Graphing

Check_MK bietet grundsätzlich eine Graphingfunktion in der Basisinstallation an. Dieses Plugin zeichnet die entsprechenden Daten über einen Zeitraum von einstellbar vier Jahren auf. Es kann somit nicht nur auf die letzten Werte, sondern auch auf „historische“ Werte des Monitorings zugegriffen werden. Auf dieses grafische Monitoring kann über die GUI zugegriffen werden und es gibt **3 Arten** davon.

Das **Perf-O-Meter** liefert direkt in der Host- und Servicetabelle ein optischer Vergleich gezogen werden. Allerdings beschränken sich diese wegen Platzgründen meist auf eine einzelne ausgewählte Metrik.

State	Service	Icons	Status detail	Age	Checkint	Perf-O-Meter
OK	Filesystem /		OK - 7.3% used (439.90 MB of 5.91 GB), trend: -1.04 B / 24 hours	2015-02-05 09:06:13	50 sec	<div style="width: 7.3%; background-color: #28a745; height: 10px;"></div> 7.3%
OK	Filesystem /boot		OK - 3.7% used (35.87 of 981.55 MB), trend: 0.00 B / 24 hours	2015-02-05 09:06:13	50 sec	<div style="width: 3.7%; background-color: #28a745; height: 10px;"></div> 3.7%
OK	Filesystem /home		OK - 41.6% used (63.53 of 152.57 GB), trend: +0.00 B / 24 hours	2015-02-05 09:06:13	50 sec	<div style="width: 41.6%; background-color: #28a745; height: 10px;"></div> 41.6%
OK	Filesystem /opt		OK - 3.3% used (67.00 MB of 1.97 GB), trend: 0.00 B / 24 hours	2015-02-05 09:06:13	50 sec	<div style="width: 3.3%; background-color: #28a745; height: 10px;"></div> 3.3%
OK	Filesystem /opt/omd		OK - 11.2% used (1.77 of 15.75 GB), trend: 0.00 B / 24 hours	2015-02-05 09:06:13	50 sec	<div style="width: 11.2%; background-color: #28a745; height: 10px;"></div> 11.2%
OK	Filesystem /srv		OK - 9.7% used (97.66 of 1007.90 MB), trend: +8.55 MB / 24 hours	2015-02-05 09:06:13	50 sec	<div style="width: 9.7%; background-color: #28a745; height: 10px;"></div> 9.7%
OK	Filesystem /tmp		OK - 8.4% used (84.56 of 1007.90 MB), trend: -26.13 kB / 24 hours	2015-02-05 09:06:13	50 sec	<div style="width: 8.4%; background-color: #28a745; height: 10px;"></div> 8.4%
OK	Filesystem /usr		OK - 38.7% used (6.09 of 15.75 GB), trend: 0.00 B / 24 hours	2015-02-05 09:06:13	50 sec	<div style="width: 38.7%; background-color: #28a745; height: 10px;"></div> 38.7%
OK	Filesystem /var		OK - 11.5% used (924.22 MB of 7.87 GB), trend: -1.31 MB / 24 hours	2015-02-05 09:06:13	50 sec	<div style="width: 11.5%; background-color: #28a745; height: 10px;"></div> 11.5%

Abbildung 55 – Check_MK Graphing-Dashboard



Eine Grafik über einen Service bezogen auf die Zeit erhält man per einen Klick auf das Graphen-Icon.

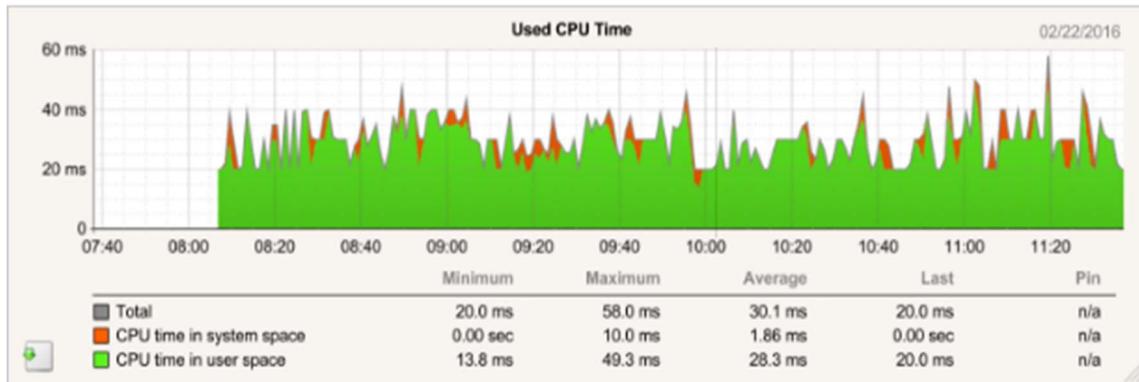


Abbildung 56 – Check_MK Graphing Beispiel

In den Host-/Servicedetails gibt es zudem eine Tabelle mit den aktuellen präzisen Messwerten für alle Metriken:

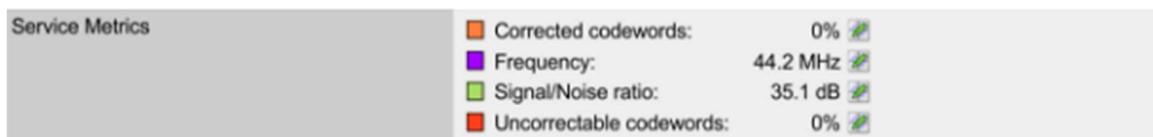


Abbildung 57 – Check_MK Graphing Messwerte

4.8.3 Icinga2 oder Check_MK?

Nachdem diese beiden Tools ausprobiert wurden wurde eine Entscheidung gefällt: **Icinga2**.

Dies hat mehrere Gründe: Zum einen scheint Icinga2 um einiges benutzerfreundlicher sowie moderner als Check_MK zu sein. Andererseits war Check_MK eine interne Idee welche gar nicht an die Firma weitergegeben wurde, da dies nur passiert wäre, wenn Check_MK um einiges besser gewesen wäre als Icinga2.

Natürlich könnte diese Überwachung auch mit Check_MK realisiert werden, jedoch hat es sich nicht als ausreichend viel besser als Icinga2 erwiesen und wurde somit nicht an die Firma weitergeleitet.



4.9 TeamViewer



Abbildung 58 - TeamViewer

Teamviewer ist ein Programm zur Herstellung von Remote Desktop Verbindungen. Die Software arbeitet als Online-Dienst durch Firewalls und NAT sowie Proxy-Server hindurch. Teamviewer funktioniert aber nur mit dem IPv4 Protokoll. Das Programm ist mit Windows, OS X, verschiedenen Linux Distributionen, Android und iOS kompatibel wobei vom Windows Phone aus nur gesteuert werden kann.

Darüber hinaus bietet das Tool seit Version 5 eine Audio- und Videoübertragung über IP-Telefonie sowie einer Anwendungsauswahl, mit welcher nur spezielle Anwendungen dargestellt werden [WIK17].

4.9.1 Verbindungsaufbau

Beim Aufbau einer Verbindung überprüft Teamviewer welche Verbindungsarten bei der bestehenden Konfiguration von Sicherheitsmaßnahmen hergestellt werden können (Firewalls, Proxy-Server usw.). In 70% der Fälle kann nach der Aushandlung der Verschlüsselung über die Teamviewer-Server eine direkte Verbindung (über TCP oder UDP) zwischen den beiden Endpunkten eine Verbindung hergestellt werden.



4.9.2 Datenübertragung

Die Datenübertragung erfolgt nicht über die Teamviewer-Server sondern in der portablen Version des Programms stimmt der Anwender zu, dass „nicht personenbezogene Daten“ gesammelt werden dürfen.

Das Programm nutzt eine Verschlüsselung auf der Basis des RSA-Kryptosystems und einer AES-Verschlüsselung mit 256 Bit. Bei einer direkten Verbindung im LAN wird eine symmetrische Verschlüsselung verwendet. Teamviewer behauptet von sich selber das keine **Man-in-the-Middle** Angriffe möglich sein sollen.

Dieses Tool fand seinen Nutzen in der Verbindung zu Windows Rechnern in dieser Diplomarbeit. Soll eine Verbindung zu einem Windows-PC hergestellt werden muss zunächst auf dem Windows-PC sowie auf dem eigenen PC Teamviewer installiert werden.

Anschließend muss Teamviewer auf dem Windows-PC so konfiguriert werden, dass eine ID und ein Standardpasswort als Login abgefragt werden (das Standardpasswort wird benutzt um sich ständig mit dem PC verbinden zu können; normales Passwort ändert sich bei jeder Sitzung).

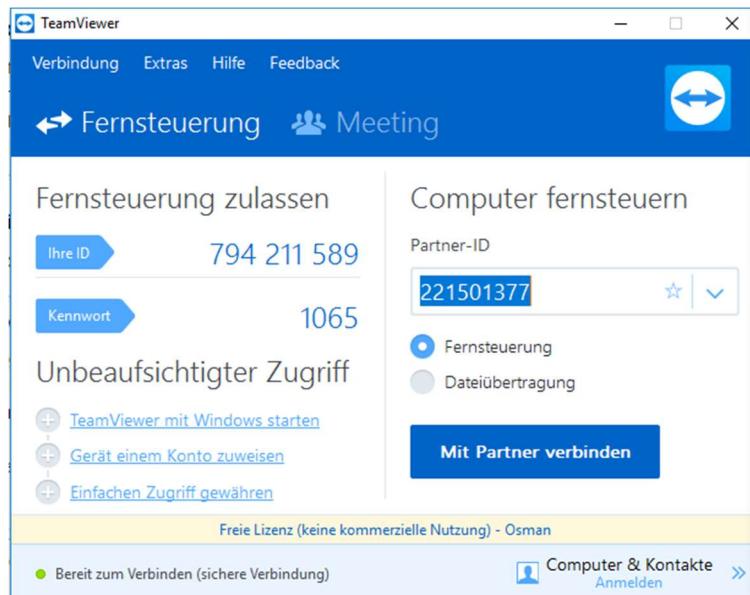


Abbildung 59 – TeamViewer Konfiguration

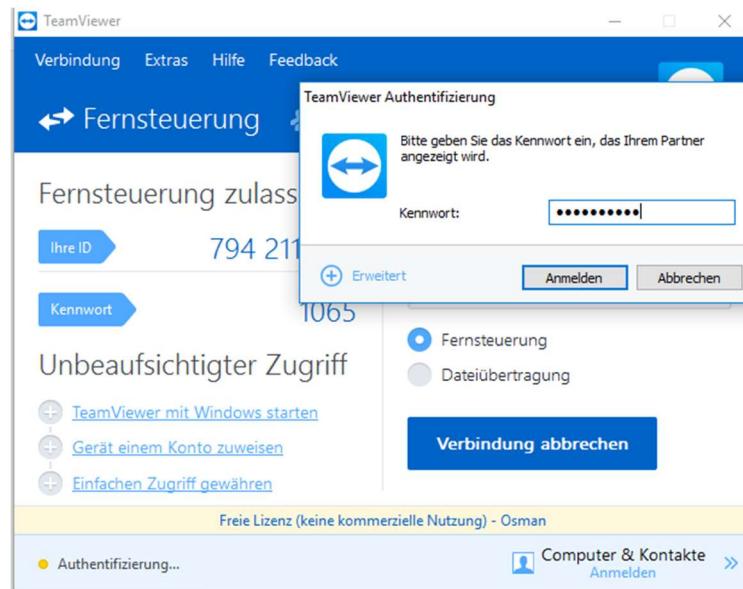


Abbildung 60 – TeamViewer Verbindungsauflaufbau

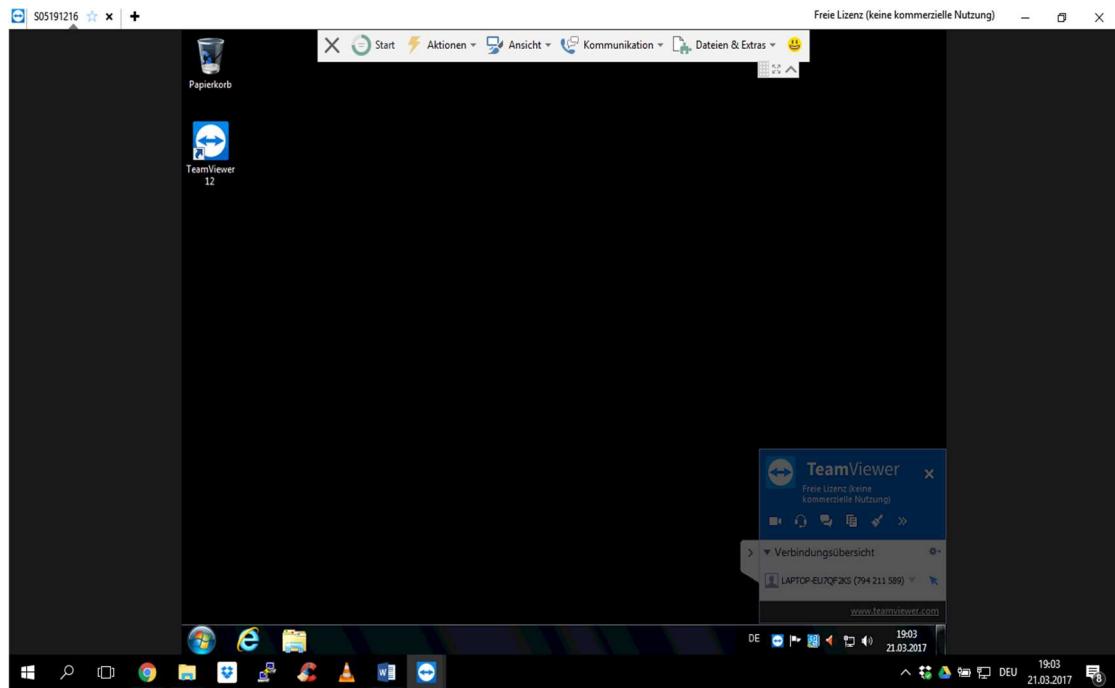


Abbildung 61 – TeamViewer Remote Ansicht

Nun ist eine Verbindung zum Windows-PC hergestellt und er kann wie ein normaler Computer genutzt werden.



5 Testumgebung

5.1 Sensorüberwachung

5.1.1 Aufgabenstellung

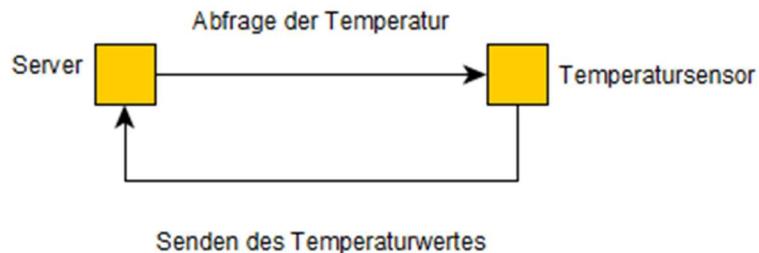


Abbildung 62 - Testumgebung 1

Das Ziel des ersten Milestones war es einen Temperatursensor von einem Server aus zu überwachen. Dabei soll konkret der Temperaturwert in einem bestimmten Intervall (z.B.: jede Minute) in Icingaweb2 ausgegeben werden. Wird ein Schwellenwert überschritten soll eine E-Mail-Benachrichtigung ausgesendet werden. Darüber hinaus soll sich der Server selber überwachen.

Der Server ist eine virtuelle Maschine basierend auf Ubuntu 16.04, der Temperatursensor ist der **HWG-STe** Sensor von HW-Group.



5.1.2 Einrichten der Virtuellen Maschine

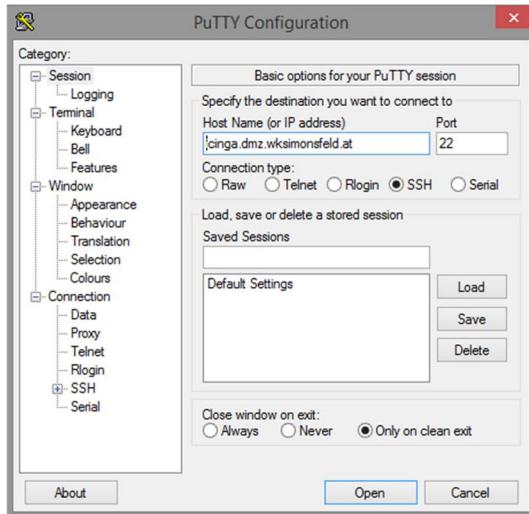


Abbildung 63 – Putty Verbindung zu Hauptserver

Zu Beginn wurde ein Verbindungstest mit Putty zur virtuellen Maschine durchgeführt, welcher negativ verlief. Als mit dem Betreuer der Firma geredet wurde, wurde der Host in ein anderes Subnet weitergeleitet. Ein anschließender Verbindungstest verlief erfolgreich.

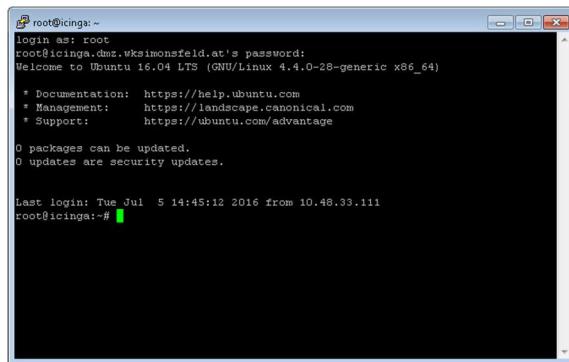


Abbildung 64 – SSH-Verbindung zu Hauptserver (Konsole)

Darauffolgend wurde Icinga2 und die GUI Icingaweb2 auf dem Server installiert.



5.1.3 Testsensor



Abbildung 65 – Temperatursensor

Der Testsensor ist der HWG-STE Temperatursensor von HW-Group. Die erfassten Temperaturwerte des Sensors können per SNMP abgefragt werden und in Verbindung mit Icinga2 in Icingaweb2 angezeigt werden (HW-Group, 2017).

5.1.4 Einbinden des Testsensors in Icinga2

Zunächst wurde der Testsensor in Icinga2 eingebunden. Hierfür wurde ein neuer Host in der Datei hosts.conf (im Icinga2 Verzeichnis: /etc/icinga2/conf.d) erstellt mit dem Namen Temperatursensor, dem Display-Namen „Temperatur Sensor“ und der IPv4-Adresse 10.48.1.201.

```
object Host "Temperatursensor" {
    import "generic-service"
    display_name = "Temperatur Sensor"
    address = "10.48.1.201"
}
```

Abbildung 66 – Hostdefinition Temperatursensor



5.1.5 Erster Test

In Icingaweb2 wurde der Sensor nicht als Host angezeigt. Stattdessen erschien unter System -> Monitoring Health die Warnung „Backend Icinga2 is not running“. Dies bedeutet, dass Icinga2 aufgrund eines Fehlers nicht mehr starten kann.

5.1.6 Fehlermeldung

Als Versuch wurde Icinga2 im Terminal zu starten (`/etc/init.d/icinga2 start`) kam es ebenfalls zu einer Fehlermeldung. Nach einer Fehlersuche wurden zwei Lösungsansätze gefunden:

1. Der Icinga Daemon funktioniert nicht
2. Die Datenbank Icinga2-ido-mysql ist nicht aktiviert

Obwohl der Icinga Daemon neu gestartet und die Datenbank erneut aktiviert wurde war der Fehler immer noch vorhanden, daher kam der Entschluss den Server neu aufzusetzen. Dies wurde vom Betreuer übernommen und anschließend wurden die beiden Programme erneut installiert sowie der Testsensor erneut eingebunden. Beim Starten von Icingaweb2 war das Problem trotzdem vorhanden.

In der Datei `startup.log` war jedoch ein Fehlerbericht vorhanden. In diesem Bericht war der Fehler ausführlich beschrieben: bei der Host-Definition ist ein Fehler aufgetreten und daher konnte Icinga2 nicht starten. Als dieser Fehler behoben wurde konnte Icinga2 problemlos starten und der „Backend Icinga is not running“ Fehler verschwand aus Icingaweb2.



5.1.7 Überwachung

Anschließend wurde noch ein Service namens „temp“ geschrieben welcher den Host über SNMP nach den Temperaturwerten abfragt.

```
apply Service "temp" {
    import "generic-service"
    check_command = "snmp"
    vars.snmp_oid = "1.3.6.1.4.1.21796.4.1.3.1.4.1"
    assign where host.address = "10.48.1.201"
}
```

Abbildung 67 – Servicedefinition Temperatursensor

Der Befehl „check_command“ & „vars.snmp_oid“ fragt den Sensor per SNMP nach den Temperaturwerten ab. Die Zeile „assign where host.address == „10.48.1.201““ weist den Service dem Temperatursensor zu.

Ein anschließender Funktionstest war erfolgreich und die laufenden Abfragen können im Icingaweb2 betrachtet werden.



5.1.8 Graphite

Um die Überwachung der Services grafisch zu veranschaulichen wurde das Plugin Graphite verwendet. Dieses wurde direkt in Icingaweb2 importiert um nicht ständig auf Graphite-Web wechseln zu müssen um den Graphen zu sehen.

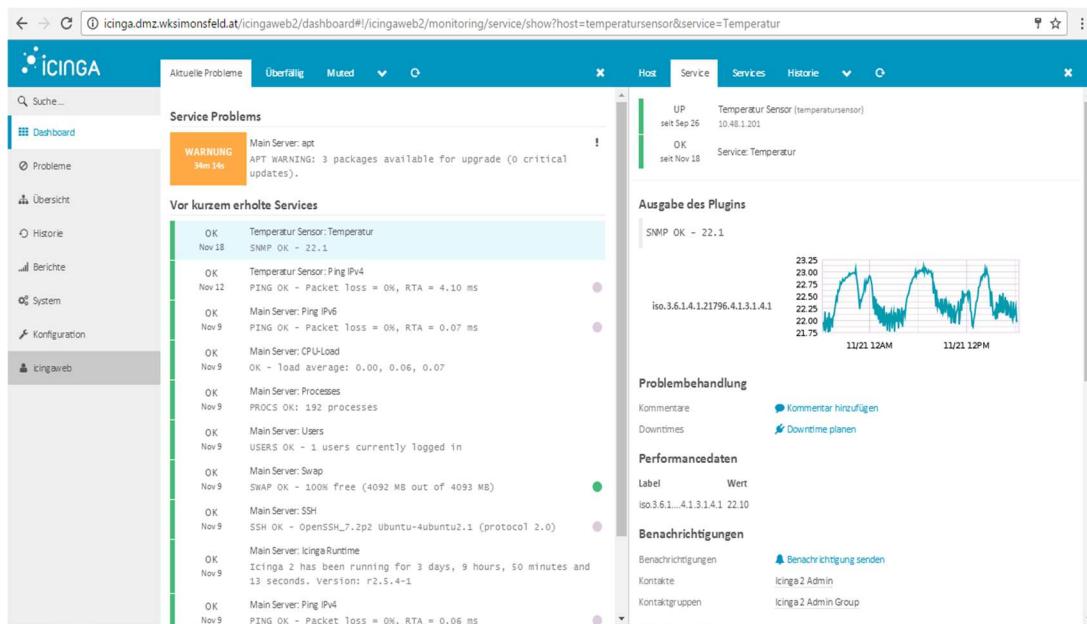


Abbildung 68-Graph Temperatursensor

5.1.9 E-Mail-Benachrichtigungen

Um die E-Mail Benachrichtigungen von Icinga2 zu aktivieren muss in der Datei "users.conf" bei einem User (in diesem Fall beim Admin) eine E-Mail Adresse eingegeben werden:
da.wks.test@gmail.com.



```
object User "icingaadmin" {
    import "generic-user"

    display_name = "Icinga 2 Admin"
    groups = [ "icingaadmins" ]

    email = "da.wks.test@gmail.com"
}
```

Abbildung 69 – E-Mail-Benachrichtigung Code Milestone 1

Nachdem in dem entsprechenden Service jeweils eine Grenze für eine Warnung und einen kritischen Zustand definiert wurde schickt Icinga2 automatisch vom Server aus eine E-Mail wenn einer dieser beiden Werte erreicht wird. Zuvor muss jedoch noch postfix installiert werden um eine Art Mailserver herzustellen.

Um die Funktionsweise der E-Mail-Benachrichtigungen zu testen wurde der Warning-Wert auf 20 gestellt da im Lagerraum (dort wo sich der Temperatursensor befindet) ständig eine Temperatur über 20°C herrscht (Klimaanlage).

```
apply Service "Temperatur" {
    import "generic-service"

    check_command = "snmp"
    vars.snmp_oid = "1.3.6.1.4.1.21796.4.1.3.1.4.1"

    vars.sla = "24x7"

    vars.snmp_warn = 20
    vars.snmp_crit = 30

    assign where host.address == "10.48.1.201"
}
```

Abbildung 70 – Servicedefinition Temperatursensor komplett



The screenshot shows the Icinga web interface with the following details:

- Service Problems:**
 - WARNING** 0m 12s Soft 1/5: Temperatur Sensor: Temperatur. SNMP WARNING - "22".
 - WARNING** 46m 40s: Main Server: apt. APT WARNING: 3 packages available for upgrade (0 critical updates).
- Vor kurzem erholte Services:**
 - OK Nov 12 Temperatur Sensor: Ping IPv4
 - OK Nov 9 Main Server: Ping IPv6
 - OK Nov 9 Main Server: CPU-Load
 - OK Nov 9 Main Server: Processes
 - OK Nov 9 Main Server: Users
 - OK Nov 9 Main Server: Swap
 - OK Nov 9 Main Server: SSH
 - OK Nov 9 Main Server: Icinga Runtime
 - OK Nov 9 Main Server: Ping IPv4
- Ausgabe des Plugins:** SNMP WARNING - "22". A graph titled 'iso.3.6.1.4.1.21796.4.1.3.1.4.1' shows temperature data over time, with values fluctuating between 21.75 and 23.25.
- Problembehandlung:** Buttons for 'Bestätigen', 'Kommentare', and 'Downtimes'.
- Performancedaten:** Label: iso.3.6.1...4.1.3.1.4.1, Wert: 22.00.
- Benachrichtigungen:** Buttons for 'Benachrichtigungen' and 'Bestätigung senden'. A note states: 'Für dieses Problem wurde keine Benachrichtigung gesendet.'

Abbildung 71 – E-Mail-Benachrichtigung Test

Gleichzeitig ist der Service nun im sogenannten Soft-Zustand und bleibt in diesem für die nächsten 4 Abfragen. Falls nach der 5. Abfrage immer noch der Zustand Warnung vorhanden ist wird eine E-Mail gesendet:

The email message content is as follows:

PROBLEM - Temperatur Sensor - Temperatur is WARNING

nagios@icinga.dmx.wksimonsfeld.at 18:24 (vor 3 Minuten)

an mich

Warum ist diese Nachricht im Spamordner? Diese Nachricht weist große Ähnlichkeiten mit anderen Nachrichten auf, die von unseren Spamfiltern als Spam eingestuft wurden. [Weitere Informationen](#)

Englisch > **Deutsch** **Nachricht übersetzen** **Deaktivieren für: Englisch**

***** Icinga *****

Notification Type: PROBLEM

Service: Temperatur
Host: Temperatur Sensor
Address: 10.48.1.201
State: WARNING

Date/Time: 2016-11-21 18:24:22 +0100

Additional Info: SNMP WARNING - "22.3"

Comment: []

Abbildung 72 – E-Mail-Benachrichtigung Temperatursensor



5.2 Verteiltes Netzwerk

5.2.1 Aufgabenstellung

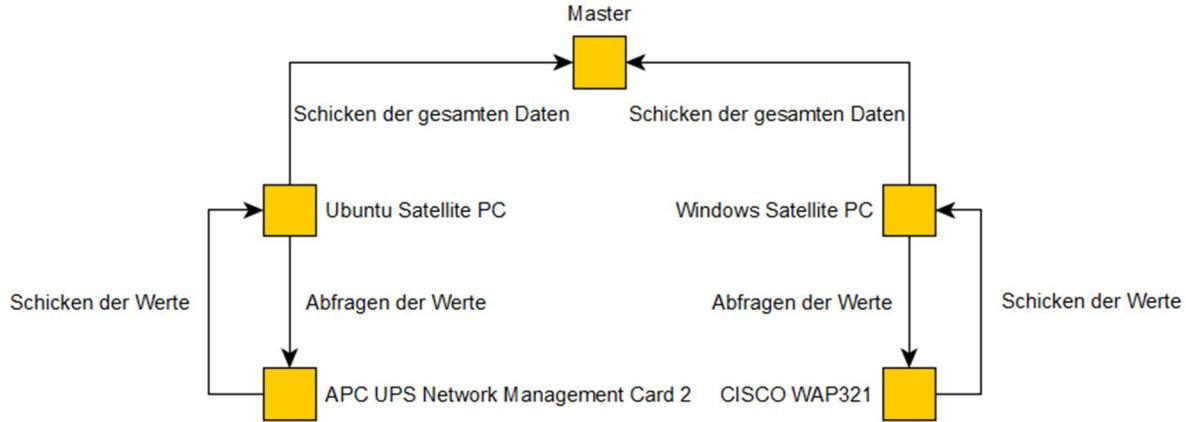


Abbildung 73 - Testumgebung 2

Die Aufgabe dieses Milestones war es ein verteiltes Netzwerk aufzubauen. Dafür wurde der von Milestone 1 verwendete Server als Master und zwei weitere Satellite PCs (1x Windows 7; 1x Ubuntu 16.04) als Slaves verwendet. Die beiden PCs konnten jedoch nicht sofort vom Betreuer zur Verfügung gestellt werden, daher wurde ein „Vorprojekt“ durchgeführt.

5.2.2 Vorprojekt

Dieses Vorprojekt wurde auf die wichtigsten Teile beschränkt: Ein PC soll einen anderen PC überwachen. Dafür wurden zwei PCs über einen Switch in ein gemeinsames Netzwerk gehängt um die Verbindung zwischen ihnen zu gewährleisten. Der Master ist ein PC basierend auf Ubuntu 16.04, da nur ein Ubuntu Rechner der Master sein kann. Der Slave ist ein Windows PC da Master und Slave Konfiguration unter Ubuntu 16.04 mehr oder weniger gleich abläuft. Darüber hinaus kann ein Windows PC nur als Slave verwendet werden. Anschließend wurde – wie in der Anleitung „Verteiltes Netzwerk in Icinga2“ zu lesen ist – ein verteiltes Netzwerk aufgebaut.



Das GUI sah dann wie folgt aus:

Services :: Icinga Web - Mozilla Firefox

localhost/icingaweb2/monitoring/list/hosts

Hosts

2 Hosts: 2

Search...

UP for 13m 12s master (2 unhandled services)
PING OK - Packet loss = 0%, RTA = 0.08 ms

UP for 5m 1s windows Zone 'windows' is connected. Log lag: less than 1 millisecond

0 row(s) selected

25 Sort by Hostname

Abbildung 74 – Verteiltes Netzwerk Vorprojekt

Unter Hosts ist nun auch der Windows PC zu sehen. Seine Services werden von ihm selbst gecheckt und anschließend an den Master weitergeleitet.

Services :: Icinga Web - Mozilla Firefox

localhost/icingaweb2/monitoring/list/hosts#/icingaweb2/monitoring/list/services?host=window

Hosts

2 Hosts: 2

Services

21 Services: 2 19

Search...

host = windows & service_state = 0

Sort by Service Name

25

UP for 13m 45s master (2 unhandled services)
PING OK - Packet loss = 0%, RTA = 0.07 ms

UP for 6m 34s windows Zone 'windows' is connected. Log lag: less than 1 millisecond

OK for 6m 12s windows:disk
DISK OK - free space:Total 786012MB (83%): C:\ 784265 MB (84%): D:\ 1747 MB (12%)

OK for 6m 22s windows:disk C:
DISK OK - free space:Total 786012MB (83%): C:\ 784265 MB (84%): D:\ 1747 MB (12%)

OK for 6m 22s windows:icinga
Icinga 2 has been running for 9 minutes and 40 seconds.
Version: v2.4.4

OK for 6m 20s windows:load
LOAD OK 7.05717%

OK for 6m 12s windows:ping4
PING OK RTA: 0ms Packet loss: 0%

OK for 6m 12s windows:ping5
PING OK RTA: 0ms Packet loss: 0%

OK for 6m 20s windows:procs
PROCS OK 102

OK for 6m 20s windows:swap
SWAP OK - 67.22% free

OK for 6m 20s windows:users
USERS OK 1 User(s) logged in

Abbildung 75 – Services Verteiltes Netzwerk Vorprojekt



5.2.3 Inbetriebnahme der PCs

Anschließend wurden vom Betreuer die beiden PCs zur Verfügung gestellt. Zunächst wurde auf beiden PCs Icinga2 installiert. Danach wurde ein Pingtest durchgeführt um zu testen, ob sich alle drei Komponenten gegenseitig anpingen können.

5.2.4 Ubuntu Setup

Zunächst wurde Icinga2 auf dem Ubuntu Satellite PC installiert (selbe Installation wie auf dem Hauptserver).

Anmerkung: Icingaweb2 wurde nicht installiert! (Daten werden daher nur in Icingaweb2 des Masters gesehen; wie gewünscht).

5.2.5 Windows Setup

Anschließend wurde Icinga2 auf dem Windows Satellite PC installiert. Hierfür wurde auf <http://packages.icinga.org/windows/> die Version Icinga2-v2.4.4.exe runtergeladen und installiert.

Anmerkung: Es wurde diese Version verwendet da bei den neueren Versionen mehrere Fehler aufgetreten sind!

Für eine genaue Installationsanleitung wurde das Video
<https://www.youtube.com/watch?v=LQ5wPrAncNc> verwendet.



5.2.6 Einbinden ins verteilte Netzwerk

Anschließend wurden die PCs durch den **icinga2 node wizard** (siehe 7.1) als Master und Slave deklariert. Nachdem alles eingestellt wurde konnten alle Standard Services des Slaves in Icingaweb2 des Masters betrachtet werden.

Abbildung 76 – Verteiltes Netzwerk Milestone 2

5.2.7 Einbinden der Clients

Hinterher wurde das System um zwei Clients erweitert:

- APC UPS Network Management Card 2 (Client 1)



Abbildung 77 – APC UPS Network Management Card 2



- CISCO WAP321 (Client 2)



Abbildung 78 – CISCO WAP321

Um die beiden Clients von den Satellite PCs überwachen zu lassen müssen diese einfach in den entsprechenden Icinga2 Konfigurationen als Hosts angelegt werden und ein entsprechender Service geschrieben werden (Client 1 wurde im Ubuntu PC als Host angelegt und ein Service geschrieben, Client 2 im Windows PC).

Anmerkung: Der Master muss also keine Hostdefinitionen oder Servicedefinitionen anlegen, sondern muss nur die Satellite PCs einbinden. Anschließend werden im Icingaweb2 die beiden Satellite PCs sowie die Clients als eigene Hosts angezeigt.

Unter Ubuntu muss man in die Dateien /etc/icinga2/conf.d/hosts.conf und /etc/icinga2/conf.d/services.conf gehen und folgende Konfigurationen anlegen:

```
object Host "Network-Card" {
    import "generic-host"

    address = "10.48.1.31"

    vars.notification["mail"] = {
        groups = [ "icingaadmins" ]
    }
}
```

Abbildung 79 – Hostdefinition Milestone 2



```
object Service "Input Voltage" {

    import "generic-service"
    host_name = "Network-Card"
    check_command = "snmp"
    vars.snmp_oid = ".1.3.6.1.4.1.318.1.1.1.3.2.1.0"

    vars.sla = "24x7"

    vars.snmp_warn = 240
    vars.snmp_crit = 250
}
```

Abbildung 80 – Servicedefinition Milestone 2

Unter Windows muss unter C:\Program Files (x86)\ICINGA2\etc\icinga2\conf.d\hosts folgende Konfiguration angelegt werden:

```
object Host "CISCO" {
    import "generic-host"
    address = "10.48.1.20"
    vars.os = "windows"
    vars.notification["mail"] = {
        groups = [ "icingaadmins" ]
    }
}
```

Abbildung 81 – Hostdefinition Milestone 2 Windows

Anmerkung: Für den Client unter Windows (CISCO WAP321) wurde kein Service angelegt da keine entsprechende MIB gefunden wurde, jedoch würde die Konfiguration wie bei Ubuntu ablaufen.



Anschließend ist in Icingaweb2 des Masters folgendes zu sehen:

Abbildung 82 – Overview Hosts Verteiltes Netzwerk

Abbildung 83 – Overview Services Verteiltes Netzwerk



Hosts

Services

Search... Hostname

Search... Service Name

1 row(s) selected

5 Hosts: 5 0 row(s) selected

36 Services: 36

Abbildung 84 – Overview Services Verteiltes Netzwerk (2)

Host

Services

History

Search... Hostname

1 row(s) selected

5 Hosts: 5

1 Service: 1

Check now Comment Notification Downtime

Plugin Output

PING OK RTA: 0ms Packet loss: 0%

Problem handling

Comments Add comment

Downtimes Schedule downtime

Performance data

Label	Value	Warning	Critical
rtt	0.00 s	3.00 s	5.00 s
pl	0%	80%	100%

Notifications

Send notification Icinga 2 Admin

Contacts Icinga 2 Admin Group

Check execution

Command dummy Process check result

Check Source windows is reachable

Last check 0m 35s ago Check now

Next check in 4m 12s Reschedule

Abbildung 85 – Hostansicht eines Client Verteiltes Netzwerk

Unter dem Reiter Hosts sind nun der Ubuntu sowie der Windows Satellite PC zu sehen und die beiden Clients als eigene Hosts. Jedes dieser Geräte besitzt nun die vorher dazu angelegten Services und wird somit vom Master überwacht. Diese Anzeige ist in Icingaweb2 zu finden und kann mit dem Graphite Plugin verbunden werden.



6 Zeitaufwand

6.1 Mujkanovic

01.03.16	9	8:00	11:30	3,50	0,00	3,50	Vorstellungen der Diplomarbeiten; Findung des Teams; Auswahl des Themas (Netzwerküberwachung Windkraft Simonsfeld)
15.03.16	11	9:45	10:15	0,50	0,00	0,50	Erstes Gespräch mit dem Betreuer (Thema festigen, Daten austauschen, Ziele festlegen)
17.03.16	11	13:30	13:45	0,25	0,00	0,25	Gespräch mit Betreuer: Netzwerk soll mit Icinga2 & Icingaweb2 überwacht werden; Aufgabe Osterferien: Icingaweb2 installieren
23.03.16	12	12:00	15:00	0,00	3,00	3,00	Ubuntu 16.04 in Virtualbox installieren, Icinga Installationsanleitung finden, Icinga anfangen zu installieren
30.03.16	13	13:30	14:00	0,50	0,00	0,50	Fehler bei Icinga2 Installation, Betreuer um Hilfe bitten; Links zu Videos über Icinga2 erhalten
20.04.16	16	13:10	16:35	3,42	0,00	3,42	Icinga2 erneut installieren, erneut Fehler bei der Installation
26.04.16	17	20:00	21:30	0,00	1,50	1,50	Icinga2 erneut installieren, erneut Fehler bei der Installation
27.04.16	17	10:00	11:20	0,00	1,33	1,33	Icinga2 Fehler beheben & installieren (Mysql-IDO und Pgsql-IDO nicht parallel verwendbar); Fehler bei Icingaweb2 Installation
02.05.16	18	16:00	18:00	0,00	2,00	2,00	Ersten beiden Videos von Betreuer schauen
04.05.16	18	8:30	9:00	0,50	0,00	0,50	Gespräch mit Betreuer (Ubuntu soll neu installiert werden da mysql und postgresql nicht parallel verwendet werden kann)
11.05.16	19	13:10	16:30	3,33	0,00	3,33	Ubuntu 16.04 neu installieren, Icinga2 neu installieren, Icingaweb2 installieren
25.05.16	21	13:10	16:30	3,33	0,00	3,33	Icingaweb2 konfigurieren, Aufgabenstellung erhalten: CPU-Auslastung und Festplattenspeicher mit Icinga überwachen
08.06.16	23	14:00	16:30	2,50	0,00	2,50	2 Videos anschauen (https://www.youtube.com/watch?v=t6pJwyh2oBQ , https://www.youtube.com/watch?v=AFoHRjBfxmw), Icingaweb2 Basics lernen
15.06.16	24	13:00	16:15	3,25	0,00	3,25	Besprechung mit der Firma (siehe Betreuungsprotokoll)
16.06.16	24	18:00	22:00	0,00	4,00	4,00	Icingaweb2 Grundlagen lernen (https://www.linuxcampus.net/component/redevent/details/65?xref=0&pop=1&tmpl=component) (http://docs.icinga.org/icinga2/latest/doc/module/icinga2/toc)
27.06.16	26	7:50	11:30	3,67	0,00	3,67	Mit Distributed Monitoring befassen (https://www.icinga.org/products/icinga-2/distributed-monitoring/)



28.06.16	26	7:50	11:30	3,67	0,00	3,67	Mit Icinga2 befassen (Weitere Youtube Videos anschauen)
02.07.16	26	9:00	11:00	0,00	2,00	2,00	Icinga2 Buch lesen
03.07.16	26	9:00	11:00	0,00	2,00	2,00	Icinga2 Buch lesen
04.07.16	27	19:00	22:00	0,00	3,00	3,00	Icinga2 Buch lesen
05.07.16	27	17:00	20:00	0,00	3,00	3,00	Icinga2 Buch lesen
06.07.16	27	17:00	20:00	0,00	3,00	3,00	Icinga2 Buch lesen
07.07.16	27	16:00	20:00	0,00	4,00	4,00	Versuchen mit Server zu verbinden, Fehlschlag, Michael Krell darüber informieren, Icinga2 Buch lesen
08.07.16	27	14:00	17:00	0,00	3,00	3,00	Icinga2 Buch lesen
08.07.16	27	21:00	22:00	0,00	1,00	1,00	HTL-VPN neu einrichten und Serververbindung darüber versuchen, immer noch Fehlschlag, weitere Dinge versuchen (Firewall deaktivieren, X11 fowarding aktivieren, immer noch Fehlschlag)
09.07.16	27	11:00	14:00	0,00	3,00	3,00	Icinga2 Buch lesen
10.07.16	27	8:00	12:00	0,00	4,00	4,00	Letze Seiten von Icinga2 Buch lesen sowie Ubuntu 16.04 & Icinga2 & Icingaweb2 auf Laptop neu installieren (System Probleme)
11.07.16	28	10:00	16:00	0,00	6,00	6,00	Mit Icinga2 befassen und Icingaweb2 (Dateien anschauen „host.conf“ „servics.conf“ usw. sowie versuchen, alle Überwachungen selber nachzuschreiben
12.07.16	28	9:00	10:00	0,00	1,00	1,00	Serververbindungsfehler versuchen zu beheben, nicht geschafft --> sobald wie möglich Nachricht an Krell schreiben
12.07.16	28	15:15	15:20	0,00	0,08	0,08	Serverbindung funktioniert, Problem: Routing Problem mit dem ISP, Host wurde in ein anderes Subnet weitergeleitet (gelöst von Krell)
12.07.16	28	18:20	21:20	0,00	3,00	3,00	Icinga2 und Icingaweb2 auf Server installieren, Fehlersuche aufgrund Pfade die Berechtigungen brauchen --> Read Write and Execute für alle User erlauben, fertig installiert
13.07.16	28	7:00	11:00	0,00	4,00	4,00	Icinga2 Zusammenfassung „Einstieg in Icinga2“ schreiben
13.07.16	28	18:00	20:30	0,00	2,50	2,50	Icinga Master installieren, versuchen CPU komplett auszulasten um Funktionalität des CPU Checks anzuschauen sowie die Icingaweb2 Oberfläche genauer anschauen
16.08.16	33	12:00	14:00	0,00	2,00	2,00	Teambesprechung (Fortschritte besprechen, Aufgaben aufteilen)
16.08.16	33	18:00	21:00	0,00	3,00	3,00	Erneutes einlesen in Icinga2 (Buch) wegen Urlaub
23.08.16	34	10:00	12:00	0,00	2,00	2,00	Testsensor als Host anlegen (Problem beim Hostnamen wegen Anführungszeichen), ping-service anlegen sowie Krell darüber informieren
24.08.16	34	10:00	12:00	0,00	2,00	2,00	Mit Testsensor "Temperatursensor HWG-Ste" befassen
27.08.16	34	12:00	14:00	0,00	2,00	2,00	Information von Schrenk: "Backend icinga is not running" -> Fehlersuche



29.08.16	35	13:30	15:30	0,00	2,00	2,00	Fehlersuche für den Fehler "Backend icinga is not running"
31.08.16	35	12:00	14:00	0,00	2,00	2,00	Fehlersuche, Information an Krell (Information aus Internet: Snapshotpakete sollen installiert werden)
03.09.16	35	12:00	13:00	0,00	1,00	1,00	Gespräch mit Schrenk: Er will sich dem Fehler widmen
09.09.16	36	7:50	10:35	2,75	0,00	2,75	Vortrag von Resel über Diplomarbeitspräsentationen, Gespräch mit Betreuer über den Fortschritt sowie sich dem Fehler weiter widmen
09.09.16	36	10:35	11:30	0,00	0,92	0,92	Fehlersuche
09.09.16	36	11:30	13:10	1,67	0,00	1,67	Fehlersuche -> keine Lösung gefunden, Krell informieren und wahrscheinlich Server neu aufsetzen
14.09.16	37	14:00	16:00	0,00	2,00	2,00	Präsentation vorbereiten
15.09.16	37	17:00	19:30	0,00	2,50	2,50	Icinga2 und Icingaweb2 deinstallieren vom Server, Präsentation vorbereiten
16.09.16	37	7:50	10:35	2,75	0,00	2,75	Kickoff-Präsentation, Icinga2 und Icingaweb2 neu installieren
16.09.16	37	10:35	11:30	0,00	0,92	0,92	Mysql Fehler (mysql.user doesn't exist) wenn mysql gestartet werden soll; mysql neu installieren --> Fehler immer noch vorhanden
20.09.16	38	12:20	13:20	1,00	0,00	1,00	Versuchen alle Icinga2 und Mysql Pakete zu deinstallieren --> Krell fragen ob er Server neu aufsetzen kann
20.09.16	38	14:50	16:30	1,67	0,00	1,67	Icinga2 Buch lesen da Server neu aufgesetzt werden muss
23.09.16	38	7:50	10:35	2,75	0,00	2,75	Icinga2 und Icingaweb2 neu installieren
23.09.16	38	10:35	11:30	0,00	0,92	0,92	Fehler während der Installation beheben
23.09.16	38	11:30	13:30	2,00	0,00	2,00	Icinga2 und Icingaweb2 neu installiert --> Fehlermeldung "Backend icinga is not running" immer noch da
26.09.16	39	18:30	21:00	0,00	2,50	2,50	Fehler "Backend icinga is not running" beheben, Temperatursensor in Icinga2 und Icingaweb2 einbinden, SNMP Abfrage für Temperatursensor schreiben --> Erfolg
30.09.16	39	7:50	10:35	2,75	0,00	2,75	Dokumentation "Wiki_1" für Krell schreiben
30.09.16	39	10:35	11:30	0,00	0,92	0,92	Schrenk bei diversen Problemen helfen
30.09.16	39	11:30	13:10	1,67	0,00	1,67	Icinga2 Buch lesen (mit Downtimes befassen)
03.10.16	40	16:30	19:00	0,00	2,50	2,50	Icinga2 Buch lesen (Verteilte Überwachung)
06.10.16	40	17:00	21:00	0,00	4,00	4,00	Im Internet suchen wie man Überwachungen grafisch darstellen könnte --> Graphite --> Installieren mit mehreren Anleitungen aus dem Internet --> Fehlschlag, Apache lässt sich nicht mehr starten
07.10.16	40	7:50	10:35	2,75	0,00	2,75	Fehler von Apache beheben --> Standardport wurde während der Installation von Graphite auf 8000 gestellt welcher gesperrt ist (auf dem Firmenserver)



07.10.16	40	10:35	11:30	0,00	0,92	0,92	Graphite installieren (https://www.youtube.com/watch?v=y39RHFZ3vhl), Krell informieren dass Port 2003 gebraucht wird
07.10.16	40	11:30	13:10	1,67	0,00	1,67	Installationsanleitung für Graphite ist auch im Icinga2 Buch vorhanden --> Alle Pakete die von dem Youtube-Video installiert wurden wieder entfernen und alles aus dem Buch installieren
10.10.16	41	14:30	17:00	0,00	2,50	2,50	Persönliche Dokumentation anfangen; alle Links zusammen suchen; alles zusammenschreiben
14.10.16	41	7:50	10:35	2,75	0,00	2,75	Ubuntu 16.04 neu installieren da es Systemfehler gibt, Icinga2 und Icingaweb2 installieren
14.10.16	41	10:35	11:30	0,00	0,92	0,92	Graphite lokal installieren, Fehler 'Couldn't listen on 0.0.0.0:2003 [Errno98] Address already in use.';
14.10.16	41	11:30	13:10	1,67	0,00	1,67	Fehlersuche --> erfolglos
19.10.16	42	15:00	15:30	0,00	0,50	0,50	Dokumentation schreiben
25.10.16	43	18:00	20:00	0,00	2,00	2,00	Dokumentation schreiben
26.10.16	43	9:00	11:00	0,00	2,00	2,00	Dokumentation schreiben
27.10.16	43	17:00	23:00	0,00	6,00	6,00	Graphite auf Server installieren; Fehler 'Couldn't listen on 0.0.0.0:2003: [Errno 98] Address already in use.'; Fehlersuche --> Fehlersuche nicht erfolgreich, Krell bitten Server neu aufzusetzen; Graphite erneut lokal installieren
28.10.16	43	7:50	10:35	2,75	0,00	2,75	Graphite nach Buch lokal installieren, FEHLER
28.10.16	43	10:35	11:30	0,00	0,92	0,92	Fehlersuche, Ubuntu & Icinga neu installieren
28.10.16	43	11:30	13:10	1,67	0,00	1,67	Graphite nach https://www.digitalocean.com/community/tutorials/how-to-install-and-use-graphite-on-an-ubuntu-14-04-server installieren --> ERFOLG!!!!!!
29.10.16	43	12:30	14:30	0,00	2,00	2,00	Graphite-Web anschauen; Unterordner "icinga2" ist zwar da jedoch werden die Services nicht angezeigt (müssen wahrscheinlich irgendwo extra für Graphite definiert werden); Graphite Modul in Icingaweb2 freischalten, statt Graph wird jedoch "No Text" angezeigt --> Fehlersuche folgt
29.10.16	43	19:00	22:00	0,00	3,00	3,00	Mit Graphite befassen (wie werden Services definiert, wie werden Graphen in Icingaweb2 und in Graphite-Web angezeigt usw.)
02.11.16	44	12:00	14:00	0,00	2,00	2,00	Graphite-Video schauen; Graphite Modul in Icinga2 aktivieren --> Graphen werden nun endlich in Graphite angezeigt (siehe Screenshot von CPU Auslastung)
02.11.16	44	19:30	21:00	0,00	1,50	1,50	Versuchen auf Server MySQL und MariaDB zu reparieren (MariaDB löschen) da Krell Server nicht zurücksetzt; Problem: dpkg unter /usr/bin wurde gelöscht --> apt-get upgrade usw. funktioniert alles nicht, versuchen dies zu



							reparieren --> erfolgslos; müssen warten auf zurücksetzen durch Krell
04.11.16	44	7:50	10:35	2,75	0,00	2,75	Fehler: /usr/bin/dpkg wurde gelöscht --> kopieren von lokalem Ubuntu: Erfolg; Nach Fehler auf Server suchen --> MySQL und MariaDB können nicht gestartet werden, bei der Deinstallation von MySQL tritt Fehler auf: MySQL wird nie komplett gelöscht
04.11.16	44	10:35	11:30	0,00	0,92	0,92	MySQL komplett löschen (Erfolg)
04.11.16	44	11:30	13:10	1,67	0,00	1,67	Graphite auf Server installieren --> Erfolg; Werte von Temperatursensor anzeigen lassen funktioniert nicht (Aufgabe bis nächsten Freitag: Temperaturwerte anzeigen lassen)
05.11.16	44	16:00	19:00	0,00	3,00	3,00	Mit Graphite befassen (warum wird keine einzige Kurve angezeigt) --> Graphite-Web ignorieren und Graphite Modul für Icingaweb2 aktivieren, Kurven von ping und apt usw. können angezeigt werden, Temperatur aufgrund von SNMP nicht
06.11.16	44	10:00	11:00	0,00	1,00	1,00	Fehlersuche --> SNMP Werte werden nicht als Perfomance Daten ausgegeben, daher auch kein Graph
06.11.16	44	19:30	20:45	0,00	1,25	1,25	Erneute Fehlersuche; Plugin funktioniert an sich, SNMP Abfrage auch, nur die Ausgabe der Perfomance Daten scheint "kaputt zu sein"
07.11.16	45	14:00	14:30	0,00	0,50	0,50	Alte Abfragen wieder hinzufügen (für den PC, z.B.: Festplattenspeicher, SSH usw.) um für Tag der offenen Tür mehrere Abfragen zu haben
08.11.16	45	13:30	16:35	3,08	0,00	3,08	Dokumentation schreiben
09.11.16	45	14:00	15:30	0,00	1,50	1,50	Fehlersuche; Mögliche Lösung: Es sind keine Werte gesetzt wann der Service den Zustand "Warning" und "Critical" einnehmen soll --> Diese Parameter setzen, Warning ab 30°C und Critical ab 40°C (keine genauen Werte, nur Zufallswerte) --> Erfolg, Ausgabe ist nun Integer und somit Perfomance Data --> Graphite Graph funktioniert
10.11.16	45	18:00	18:30	0,00	0,50	0,50	Test-Email-Adresse anlegen (da.wks.test@gmail.com) --> mit Email-Notifications befassen
11.11.16	45	7:50	10:35	2,75	0,00	2,75	Gespräch mit Betreuer --> Check_MK installieren um Unterschied zu Icinga zu sehen; Installation beginnen
11.11.16	45	10:35	11:30	0,92	0,00	0,92	Check_MK versuchen zu installieren --> Fehlschlag
11.11.16	45	11:30	13:10	1,67	0,00	1,67	Check_MK versuchen zu installieren --> Fehlschlag
17.11.16	46	14:55	16:35	1,67	0,00	1,67	Email Notification für Icinga einrichten (http://www.modius-techblog.de/linux/icinga-2-notifications/) --> Fehler Emails werden nicht gesendet



17.11.16	46	18:00	20:00	0,00	2,00	2,00	Fehlersuche --> postfix installieren, Email Benachrichtigungen funktionieren für Temperatur Service und alle anderen Services
18.11.16	46	7:50	10:35	2,75	0,00	2,75	Gespräch mit Betreuer, check_mk soll installiert werden; Versuchen zu installieren bis 8:40, danach aufbauen für Tag der offenen Tür
18.11.16	46	10:35	12:20	0,00	1,75	1,75	Aufbauen für Tag der offenen Tür
18.11.16	46	13:10	17:00	0,00	3,83	3,83	Tag der offenen Tür
19.11.16	46	8:30	12:30	0,00	4,00	4,00	Tag der offenen Tür
21.11.16	47	17:00	20:00	0,00	3,00	3,00	Check_MK versuchen zu installieren; Erkenntnis: Kann nicht für Ubuntu 16.04 installiert werden; Dokumentation für Krell schreiben (Milestone 1: Erweitern um Graphite & E-Mail Benachrichtigung)
22.11.16	47	14:00	14:50	0,83	0,00	0,83	Dokumentation für Krell fertig stellen
24.11.16	47	18:00	22:00	0,00	4,00	4,00	Ubuntu 14.04 auf Laptop installieren um Check_mk lokal zu installieren; Check_mk installieren; Verteilte Überwachung aus Icinga2 Buch lesen
25.11.16	47	7:50	10:35	2,75	0,00	2,75	Gespräch mit Betreuer: Check_MK herzeigen, Aufgabe: Verteiltes Netzwerk zuhause aufbauen und beide Programme verwenden; Check_MK Tutorials lesen
25.11.16	47	10:35	11:30	0,00	0,92	0,92	Betreuungsprotokolle nachschreiben
25.11.16	47	11:30	13:10	1,67	0,00	1,67	Betreuungsprotokolle nachschreiben
25.11.16	47	21:30	0:00	0,00	2,50	2,50	Videos von Betreuer schauen (Verteilte Überwachung in Icinga2)
27.11.16	47	15:15	19:05	0,00	3,83	3,83	Plan: Laptop ist Master, PC ist Slave (Windows & Ubuntu durch Virtualbox); Virtualbox auf Stand-PC installieren mit Ubuntu 14.04 (damit Check_MK laufen kann); Problem bei Installation: Stand-PC hat AMD-CPU (nur 32 bit Ubuntu möglich); Ubuntu nach 1. Installation nicht richtig gebootet --> erneute Installation; Icinga2 & Icingaweb2 auf Stand-PC installieren (Ubuntu); Check_MK installieren auf Stand-PC (Ubuntu) Erkenntnis: Check_MK kann auch unter Ubuntu 16.04 installiert werden, konnte damals nicht installiert werden da eine alte Version des Programms runtergeladen wurde; Icinga2 auf Windows des Stand-PC installieren
28.11.16	48	17:50	18:30	0,00	0,67	0,67	Icinga2 auf Laptop installieren (da nicht mehr vorhanden; Grund: unbekannt); Ubuntu 14.04 System Fehler --> Neuinstallation von Ubuntu (sobald wie möglich tun)
28.11.16	48	19:10	20:40	0,00	1,50	1,50	Ubuntu 14.04 auf Laptop installieren; Icinga2 und Icingaweb2 installieren; Master-Setup auf Laptop installieren; Versuchen Windows-Satellite als Satellite einzusetzen und zu überwachen --> Fehler: Windows Host kann den



							Master nicht finden, wahrscheinlich Fehler in der Icinga2 Node Wizard Konfiguration
29.11.16	48	19:20	21:30	0,00	2,17	2,17	Fehlersuche --> Seite finden; Icinga2 Node Wizard auf Laptop löschen & Verteilte Überwachung mit Windows Client nach der Seite zu installieren; Erfolg, Windows Satellite wurde in Icingaweb2 eingebunden und abfragen werden durchgeführt; Nächster Schritt: Wie können diese Services verändert werden + Test (was passiert wenn die Verbindung zwischen Master und Satellite ausfällt)
30.11.16	48	14:30	18:00	0,00	3,50	3,50	Icinga2 Fehler --> Icinga2 lässt sich nicht mehr starten, Fehler "Satellite is in more than one zone" obwohl Satellite gar nicht mehr angelegt ist --> Fehlersuche; Resultat: Icinga2 neu installieren (alle Dateien zurücksetzen); Icinga2 & Icingaweb2 & Icinga2 Node Wizard auf Laptop neu installieren; Währenddessen--> Ubuntu auf Stand-PC neu installieren da Virtualbox nicht angepingt werden kann; BIOS erkennt USB-Stick nicht --> Fehlersuche; Erfolg --> Ubuntu sowie Icinga2 & Icingaweb2 auf Stand-PC installieren
01.12.16	48	10:35	11:25	0,83	0,00	0,83	Stundentafel "verbessern" (Einträge neu schreiben bzw. bearbeiten)
01.12.16	48	18:00	21:00	0,00	3,00	3,00	Icinga2 Node Wizard auf Stand-PC installieren (Ubuntu Satellite Setup); Fehler auf Master Laptop (Master wurde in mehreren Dateien definiert) --> Dateien zurücksetzen; Fehler bei Satellite hinzufügen: Unauthorized
02.12.16	48	7:50	10:35	2,75	0,00	2,75	Aufbauen von verteiltem Netzwerk (Master und Windows Satellite; auf beiden Icinga2 und nur auf Master Icingaweb2 installieren)
02.12.16	48	10:35	11:30	0,00	0,92	0,92	Windows PC kommt nicht ins Internet --> Fehlersuche
02.12.16	48	11:30	13:10	1,67	0,00	1,67	Fehler behoben (Windows Firewall deaktiviert); Einbinden von Windows Satellite in Master Icingaweb2 --> Fehler: nicht erreichbar (jedoch als normaler Host erreichbar)
02.12.16	48	19:20	21:00	0,00	1,67	1,67	Icinga auf Laptop zurücksetzen (um Standardeinstellungen zu haben) und in den Master erneut Stand-PC einbinden (mit Windows) um zu testen, ob Fehler von Schule an Netzwerk oder PC liegt; Fehler lag nicht am PC; Test (Verbindung trennen) lieferte nicht das erwünschte Szenario --> Icinga hat nichts getan als die Verbindung wieder hergestellt wurde (außer weiter Checks ausführen)
03.12.16	48	19:00	20:00	0,00	1,00	1,00	Mit Unterschied zwischen Top-Down und Bottom-Up befassen
05.12.16	49	18:00	20:00	0,00	2,00	2,00	Schul-PC in Netzwerk zuhause einbinden --> Icinga Überwachung funktioniert



16.12.16	50	7:50	10:35	2,75	0,00	2,75	Schul-PC nochmal in Schule ausprobieren; Ubuntu und Icinga auf Master installieren
16.12.16	50	10:35	11:30	0,00	0,92	0,92	Icinga2-ido-mysql Fehler
16.12.16	50	11:30	13:10	1,67	0,00	1,67	Icinga2-ido-mysql Fehler
17.12.16	50	12:30	16:00	0,00	3,50	3,50	Zuhause Icinga2 System erneut aufsetzen um Bottom Up hinzukriegen --> Laptop ist Master, Stand-PC ist Slave; E-Mail Benachrichtigung am Master einrichten & Test über Satellite-CPU ob E-Mail auch für Satellites gesendet wird; Postfix wurde installiert aber es werden keine E-Mails gesendet (weder über Icinga noch über mail -s ...)
18.12.16	50	13:00	15:30	0,00	2,50	2,50	Versuchen E-Mails von Laptop per Postfix zu versenden, erfolgslos --> Fehlersuche ebenfalls erfolgslos
27.12.16	52	11:00	15:00	0,00	4,00	4,00	Verteilte Überwachung V2 schreiben; E-Mail Benachrichtigung erstmal vergessen --> Versuchen Bottom Up hinzukriegen --> erfolgslos; Graphite installieren um zu schauen was mit Graph passiert wenn Überwachung über einen Zeitraum ausfällt; Graphite Fehler bei Installation
01.01.17	52	16:00	18:00	0,00	2,00	2,00	Verteilte Überwachung (Icinga2) wird auf Schrenk verschoben da von Krell Satellite PCs zur Verfügung gestellt wurden; Check_MK ausprobieren (Verteilte Überwachung)
03.01.17	1	12:00	16:00	0,00	4,00	4,00	Mit Check_MK befassen
04.01.17	1	11:00	13:30	0,00	2,50	2,50	Mit Check_MK befassen
04.01.17	1	18:00	20:00	0,00	2,00	2,00	Check_MK versuchen Host einzufügen, Fehlschlag da Funktion von Check_MK noch nicht ganz verstanden
05.01.17	1	11:00	14:00	0,00	3,00	3,00	Mit Check_MK befassen
05.01.17	1	19:00	21:00	0,00	2,00	2,00	Check_MK einrichten sowie einen weiteren PC aus dem Netzwerk als Host einfügen
06.01.17	1	8:00	11:00	0,00	3,00	3,00	Persönliche Dokumentation weiterschreiben
09.01.17	2	18:00	20:30	0,00	2,50	2,50	Unterschiede zwischen Check_MK und Icinga2 feststellen; Für Icinga2 entscheiden da es in der Open Source Version mehr Funktionen hat und die Oberfläche (subjektiv betrachtet) besser scheint
13.01.17	2	7:50	10:35	2,75	0,00	2,75	Icinga2 auf Satellite von der Firma installieren --> Satellite ist DOWN gegangen, keine Verbindung möglich; Satellite geht wieder
13.01.17	2	10:35	11:30	0,00	0,92	0,92	Icinga2 installieren; Satellite in Master einbinden --> Erfolg
13.01.17	2	11:30	13:10	1,67	0,00	1,67	Network Card versuchen zu überwachen --> befassen mit APC UPS Network Card 2
13.01.17	2	14:00	18:00	0,00	4,00	4,00	Nach OIDs und MIBs von Network Card suchen --> Suche verlief nicht erfolgreich
14.01.17	2	12:00	14:00	0,00	2,00	2,00	Mit Network Card befassen um Funktion zu verstehen



14.01.17	2	22:00	0:00	0,00	2,00	2,00	Präsentation vorbereiten für interne Präsentation
19.01.17	3	18:00	21:00	0,00		3,00	APC UPS Network Management Card 2 in Icinga Überwachung einbinden; Fehlschlag (Master akzeptiert Services zu Card nicht); Präsentation vorbereiten
20.01.17	3	7:50	10:35	2,75		2,75	Präsentation halten und anhören von Kollegen
20.01.17	3	10:35	11:30	0,00	0,92	0,92	Präsentation verbessern (Veränderungsvorschläge von Betreuer)
20.01.17	3	11:30	13:10	1,67	0,00	1,67	Präsentation verbessern (Veränderungsvorschläge von Betreuer)
20.01.17	3	18:00	20:00	0,00	2,00	2,00	Logo für Präsentation/Diplomarbeit erstellen (V1)
21.01.17	3	10:00	12:00	0,00	2,00	2,00	Präsentation fertig stellen (für bisherigen Stand)
22.01.17	3	13:00	15:00	0,00	2,00	2,00	Weiter nach Möglichkeit suchen eigene Services bei Satellite einzubinden --> ohne Erfolg
23.01.17	4	15:00	17:00	0,00	2,00	2,00	Logo V2 entwickeln
24.01.17	4	12:20	16:35	4,25	0,00	4,25	Windows Satellite PC mit Schrenk einbinden (ihm beibringen); persönliche Doku weiterschreiben
24.01.17	4	19:00	21:00	0,00	2,00	2,00	Icinga2 Konfiguration auf Standard zurücksetzen und neu konfigurieren da Satellite Konfiguration die Icinga2 Konfiguration zerstört hat --> alte Konfiguration wieder herstellen; Fehler: Icinga2 wird auf Server und auf Satellite nicht mehr richtig installiert (/etc/icinga2 fehlt); Icinga2 auf Server und auf Ubuntu neu installieren
25.01.17	4	19:00	20:00	0,00	1,00	1,00	Icinga2 Verteiltes Netzwerk erneut aufbauen --> Fehler mit Apache2 und anderen Programmen; Krell fragen ob er den Server neu aufsetzen kann (schnellste Möglichkeit)
27.01.17	4	7:50	10:35	2,75	0,00	2,75	Aufbauen für Tag der offenen Tür
27.01.17	4	10:35	11:30	0,00	0,92	0,92	Vorbereiten für Tag der offenen Tür
27.01.17	4	11:30	13:10	1,67	0,00	1,67	Vorbereiten für Tag der offenen Tür
28.01.17	4	8:00	12:30	0,00	4,50	4,50	Tag der offenen Tür
31.01.17	5	12:20	14:50	2,50	0,00	2,50	Logo V2 entwickeln
01.02.17	5	15:30	16:30	0,00	1,00	1,00	Hauptserver wurde neu aufgesetzt --> Icinga2, Icingaweb2 installieren sowie Windows PC einbinden
03.02.17	5	7:50	10:35	2,75	0,00	2,75	Ubuntu PC einbinden in Überwachung, mit Bottom Up und Top Down befassen
06.02.17	6	14:00	17:00	0,00	3,00	3,00	Mit Bottom Up und Top Down befassen (https://docs.icinga.com/icinga2/latest/doc/module/icinga2/toc#!/icinga2/latest/doc/module/icinga2/chapter/distributed-monitoring); Doku der letztjährigen Diplomarbeiten lesen und eigene DA Doku beginnen: Kapitel 1 schreiben
07.02.17	6	12:00	15:00	0,00	3,00	3,00	Dokumentation schreiben: Kapitel 1 lesen und Kapitel 2 bis 2.5 Icingaweb2 schreiben; Network Card einbinden über Ubuntu PC (Keine Satellite Services)



08.02.17	6	12:00	15:00	0,00	3,00	3,00	Dokumentation: Alles bisher Geschriebene lesen und verbessern
09.02.17	6	10:20	12:20	0,00	2,00	2,00	Dokumentation weiterschreiben bis 3.3 MIB
09.02.17	6	12:30	14:30	0,00	2,00	2,00	Satellite-Service für CISCO (Windows) und Network Card (Ubuntu) schreiben; Beides wird zwar eingebunden aber leider nur Pending Zustand --> Fehlersuche; Erkenntnis: Check muss nur auf Satellite in conf.d sein um Bottom Up zu haben --> Problem gelöst
09.02.17	6	16:15	17:15	0,00	1,00	1,00	Windows Satellite versuchen CISCO überwachen zu lassen
10.02.17	6	12:30	13:30	0,00	1,00	1,00	Icinga 2.4.4. downloaden --> In Windows Satellite alles CISCO Router einrichten --> Erfolg! Doku_Wiki2 schreiben und Krell mitteilen dass letzter Milestone begonnen werden kann
17.02.17	7	7:50	10:35	2,75	0,00	2,75	Graphite installieren
17.02.17	7	10:35	11:30	0,00	0,92	0,92	Graphite installieren
17.02.17	7	11:30	13:10	1,67	0,00	1,67	Graphite installieren
17.02.17	7	18:00	22:00	0,00	4,00	4,00	Doku_Wiki2 für Krell schreiben
20.02.17	8	16:00	18:00	0,00	2,00	2,00	Dokumentation schreiben
21.02.17	8	18:00	21:00	0,00	3,00	3,00	Dokumentation schreiben
23.02.17	8	18:00	20:00	0,00	2,00	2,00	Dokumentation schreiben
24.02.17	8	7:50	10:35	2,75	0,00	2,75	Gespräch mit Betreuer über die Dokumentation; Schrenk bei SNMP Problem helfen
24.02.17	8	10:35	11:30	0,00	0,92	0,92	Schrenk bei SNMP Problem helfen
24.02.17	8	11:30	13:10	1,67	0,00	1,67	Schrenk bei SNMP Problem helfen
24.02.17	8	13:30	17:30	0,00	4,00	4,00	Dokumentation schreiben
27.02.17	9	16:15	17:45	0,00	1,50	1,50	Dokumentation schreiben
28.02.17	9	18:00	19:00	0,00	1,00	1,00	Dokumentation schreiben
02.03.17	9	17:00	20:00	0,00	3,00	3,00	Dokumentation schreiben
03.03.17	9	8:00	12:00	0,00	4,00	4,00	Dokumentation schreiben
08.03.17	10	13:00	16:00	0,00	3,00	3,00	Dokumentation schreiben
09.03.17	10	15:00	18:00	0,00	3,00	3,00	Dokumentation schreiben
10.03.17	10	7:50	10:35	2,75	0,00	2,75	Apache2 auf Hauptserver funktioniert nicht; alles deinstallieren und alles neu installieren; Apache funktioniert aber der Befehl icingacli nicht --> Icingaweb2 kann nicht installiert werden
10.03.17	10	10:35	11:30	0,00	0,92	0,92	Dokumentation schreiben
10.03.17	10	11:30	13:10	1,67	0,00	1,67	Icingacli Kommando war beschädigt --> Icingaweb2 installieren jedoch werden alle Satellites als DOWN bezeichnet obwohl sie online sind und angepingt werden können
11.03.17	10	17:00	19:00	0,00	2,00	2,00	Dokumentation schreiben
12.03.17	10	18:00	20:00	0,00	2,00	2,00	Satellites sind DOWN in Icinga2 obwohl sie UP sind --> Versuchen zu lösen --> Ohne Erfolg
13.03.17	11	15:00	17:00	0,00	2,00	2,00	Satellites sind DOWN in Icinga2 obwohl sie UP sind --> Versuchen zu lösen mit Satellites entfernen und neu hinzufügen --> Erfolg; Graphite macht aber einen Fehler --> Wenn



							apache2-graphite Seite aktiviert wird kann Apache2 nicht mehr gestartet werden
14.03.17	11	18:00	21:00	0,00	3,00	3,00	Graphite-Fehler lösen (WSGI war nicht installiert); Graphite-web ist nun verfügbar jedoch kann Graphite Plugin für Icinga2 nicht gestartet/enabled werden --> Fehlersuche; Dokumentation schreiben
15.03.17	11	15:00	17:00	0,00	2,00	2,00	Dokumentation schreiben
16.03.17	11	18:00	21:00	0,00	3,00	3,00	Dokumentation schreiben
17.03.17	11	7:50	10:35	2,75	0,00	2,75	Graphite versuchen zu installieren
17.03.17	11	10:35	11:30	0,00	0,92	0,92	Graphite versuchen zu installieren --> Platzhalter sind eingefügt aber die Graphen fehlen immer noch (No Data wird angezeigt)
17.03.17	11	11:30	13:10	1,67	0,00	1,67	Graphite versuchen zu installieren (nun werden statt No Data leere Felder angezeigt)
17.03.17	11	13:30	14:30	0,00	1,00	1,00	Graphite versuchen zu installieren (ohne Erfolg; es funktioniert einfach nicht!!!) --> Entschluss: Graphite wird nicht installiert, es hat bereits 1 mal funktioniert und es wurden genug Screenshots für die Doku aufgenommen
18.03.17	11	17:15	18:15	0,00	1,00	1,00	Dokumentation analysieren sowie Stundentafel dem Betreuer schicken
19.03.17	11	10:30	11:30	0,00	1,00	1,00	Dokumentation analysieren und verbessern
19.03.17	11	19:30	20:30	0,00	1,00	1,00	Dokumentation analysieren und verbessern
21.03.17	12	18:00	19:00	0,00	1,00	1,00	Screenshot von Vorprojekt suchen (MS2) --> nicht auffindbar; Dokumentation weiterschreiben
23.03.17	12	10:00	12:30	0,00	2,50	2,50	Dokumentation weiterschreiben
23.03.17	12	14:00	16:00	0,00	2,00	2,00	Dokumentation analysieren und verbessern
23.03.17	12	19:30	20:00	0,00	0,50	0,50	Dokumentation analysieren und verbessern
28.03.17	13	18:00	21:00	0,00	3,00	3,00	Verbesserte Dokumentation von Betreuer erhalten --> Verbesserungsvorschläge in die Dokumentation einarbeiten
29.03.17	13	18:00	20:00	0,00	2,00	2,00	Verbesserungsvorschläge in die Dokumentation einarbeiten
30.03.17	13	20:00	21:00	0,00	1,00	1,00	Zusammengefügte Dokumentation analysieren sowie verbessern (von Schrenk zusammengefügt)
31.03.17	13	7:50	10:35	2,75	0,00	2,75	Dokumentation analysieren und verbessern
31.03.17	13	10:35	11:30	0,00	0,92	0,92	Dokumentation analysieren und verbessern
31.03.17	13	11:30	13:10	1,67	0,00	1,67	Dokumentation analysieren und verbessern
03.04.17	14	16:30	18:30	0,00	2,00	2,00	Dokumentation analysieren und verbessern



01.03.16	9	8:00	11:30	3,50	0,00	3,50	Vorstellungen der Diplomarbeiten; Findung des Teams; Auswahl des Themas (Netzwerküberwachung Windkraft Simonsfeld)
15.03.16	11	9:45	10:15	0,50	0,00	0,50	Erstes Gespräch mit dem Betreuer (Thema festigen, Daten austauschen, Ziele festlegen)
17.03.16	11	13:30	13:45	0,25	0,00	0,25	Gespräch mit Betreuer: Netzwerk soll mit Icinga2 & Icingaweb2 überwacht werden; Aufgabe Osterferien: Icingaweb2 installieren
23.03.16	12	12:00	15:00	0,00	3,00	3,00	Ubuntu 16.04 in Virtualbox installieren, Icinga Installationsanleitung finden, Icinga anfangen zu installieren
30.03.16	13	13:30	14:00	0,50	0,00	0,50	Fehler bei Icinga2 Installation, Betreuer um Hilfe bitten; Links zu Videos über Icinga2 erhalten
20.04.16	16	13:10	16:35	3,42	0,00	3,42	Icinga2 erneut installieren, erneut Fehler bei der Installation
26.04.16	17	20:00	21:30	0,00	1,50	1,50	Icinga2 erneut installieren, erneut Fehler bei der Installation
27.04.16	17	10:00	11:20	0,00	1,33	1,33	Icinga2 Fehler beheben & installieren (Mysql-IDO und Pgsql-IDO nicht parallel verwendbar); Fehler bei Icingaweb2 Installation
02.05.16	18	16:00	18:00	0,00	2,00	2,00	Ersten beiden Videos von Betreuer schauen
04.05.16	18	8:30	9:00	0,50	0,00	0,50	Gespräch mit Betreuer (Ubuntu soll neu installiert werden da mysql und postgresql nicht parallel verwendet werden kann)
11.05.16	19	13:10	16:30	3,33	0,00	3,33	Ubuntu 16.04 neu installieren, Icinga2 neu installieren, Icingaweb2 installieren
25.05.16	21	13:10	16:30	3,33	0,00	3,33	Icingaweb2 konfigurieren, Aufgabenstellung erhalten: CPU-Auslastung und Festplattenspeicher mit Icinga überwachen
08.06.16	23	14:00	16:30	2,50	0,00	2,50	2 Videos anschauen (https://www.youtube.com/watch?v=t6pJwyh2oBQ , https://www.youtube.com/watch?v=AFoHRjBfxmw), Icingaweb2 Basics lernen
15.06.16	24	13:00	16:15	3,25	0,00	3,25	Besprechung mit der Firma (siehe Betreuungsprotokoll)
16.06.16	24	18:00	22:00	0,00	4,00	4,00	Icingaweb2 Grundlagen lernen (https://www.linuxcampus.net/component/redevent/details/65?xref=0&pop=1&tmpl=component) (http://docs.icinga.org/icinga2/latest/doc/module/icinga2/toc)
27.06.16	26	7:50	11:30	3,67	0,00	3,67	Mit Distributed Monitoring befassen (https://www.icinga.org/products/icinga-2/distributed-monitoring/)
28.06.16	26	7:50	11:30	3,67	0,00	3,67	Mit Icinga2 befassen (Weitere Youtube Videos anschauen)



02.07.16	26	9:00	11:00	0,00	2,00	2,00	Icinga2 Buch lesen
03.07.16	26	9:00	11:00	0,00	2,00	2,00	Icinga2 Buch lesen
04.07.16	27	19:00	22:00	0,00	3,00	3,00	Icinga2 Buch lesen
05.07.16	27	17:00	20:00	0,00	3,00	3,00	Icinga2 Buch lesen
06.07.16	27	17:00	20:00	0,00	3,00	3,00	Icinga2 Buch lesen
07.07.16	27	16:00	20:00	0,00	4,00	4,00	Versuchen mit Server zu verbinden, Fehlschlag, Michael Krell darüber informieren, Icinga2 Buch lesen
08.07.16	27	14:00	17:00	0,00	3,00	3,00	Icinga2 Buch lesen
08.07.16	27	21:00	22:00	0,00	1,00	1,00	HTL-VPN neu einrichten und Serververbindung darüber versuchen, immer noch Fehlschlag, weitere Dinge versuchen (Firewall deaktivieren, X11 fowarding aktivieren, immer noch Fehlschlag)
09.07.16	27	11:00	14:00	0,00	3,00	3,00	Icinga2 Buch lesen
10.07.16	27	8:00	12:00	0,00	4,00	4,00	Letze Seiten von Icinga2 Buch lesen sowie Ubuntu 16.04 & Icinga2 & Icingaweb2 auf Laptop neu installieren (System Probleme)
11.07.16	28	10:00	16:00	0,00	6,00	6,00	Mit Icinga2 befassen und Icingaweb2 (Dateien anschauen „host.conf“ „serivces.conf“ usw. sowie versuchen, alle Überwachungen selber nachzuschreiben)
12.07.16	28	9:00	10:00	0,00	1,00	1,00	Serververbindungsfehler versuchen zu beheben, nicht geschafft --> sobald wie möglich Nachricht an Krell schreiben
12.07.16	28	15:15	15:20	0,00	0,08	0,08	Serverbindung funktioniert, Problem: Routing Problem mit dem ISP, Host wurde in ein anderes Subnet weitergeleitet (gelöst von Krell)
12.07.16	28	18:20	21:20	0,00	3,00	3,00	Icinga2 und Icingaweb2 auf Server installieren, Fehlersuche aufgrund Pfade die Berechtigungen brauchen --> Read Write and Execute für alle User erlauben, fertig installiert
13.07.16	28	7:00	11:00	0,00	4,00	4,00	Icinga2 Zusammenfassung „Einstieg in Icinga2“ schreiben
13.07.16	28	18:00	20:30	0,00	2,50	2,50	Icinga Master installieren, versuchen CPU komplett auszulasten um Funktionalität des CPU Checks anzuschauen sowie die Icingaweb2 Oberfläche genauer anschauen
16.08.16	33	12:00	14:00	0,00	2,00	2,00	Teambesprechung (Fortschritte besprechen, Aufgaben aufteilen)
16.08.16	33	18:00	21:00	0,00	3,00	3,00	Erneutes einlesen in Icinga2 (Buch) wegen Urlaub
23.08.16	34	10:00	12:00	0,00	2,00	2,00	Testsensor als Host anlegen (Problem beim Hostnamen wegen Anführungszeichen), ping-service anlegen sowie Krell darüber informieren



24.08.16	34	10:00	12:00	0,00	2,00	2,00	Mit Testsensor "Temperatursensor HWG-Ste" befassen
27.08.16	34	12:00	14:00	0,00	2,00	2,00	Information von Schrenk: "Backend icinga is not running" -> Fehlersuche
29.08.16	35	13:30	15:30	0,00	2,00	2,00	Fehlersuche für den Fehler "Backend icinga is no running"
31.08.16	35	12:00	14:00	0,00	2,00	2,00	Fehlersuche, Information an Krell (Information aus Internet: Snapshotpakete sollen installiert werden)
03.09.16	35	12:00	13:00	0,00	1,00	1,00	Gespräch mit Schrenk: Er will sich dem Fehler widmen
09.09.16	36	7:50	10:35	2,75	0,00	2,75	Vortrag von Resel über Diplomarbeitspräsentationen, Gespräch mit Betreuer über den Fortschritt sowie sich dem Fehler weiter widmen
09.09.16	36	10:35	11:30	0,00	0,92	0,92	Fehlersuche
09.09.16	36	11:30	13:10	1,67	0,00	1,67	Fehlersuche -> keine Lösung gefunden, Krell informieren und wahrscheinlich Server neu aufsetzen
14.09.16	37	14:00	16:00	0,00	2,00	2,00	Präsentation vorbereiten
15.09.16	37	17:00	19:30	0,00	2,50	2,50	Icinga2 und Icingaweb2 deinstallieren vom Server, Präsentation vorbereiten
16.09.16	37	7:50	10:35	2,75	0,00	2,75	Kickoff-Präsentation, Icinga2 und Icingaweb2 neu installieren
16.09.16	37	10:35	11:30	0,00	0,92	0,92	Mysql Fehler (mysql.user doesn't exist) wenn mysql gestartet werden soll; mysql neu installieren --> Fehler immer noch vorhanden
20.09.16	38	12:20	13:20	1,00	0,00	1,00	Versuchen alle Icinga2 und Mysql Pakete zu deinstallieren --> Krell fragen ob er Server neu aufsetzen kann
20.09.16	38	14:50	16:30	1,67	0,00	1,67	Icinga2 Buch lesen da Server neu aufgesetzt werden muss
23.09.16	38	7:50	10:35	2,75	0,00	2,75	Icinga2 und Icingaweb2 neu installieren
23.09.16	38	10:35	11:30	0,00	0,92	0,92	Fehler während der Installation beheben
23.09.16	38	11:30	13:30	2,00	0,00	2,00	Icinga2 und Icingaweb2 neu installiert --> Fehlermeldung "Backend icinga is not running" immer noch da
26.09.16	39	18:30	21:00	0,00	2,50	2,50	Fehler "Backend icinga is not running" beheben, Temperatursensor in Icinga2 und Icingaweb2 einbinden, SNMP Abfrage für Temperatursensor schreiben --> Erfolg
30.09.16	39	7:50	10:35	2,75	0,00	2,75	Dokumentation "Wiki_1" für Krell schreiben
30.09.16	39	10:35	11:30	0,00	0,92	0,92	Schrenk bei diversen Problemen helfen
30.09.16	39	11:30	13:10	1,67	0,00	1,67	Icinga2 Buch lesen (mit Downtimes befassen)
03.10.16	40	16:30	19:00	0,00	2,50	2,50	Icinga2 Buch lesen (Verteilte Überwachung)



06.10.16	40	17:00	21:00	0,00	4,00	4,00	Im Internet suchen wie man Überwachungen grafisch darstellen könnte --> Graphite --> Installieren mit mehreren Anleitungen aus dem Internet --> Fehlschlag, Apache lässt sich nicht mehr starten
07.10.16	40	7:50	10:35	2,75	0,00	2,75	Fehler von Apache beheben --> Standardport wurde während der Installation von Graphite auf 8000 gestellt welcher gesperrt ist (auf dem Firmenserver)
07.10.16	40	10:35	11:30	0,00	0,92	0,92	Graphite installieren (https://www.youtube.com/watch?v=y39RHFZ3vhl), Krell informieren dass Port 2003 gebraucht wird
07.10.16	40	11:30	13:10	1,67	0,00	1,67	Installationsanleitung für Graphite ist auch im Icinga2 Buch vorhanden --> Alle Pakete die von dem Youtube-Video installiert wurden wieder entfernen und alles aus dem Buch installieren
10.10.16	41	14:30	17:00	0,00	2,50	2,50	Persönliche Dokumentation anfangen; alle Links zusammen suchen; alles zusammenschreiben
14.10.16	41	7:50	10:35	2,75	0,00	2,75	Ubuntu 16.04 neu installieren da es Systemfehler gibt, Icinga2 und Icingaweb2 installieren
14.10.16	41	10:35	11:30	0,00	0,92	0,92	Graphite lokal installieren, Fehler 'Couldn't listen on 0.0.0.0:2003 [Errno98] Address already in use.';
14.10.16	41	11:30	13:10	1,67	0,00	1,67	Fehlersuche --> erfolglos
19.10.16	42	15:00	15:30	0,00	0,50	0,50	Dokumentation schreiben
25.10.16	43	18:00	20:00	0,00	2,00	2,00	Dokumentation schreiben
26.10.16	43	9:00	11:00	0,00	2,00	2,00	Dokumentation schreiben
27.10.16	43	17:00	23:00	0,00	6,00	6,00	Graphite auf Server installieren; Fehler 'Couldn't listen on 0.0.0.0:2003: [Errno 98] Address already in use.'; Fehlersuche --> Fehlersuche nicht erfolgreich, Krell bitten Server neu aufzusetzen; Graphite erneut lokal installieren
28.10.16	43	7:50	10:35	2,75	0,00	2,75	Graphite nach Buch lokal installieren, FEHLER
28.10.16	43	10:35	11:30	0,00	0,92	0,92	Fehlersuche, Ubuntu & Icinga neu installieren
28.10.16	43	11:30	13:10	1,67	0,00	1,67	Graphite nach https://www.digitalocean.com/community/tutorials/how-to-install-and-use-graphite-on-an-ubuntu-14-04-server installieren --> ERFOLG!!!!!!
29.10.16	43	12:30	14:30	0,00	2,00	2,00	Graphite-Web anschauen; Unterordner "icinga2" ist zwar da jedoch werden die Services nicht angezeigt (müssen wahrscheinlich irgendwo extra für Graphite



								definiert werden); Graphite Modul in Icingaweb2 freischalten, statt Graph wird jedoch "No Text" angezeigt --> Fehlersuche folgt
29.10.16	43	19:00	22:00	0,00	3,00	3,00		Mit Graphite befassen (wie werden Services definiert, wie werden Graphen in Icingaweb2 und in Graphite-Web angezeigt usw.)
02.11.16	44	12:00	14:00	0,00	2,00	2,00		Graphite-Video schauen; Graphite Modul in Icinga2 aktivieren --> Graphen werden nun endlich in Graphite angezeigt (siehe Screenshot von CPU Auslastung)
02.11.16	44	19:30	21:00	0,00	1,50	1,50		Versuchen auf Server MySQL und MariaDB zu reparieren (MariaDB löschen) da Krell Server nicht zurücksetzt; Problem: dpkg unter /usr/bin wurde gelöscht --> apt-get upgrade usw. funktioniert alles nicht, versuchen dies zu reparieren --> erfolgslos; müssen warten auf zurücksetzen durch Krell
04.11.16	44	7:50	10:35	2,75	0,00	2,75		Fehler: /usr/bin/dpkg wurde gelöscht --> kopieren von lokalem Ubuntu: Erfolg; Nach Fehler auf Server suchen --> MySQL und MariaDB können nicht gestartet werden, bei der Deinstallation von MySQL tritt Fehler auf: MySQL wird nie komplett gelöscht
04.11.16	44	10:35	11:30	0,00	0,92	0,92		MySQL komplett löschen (Erfolg)
04.11.16	44	11:30	13:10	1,67	0,00	1,67		Graphite auf Server installieren --> Erfolg; Werte von Temperatursensor anzeigen lassen funktioniert nicht (Aufgabe bis nächsten Freitag: Temperaturwerte anzeigen lassen)
05.11.16	44	16:00	19:00	0,00	3,00	3,00		Mit Graphite befassen (warum wird keine einzige Kurve angezeigt) --> Graphite-Web ignorieren und Graphite Modul für Icingaweb2 aktivieren, Kurven von ping und apt usw. können angezeigt werden, Temperatur aufgrund von SNMP nicht
06.11.16	44	10:00	11:00	0,00	1,00	1,00		Fehlersuche --> SNMP Werte werden nicht als Performance Daten ausgegeben, daher auch kein Graph
06.11.16	44	19:30	20:45	0,00	1,25	1,25		Erneute Fehlersuche; Plugin funktioniert an sich, SNMP Abfrage auch, nur die Ausgabe der Performance Daten scheint "kaputt zu sein"
07.11.16	45	14:00	14:30	0,00	0,50	0,50		Alte Abfragen wieder hinzufügen (für den PC, z.B.: Festplattenspeicher, SSH usw.) um für Tag der offenen Tür mehrere Abfragen zu haben
08.11.16	45	13:30	16:35	3,08	0,00	3,08		Dokumentation schreiben
09.11.16	45	14:00	15:30	0,00	1,50	1,50		Fehlersuche; Mögliche Lösung: Es sind keine Werte gesetzt wann der Service den Zustand "Warning" und "Critical" einnehmen soll --> Diese Parameter setzen, Warning ab 30°C und Critical ab 40°C (keine genauen Werte, nur Zufallswerte) --> Erfolg, Ausgabe ist nun



								Integer und somit Perfomance Data --> Graphite Graph funktioniert
10.11.16	45	18:00	18:30	0,00	0,50	0,50		Test-Email-Adresse anlegen (da.wks.test@gmail.com) --> mit Email-Notifications befassen
11.11.16	45	7:50	10:35	2,75	0,00	2,75		Gespräch mit Betreuer --> Check_MK installieren um Unterschied zu Icinga zu sehen; Installation beginnen
11.11.16	45	10:35	11:30	0,92	0,00	0,92		Check_MK versuchen zu installieren --> Fehlschlag
11.11.16	45	11:30	13:10	1,67	0,00	1,67		Check_MK versuchen zu installieren --> Fehlschlag
17.11.16	46	14:55	16:35	1,67	0,00	1,67		Email Notification für Icinga einrichten (http://www.modius-techblog.de/linux/icinga-2-notifications/) --> Fehler Emails werden nicht gesendet
17.11.16	46	18:00	20:00	0,00	2,00	2,00		Fehlersuche --> postfix installieren, Email Benachrichtigungen funktionieren für Temperatur Service und alle anderen Services
18.11.16	46	7:50	10:35	2,75	0,00	2,75		Gespräch mit Betreuer, check_mk soll installiert werden; Versuchen zu installieren bis 8:40, danach aufbauen für Tag der offenen Tür
18.11.16	46	10:35	12:20	0,00	1,75	1,75		Aufbauen für Tag der offenen Tür
18.11.16	46	13:10	17:00	0,00	3,83	3,83		Tag der offenen Tür
19.11.16	46	8:30	12:30	0,00	4,00	4,00		Tag der offenen Tür
21.11.16	47	17:00	20:00	0,00	3,00	3,00		Check_MK versuchen zu installieren; Erkenntnis: Kann nicht für Ubuntu 16.04 installiert werden; Dokumentation für Krell schreiben (Milestone 1: Erweitern um Graphite & E-Mail Benachrichtigung)
22.11.16	47	14:00	14:50	0,83	0,00	0,83		Dokumentation für Krell fertig stellen
24.11.16	47	18:00	22:00	0,00	4,00	4,00		Ubuntu 14.04 auf Laptop installieren um Check_mk lokal zu installieren; Check_mk installieren; Verteilte Überwachung aus Icinga2 Buch lesen
25.11.16	47	7:50	10:35	2,75	0,00	2,75		Gespräch mit Betreuer: Check_MK herzeigen, Aufgabe: Verteiltes Netzwerk zuhause aufbauen und beide Programme verwenden; Check_MK Tutorials lesen
25.11.16	47	10:35	11:30	0,00	0,92	0,92		Betreuungsprotokolle nachschreiben
25.11.16	47	11:30	13:10	1,67	0,00	1,67		Betreuungsprotokolle nachschreiben
25.11.16	47	21:30	0:00	0,00	2,50	2,50		Videos von Betreuer schauen (Verteilte Überwachung in Icinga2)
27.11.16	47	15:15	19:05	0,00	3,83	3,83		Plan: Laptop ist Master, PC ist Slave (Windows & Ubuntu durch Virtualbox); Virtualbox auf Stand-PC installieren mit Ubuntu 14.04 (damit Check_MK laufen kann); Problem bei



							Installation: Stand-PC hat AMD-CPU (nur 32 bit Ubuntu möglich); Ubuntu nach 1. Installation nicht richtig gebootet --> erneute Instalaltion; Icinga2 & Icingaweb2 auf Stand-PC installieren (Ubuntu); Check_MK installieren auf Stand-PC (Ubuntu) Erkenntnis: Check_MK kann auch unter Ubuntu 16.04 installiert werden, konnte damals nicht installiert werden da eine alte Version des Programms runtergeladen wurde; Icinga2 auf Windows des Stand-PC installieren
28.11.16	48	17:50	18:30	0,00	0,67	0,67	Icinga2 auf Laptop installieren (da nicht mehr vorhanden; Grund: unbekannt); Ubuntu 14.04 System Fehler --> Neuinstallation von Ubuntu (sobald wie möglich tun)
28.11.16	48	19:10	20:40	0,00	1,50	1,50	Ubuntu 14.04 auf Laptop installieren; Icinga2 und Icingaweb2 installieren; Master-Setup auf Laptop installieren; Versuchen Windows-Satellite als Satellite einzusetzen und zu überwachen --> Fehler: Windows Host kann den Master nicht finden, wahrscheinlich Fehler in der Icinga2 Node Wizard Konfiguration
29.11.16	48	19:20	21:30	0,00	2,17	2,17	Fehlersuche --> Seite finden; Icinga2 Node Wizard auf Laptop löschen & Verteilte Überwachung mit Windows Client nach der Seite zu installieren; Erfolg, Windows Satellite wurde in Icingaweb2 eingebunden und abfragen werden durchgeführt; Nächster Schritt: Wie können diese Services verändert werden + Test (was passiert wenn die Verbindung zwischen Master und Satellite ausfällt)
30.11.16	48	14:30	18:00	0,00	3,50	3,50	Icinga2 Fehler --> Icinga2 lässt sich nicht mehr starten, Fehler "Satellite is in more than one zone" obwohl Satellite gar nicht mehr angelegt ist --> Fehlersuche; Resultat: Icinga2 neu installieren (alle Dateien zurücksetzen); Icinga2 & Icingaweb2 & Icinga2 Node Wizard auf Laptop neu installieren; Währenddessen--> Ubuntu auf Stand-PC neu installieren da Virtualbox nicht angepingt werden kann; BIOS erkennt USB-Stick nicht --> Fehlersuche; Erfolg --> Ubuntu sowie Icinga2 & Icingaweb2 auf Stand-PC installieren
01.12.16	48	10:35	11:25	0,83	0,00	0,83	Stundentafel "verbessern" (Einträge neu schreiben bzw. bearbeiten)
01.12.16	48	18:00	21:00	0,00	3,00	3,00	Icinga2 Node Wizard auf Stand-PC installieren (Ubuntu Satellite Setup); Fehler auf Master Laptop (Master wurde in mehreren Dateien definiert) --> Dateien zurücksetzen; Fehler bei Satellite hinzufügen: Unauthorized



02.12.16	48	7:50	10:35	2,75	0,00	2,75	Aufbauen von verteiltem Netzwerk (Master und Windows Satellite; auf beiden Icinga2 und nur auf Master Icingaweb2 installieren)
02.12.16	48	10:35	11:30	0,00	0,92	0,92	Windows PC kommt nicht ins Internet --> Fehlersuche
02.12.16	48	11:30	13:10	1,67	0,00	1,67	Fehler behoben (Windows Firewall deaktiviert); Einbinden von Windows Satellite in Master Icingaweb2 --> Fehler: nicht erreichbar (jedoch als normaler Host erreichbar)
02.12.16	48	19:20	21:00	0,00	1,67	1,67	Icinga2 auf Laptop zurücksetzen (um Standardeinstellungen zu haben) und in den Master erneut Stand-PC einbinden (mit Windows) um zu testen, ob Fehler von Schule an Netzwerk oder PC liegt; Fehler lag nicht am PC; Test (Verbindung trennen) lieferte nicht das erwünschte Szenario --> Icinga hat nichts getan als die Verbindung wieder hergestellt wurde (außer weiter Checks ausführen)
03.12.16	48	19:00	20:00	0,00	1,00	1,00	Mit Unterschied zwischen Top-Down und Bottom-Up befassen
05.12.16	49	18:00	20:00	0,00	2,00	2,00	Schul-PC in Netzwerk zuhause einbinden --> Icinga Überwachung funktioniert
16.12.16	50	7:50	10:35	2,75	0,00	2,75	Schul-PC nochmal in Schule ausprobieren; Ubuntu und Icinga auf Master installieren
16.12.16	50	10:35	11:30	0,00	0,92	0,92	Icinga2-ido-mysql Fehler
16.12.16	50	11:30	13:10	1,67	0,00	1,67	Icinga2-ido-mysql Fehler
17.12.16	50	12:30	16:00	0,00	3,50	3,50	Zuhause Icinga2 System erneut aufsetzen um Bottom Up hinzukriegen --> Laptop ist Master, Stand-PC ist Slave; E-Mail Benachrichtigung am Master einrichten & Test über Satellite-CPU ob E-Mail auch für Satellites gesendet wird; Postfix wurde installiert aber es werden keine E-Mails gesendet (weder über Icinga noch über mail -s ...)
18.12.16	50	13:00	15:30	0,00	2,50	2,50	Versuchen E-Mails von Laptop per Postfix zu versenden, erfolgslos --> Fehlersuche ebenfalls erfolgslos
27.12.16	52	11:00	15:00	0,00	4,00	4,00	Verteilte Überwachung V2 schreiben; E-Mail Benachrichtigung erstmal vergessen --> Versuchen Bottom Up hinzukriegen --> erfolgslos; Graphite installieren um zu schauen was mit Graph passiert wenn Überwachung über einen Zeitraum ausfällt; Graphite Fehler bei Installation
01.01.17	52	16:00	18:00	0,00	2,00	2,00	Verteilte Überwachung (Icinga2) wird auf Schrenk verschoben da von Krell Satellite PCs zur Verfügung gestellt wurden; Check_MK ausprobieren (Verteilte Überwachung)
03.01.17	1	12:00	16:00	0,00	4,00	4,00	Mit Check_MK befassen



04.01.17	1	11:00	13:30	0,00	2,50	2,50	Mit Check_MK befassen
04.01.17	1	18:00	20:00	0,00	2,00	2,00	Check_MK versuchen Host einzufügen, Fehlschlag da Funktion von Check_MK noch nicht ganz verstanden
05.01.17	1	11:00	14:00	0,00	3,00	3,00	Mit Check_MK befassen
05.01.17	1	19:00	21:00	0,00	2,00	2,00	Check_MK einrichten sowie einen weiteren PC aus dem Netzwerk als Host einfügen
06.01.17	1	8:00	11:00	0,00	3,00	3,00	Persönliche Dokumentation weiterschreiben
09.01.17	2	18:00	20:30	0,00	2,50	2,50	Unterschiede zwischen Check_MK und Icinga2 feststellen; Für Icinga2 entscheiden da es in der Open Source Version mehr Funktionen hat und die Oberfläche (subjektiv betrachtet) besser scheint
13.01.17	2	7:50	10:35	2,75	0,00	2,75	Icinga2 auf Satellite von der Firma installieren --> Satellite ist DOWN gegangen, keine Verbindung möglich; Satellite geht wieder
13.01.17	2	10:35	11:30	0,00	0,92	0,92	Icinga2 installieren; Satellite in Master einbinden --> Erfolg
13.01.17	2	11:30	13:10	1,67	0,00	1,67	Network Card versuchen zu überwachen --> befassen mit APC UPS Network Card 2
13.01.17	2	14:00	18:00	0,00	4,00	4,00	Nach OIDs und MIBs von Network Card suchen --> Suche verlief nicht erfolgreich
14.01.17	2	12:00	14:00	0,00	2,00	2,00	Mit Network Card befassen um Funktion zu verstehen
14.01.17	2	22:00	0:00	0,00	2,00	2,00	Präsentation vorbereiten für interne Präsentation
19.01.17	3	18:00	21:00	0,00		3,00	APC UPS Network Management Card 2 in Icinga Überwachung einbinden; Fehlschlag (Master akzeptiert Services zu Card nicht); Präsentation vorbereiten
20.01.17	3	7:50	10:35	2,75		2,75	Präsentation halten und anhören von Kollegen
20.01.17	3	10:35	11:30	0,00	0,92	0,92	Präsentation verbessern (Veränderungsvorschläge von Betreuer)
20.01.17	3	11:30	13:10	1,67	0,00	1,67	Präsentation verbessern (Veränderungsvorschläge von Betreuer)
20.01.17	3	18:00	20:00	0,00	2,00	2,00	Logo für Präsentation/Diplomarbeit erstellen (V1)
21.01.17	3	10:00	12:00	0,00	2,00	2,00	Präsentation fertig stellen (für bisherigen Stand)
22.01.17	3	13:00	15:00	0,00	2,00	2,00	Weiter nach Möglichkeit suchen eigene Services bei Satellite einzubinden --> ohne Erfolg
23.01.17	4	15:00	17:00	0,00	2,00	2,00	Logo V2 entwickeln
24.01.17	4	12:20	16:35	4,25	0,00	4,25	Windows Satellite PC mit Schrenk einbinden (ihm beibringen); persönliche Doku weiterschreiben
24.01.17	4	19:00	21:00	0,00	2,00	2,00	Icinga2 Konfiguration auf Standard zurücksetzen und neu konfigurieren da



							Satellite Konfiguration die Icinga2 Konfiguration zerstört hat --> alte Konfiguration wieder herstellen; Fehler: Icinga2 wird auf Server und auf Satellite nicht mehr richtig installiert (/etc/icinga2 fehlt); Icinga2 auf Server und auf Ubuntu neu installieren
25.01.17	4	19:00	20:00	0,00	1,00	1,00	Icinga2 Verteiltes Netzwerk erneut aufbauen --> Fehler mit Apache2 und anderen Programmen; Krell fragen ob er den Server neu aufsetzen kann (schnellste Möglichkeit)
27.01.17	4	7:50	10:35	2,75	0,00	2,75	Aufbauen für Tag der offenen Tür
27.01.17	4	10:35	11:30	0,00	0,92	0,92	Vorbereiten für Tag der offenen Tür
27.01.17	4	11:30	13:10	1,67	0,00	1,67	Vorbereiten für Tag der offenen Tür
28.01.17	4	8:00	12:30	0,00	4,50	4,50	Tag der offenen Tür
31.01.17	5	12:20	14:50	2,50	0,00	2,50	Logo V2 entwickeln
01.02.17	5	15:30	16:30	0,00	1,00	1,00	Hauptserver wurde neu aufgesetzt --> Icinga2, Icingaweb2 installieren sowie Windows PC einbinden
03.02.17	5	7:50	10:35	2,75	0,00	2,75	Ubuntu PC einbinden in Überwachung, mit Bottom Up und Top Down befassen
06.02.17	6	14:00	17:00	0,00	3,00	3,00	Mit Bottom Up und Top Down befassen (https://docs.icinga.com/icinga2/latest/doc/module/icinga2/toc#/icinga2/latest/doc/module/icinga2/chapter/distributed-monitoring); Doku der letztjährigen Diplomarbeiten lesen und eigene DA Doku beginnen: Kapitel 1 schreiben
07.02.17	6	12:00	15:00	0,00	3,00	3,00	Dokumentation schreiben: Kapitel 1 lesen und Kapitel 2 bis 2.5 Icingaweb2 schreiben; Network Card einbinden über Ubuntu PC (Keine Satellite Services)
08.02.17	6	12:00	15:00	0,00	3,00	3,00	Dokumentation: Alles bisher Geschriebene lesen und verbessern
09.02.17	6	10:20	12:20	0,00	2,00	2,00	Dokumentation weiterschreiben bis 3.3 MIB
09.02.17	6	12:30	14:30	0,00	2,00	2,00	Satellite-Service für CISCO (Windows) und Network Card (Ubuntu) schreiben; Beides wird zwar eingebunden aber leider nur Pending Zustand --> Fehlersuche; Erkenntnis: Check muss nur auf Satellite in conf.d sein um Bottom Up zu haben --> Problem gelöst
09.02.17	6	16:15	17:15	0,00	1,00	1,00	Windows Satellite versuchen CISCO überwachen zu lassen
10.02.17	6	12:30	13:30	0,00	1,00	1,00	Icinga 2.4.4. downloaden --> In Windows Satellite alles CISCO Router einrichten --> Erfolg! Doku_Wiki2 schreiben und Krell mitteilen dass letzter Milestone begonnen werden kann



17.02.17	7	7:50	10:35	2,75	0,00	2,75	Graphite installieren
17.02.17	7	10:35	11:30	0,00	0,92	0,92	Graphite installieren
17.02.17	7	11:30	13:10	1,67	0,00	1,67	Graphite installieren
17.02.17	7	18:00	22:00	0,00	4,00	4,00	Doku_Wiki2 für Krell schreiben
20.02.17	8	16:00	18:00	0,00	2,00	2,00	Dokumentation schreiben
21.02.17	8	18:00	21:00	0,00	3,00	3,00	Dokumentation schreiben
23.02.17	8	18:00	20:00	0,00	2,00	2,00	Dokumentation schreiben
24.02.17	8	7:50	10:35	2,75	0,00	2,75	Gespräch mit Betreuer über die Dokumentation; Schrenk bei SNMP Problem helfen
24.02.17	8	10:35	11:30	0,00	0,92	0,92	Schrenk bei SNMP Problem helfen
24.02.17	8	11:30	13:10	1,67	0,00	1,67	Schrenk bei SNMP Problem helfen
24.02.17	8	13:30	17:30	0,00	4,00	4,00	Dokumentation schreiben
27.02.17	9	16:15	17:45	0,00	1,50	1,50	Dokumentation schreiben
28.02.17	9	18:00	19:00	0,00	1,00	1,00	Dokumentation schreiben
02.03.17	9	17:00	20:00	0,00	3,00	3,00	Dokumentation schreiben
03.03.17	9	8:00	12:00	0,00	4,00	4,00	Dokumentation schreiben
08.03.17	10	13:00	16:00	0,00	3,00	3,00	Dokumentation schreiben
09.03.17	10	15:00	18:00	0,00	3,00	3,00	Dokumentation schreiben
10.03.17	10	7:50	10:35	2,75	0,00	2,75	Apache2 auf Hauptserver funktioniert nicht; alles deinstallieren und alles neu installieren; Apache funktioniert aber der Befehl icingacli nicht --> Icingaweb2 kann nicht installiert werden
10.03.17	10	10:35	11:30	0,00	0,92	0,92	Dokumentation schreiben
10.03.17	10	11:30	13:10	1,67	0,00	1,67	Icingacli Kommando war beschädigt --> Icingaweb2 installieren jedoch werden alle Satellites als DOWN bezeichnet obwohl sie online sind und angepingt werden können
11.03.17	10	17:00	19:00	0,00	2,00	2,00	Dokumentation schreiben
12.03.17	10	18:00	20:00	0,00	2,00	2,00	Satellites sind DOWN in Icinga2 obwohl sie UP sind --> Versuchen zu lösen --> Ohne Erfolg
13.03.17	11	15:00	17:00	0,00	2,00	2,00	Satellites sind DOWN in Icinga2 obwohl sie UP sind --> Versuchen zu lösen mit Satellites entfernen und neu hinzufügen --> Erfolg; Graphite macht aber einen Fehler --> Wenn apache2-graphite Seite aktiviert wird kann Apache2 nicht mehr gestartet werden



14.03.17	11	18:00	21:00	0,00	3,00	3,00	Graphite-Fehler lösen (WSGI war nicht installiert); Graphite-web ist nun verfügbar jedoch kann Graphite Plugin für Icinga2 nicht gestartet/enabled werden --> Fehlersuche; Dokumentation schreiben
15.03.17	11	15:00	17:00	0,00	2,00	2,00	Dokumentation schreiben
16.03.17	11	18:00	21:00	0,00	3,00	3,00	Dokumentation schreiben
17.03.17	11	7:50	10:35	2,75	0,00	2,75	Graphite versuchen zu installieren
17.03.17	11	10:35	11:30	0,00	0,92	0,92	Graphite versuchen zu installieren --> Platzhalter sind eingefügt aber die Graphen fehlen immer noch (No Data wird angezeigt)
17.03.17	11	11:30	13:10	1,67	0,00	1,67	Graphite versuchen zu installieren (nun werden statt No Data leere Felder angezeigt)
17.03.17	11	13:30	14:30	0,00	1,00	1,00	Graphite versuchen zu installieren (ohne Erfolg; es funktioniert einfach nicht!!!) --> Entschluss: Graphite wird nicht installiert, es hat bereits 1 mal funktioniert und es wurden genug Screenshots für die Doku aufgenommen
18.03.17	11	17:15	18:15	0,00	1,00	1,00	Dokumentation analysieren sowie Stundentafel dem Betreuer schicken
19.03.17	11	10:30	11:30	0,00	1,00	1,00	Dokumentation analysieren und verbessern
19.03.17	11	19:30	20:30	0,00	1,00	1,00	Dokumentation analysieren und verbessern
21.03.17	12	18:00	19:00	0,00	1,00	1,00	Screenshot von Vorprojekt suchen (MS2) --> nicht auffindbar; Dokumentation weiterschreiben
23.03.17	12	10:00	12:30	0,00	2,50	2,50	Dokumentation weiterschreiben
23.03.17	12	14:00	16:00	0,00	2,00	2,00	Dokumentation analysieren und verbessern
23.03.17	12	19:30	20:00	0,00	0,50	0,50	Dokumentation analysieren und verbessern
28.03.17	13	18:00	21:00	0,00	3,00	3,00	Verbesserte Dokumentation von Betreuer erhalten --> Verbesserungsvorschläge in die Dokumentation einarbeiten
29.03.17	13	18:00	20:00	0,00	2,00	2,00	Verbesserungsvorschläge in die Dokumentation einarbeiten
30.03.17	13	20:00	21:00	0,00	1,00	1,00	Zusammengefügte Dokumentation analysieren sowie verbessern (von Schrenk zusammengefügt)
31.03.17	13	7:50	10:35	2,75	0,00	2,75	Dokumentation analysieren und verbessern
31.03.17	13	10:35	11:30	0,00	0,92	0,92	Dokumentation analysieren und verbessern
31.03.17	13	11:30	13:10	1,67	0,00	1,67	Dokumentation analysieren und verbessern
03.04.17	14	16:30	18:30	0,00	2,00	2,00	Dokumentation analysieren und verbessern

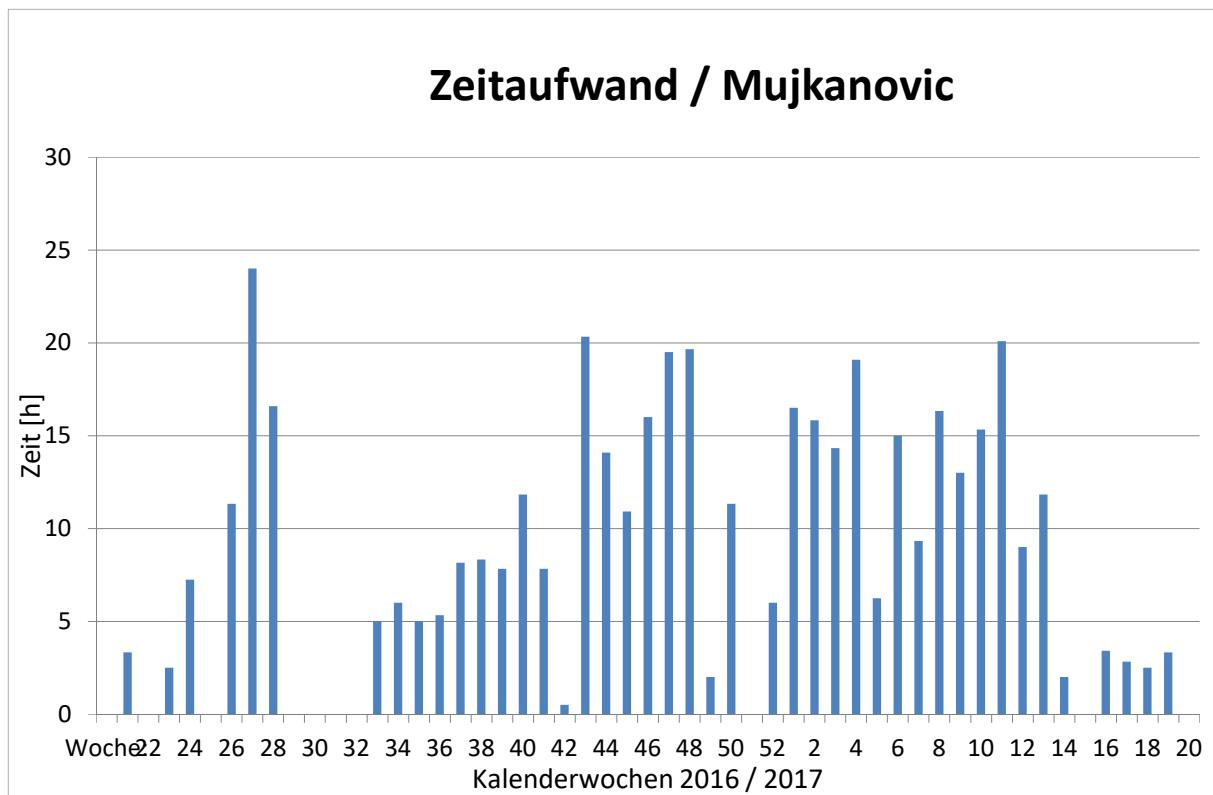


Abbildung 86 - Zeitaufwand Mujkanovic 13

Gesamt: 446,67h

Freizeit: 305,94h

Schule: 137,73h



6.2 Schrenk

01.03.16	9	8:00	11:30	3,50	0,00	3,50	Vorstellung der Diplomarbeiten und Auswahl des Themas
15.03.16	11	9:45	10:15	0,50	0,00	0,50	Besprechung mit Betreuer; Datenaustausch und Festlegung der Ziele
17.03.16	11	13:30	13:45	0,25	0,00	0,25	Besprechung mit Betreuer bzgl. Aufgabe in den Osterferien
26.03.16	12	9:00	11:00		2,00	2,00	Installation Ubuntu 14.04LTS; befassen mit Icinga Dokumentation
26.03.16	12	17:00	18:00		1,00	1,00	Installation Icinga2 lt. Dokumentation begonnen
							Installation Icinga2 lt. Dokumentation fertiggestellt; http://docs.icinga.org/icinga2/latest/doc/module/icinga2/toc#!/icinga2/latest/doc/module/icinga2/chapter/getting-started#installing-icinga2
20.04.16	16	13:10	16:35	3,42	0,00	3,42	
27.04.16	17	13:10	16:35	3,42	0,00	3,42	Ersten Videos von Betreuer geschaut
11.05.16	19	13:10	16:35	3,42	0,00	3,42	Installation Icingaweb2; https://github.com/Icinga/icingaweb2/blob/master/doc/02-Installation.md
25.05.16	21	13:10	16:35	3,42	0,00	3,42	Videos von Betreuer geschaut
01.06.16	22	13:10	16:35	3,42	0,00	3,42	Icingaweb2 Konfiguration
08.06.16	23	14:00	16:35	2,58	0,00	2,58	Installation der PHP5 Module über Terminal
15.06.16	24	13:00	16:15	3,25	0,00	3,25	Besprechung mit Firma in Simonsfeld
17.06.16	24	8:00	10:00		2,00	2,00	Mit Distributed Monitoring befasst; http://docs.icinga.org/latest/de/distributed.html
18.06.16	24	19:00	20:15		1,25	1,25	Mit Distributed Monitoring befasst; http://docs.icinga.org/latest/de/distributed.html
18.06.16	24	22:15	22:45		0,50	0,50	Mit Distributed Monitoring befasst; http://docs.icinga.org/latest/de/distributed.html
27.06.16	26	7:50	12:30	4,67	0,00	4,67	In Icinga2 Dokumentation einlesen
28.06.16	26	7:50	12:30	4,67	0,00	4,67	In Icinga2 Dokumentation einlesen
07.07.16	27	9:00	11:30		2,50	2,50	Versuch Ververbindung zu erstellen (Putty) -> Fehlgeschlagen; Internetverbindung getestet, Proxy, Port-Freigabe, anderes Netzwerk, ...
07.07.16	27	12:30	16:00		3,50	3,50	Versuch Ververbindung zu erstellen (Putty) -> Fehlgeschlagen; Internetverbindung getestet, Proxy, Port-Freigabe, anderes Netzwerk, ...
08.07.16	27	7:50	11:30		3,67	3,67	Fehlersuche Verbindung -> Firewall & Virenschutz aus, X11-forwarding & xming -> kein Erfolg; Michael Krell informiert
16.07.16	28	9:00	11:30		2,50	2,50	neuen Laptop aufgesetzt und für Diplomarbeit vorbereitet, Daten der DA kopiert
16.07.16	28	16:00	20:00		4,00	4,00	neu Laptop fertiggestellt
20.07.16	29	7:00	10:00		3,00	3,00	Ubuntu 14.04, Icinga2 und Icingaweb2 auf neuem Laptop installiert. (putty, xming installiert, nötige Einstellungen getroffen)
25.07.16	30	19:00	21:00		2,00	2,00	mit Icinga2 Dokumentation befasst



26.07.16	30	19:00	20:15		1,25	1,25	mit Icinga2 Dokumentation befasst
27.07.16	30	22:15	0:00		1,75	1,75	mit Icinga2 Dokumentation befasst
27.07.16	30	20:00	21:00		1,00	1,00	Icinga2 Buch lesen
28.07.16	30	7:00	10:00		3,00	3,00	Icinga2 Buch lesen
16.08.16	33	12:00	14:00		2,00	2,00	Teambesprechung (Fortschritte, Aufgaben...)
31.07.16	30	19:00	20:45		1,75	1,75	Icinga2 Buch lesen
18.08.16	33	8:00	12:00		4,00	4,00	Icinga2 Buch lesen
19.08.16	33	16:00	18:00		2,00	2,00	Icinga2 Buch lesen
23.08.16	34	17:00	20:00		3,00	3,00	Fehlersuche nach Update (sources.list, Neustart, ...)
03.09.16	35	12:00	13:00		1,00	1,00	Teambesprechung bzgl Fehler
26.08.16	34	13:00	15:00		2,00	2,00	Fehlersuche nach Update (sources.list, Neustart, ...)
09.09.16	36	7:50	13:10	4,42	0,91	5,33	Gespräch mit Betreuer, Fehlersuche
14.09.16	37	15:15	17:15		2,00	2,00	Präsentation Kick off
15.09.16	37	17:30	19:30		2,00	2,00	Deinstallieren und löschen Icinga2, Icingaweb2 und Apache wegen Fehler
15.09.16	37	20:30	22:00		1,50	1,50	Präsentation Kick off
16.09.16	37	7:50	13:10	4,42	0,91	5,33	Präsentationen, In InfluxDB einlesen
20.09.16	38	12:20	13:10	0,83	0,00	0,83	Installation InfluxDB in VM
20.09.16	38	14:50	16:35	1,75	0,00	1,75	Installation InfluxDB in VM
29.09.16	39	19:45	20:30		0,75	0,75	Ausprobieren InfluxDB; anlegen einer Datenbank
30.09.16	39	7:50	13:10	4,42	0,91	5,33	Eintragen Testwerte, Befassen mit Measurements, Grundbefehle InfluxDB
07.10.16	40	7:50	13:10	4,42	0,91	5,33	Grundlagen Python, erste Testscripts
10.10.16	41	15:00	17:20		2,33	2,33	Python Videos, Tutorialspoint.com Grundlagen lernen
14.10.16	41	7:50	13:10	4,42	0,91	5,33	Python Script zur Temperaturabfrage, Informationsbeschaffung Kompatibilität Python + InfluxDB
15.10.16	41	9:00	10:30		1,50	1,50	snmpget Temperaturwert abfragen und String formatieren
21.10.16	42	7:50	13:10	4,42	0,91	5,33	Zusammenfassung diverser Quellen, suche InfluxDB in Python zu importieren, erste Tests
23.10.16	42	17:00	17:45		0,75	0,75	Json für InfluxDB lernen
27.10.16	43	16:40	17:30		0,83	0,83	Fehlersuche InfluxDB startet auf Server nicht -> Neuinstallation
28.10.16	43	7:50	13:10	4,42	0,91	5,33	lokale Installation Grafana, Konfiguration, Video angeschaut, Dashboard erstellt, Testdatenbank anlegen
28.10.16	43	16:30	18:00		1,50	1,50	Behebung Fehler "retention policy not found", Testwerte angelegt, Python Test Simulation der snmp-Abfrage
03.11.16	44	16:45	17:25		0,67	0,67	virtuelle Maschine startet nicht -> Fehlerbehebung virtual Disk
04.11.16	44	7:50	13:10	4,16	1,17	5,33	Python Script Temperaturwerte in Datenbank schreiben funktioniert, Testwerte in Grafana anzeigen, mit SNMP in Python befasst



08.11.16	45	13:30	16:35	3,08	0,00	3,08	SNMP in Python Suche, Zusammenfassung diverser Quellen
11.11.16	45	7:50	13:10	4,42	0,91	5,33	installieren NET-SNMP, easysnmp auf Server und schreiben des Scripts zur Abfrage; String Manipulation funktioniert nicht weil kein String, Suche nach Rückgabewert des Befehls
11.11.16	45	16:15	18:30		2,25	2,25	erneutes schreiben Datenbank-&SNMP-Scripts (bei ordnen des Servers versehentlich gelöscht, kein verwendbarer Wiederherstellungspunkt)
11.11.16	45	21:00	22:45		1,75	1,75	Versuche Befehl rms (löschen mit recycle bin um erneutes versehentliches löschen zu verhindern), Befehl selbst gemacht (alias, function)
13.11.16	45	17:00	18:10		1,17	1,17	rms Befehl Recherche; Problem: Pfad des files "rm" ist in PATH Variable eingetragen ./rm wird aber nicht erkannt
13.11.16	45	18:15	19:45		1,50	1,50	rm Befehl wird nicht mehr gefunden -> Recherche; Stackoverflow Question Diskussion
14.11.16	46	22:00	23:25		1,42	1,42	weitere Suche -> Stackoverflow Kommentar (rm file leer) -> besorgen des rm files und auf Server kopieren -> funktioniert
17.11.16	46	18:00	21:30		3,50	3,50	Vorbereitung TdoT, Problem snmp return value kein String -> value ist zu extrahieren -> class easysnmp.variables.SNMPVariable
18.11.16	46	7:50	12:20	4,42	0,08	4,50	Tag der offenen Tür
18.11.16	46	12:20	17:15		4,92	4,92	Tag der offenen Tür
19.11.16	46	7:50	13:00		5,17	5,17	Tag der offenen Tür
22.11.16	47	13:00	14:30		1,50	1,50	neues Measurement für live-Werte, Script in Autostart -> Fehlersuche
24.11.16	47	17:00	18:30		1,50	1,50	Installation und Konfiguration Grafana
25.11.16	47	7:50	13:10	4,42	0,91	5,33	Problem mit Grafana Port, Problem Werte werden nicht angezeigt -> Fehlersuche
25.11.16	47	19:00	21:30		2,50	2,50	überarbeiten des Aufwands-Excels und Dokumentation
27.11.16	47	13:00	14:50		1,83	1,83	Fehlersuche bzgl. Grafana zeigt keine Werte an; Informationen zu verteilter Überwachung beschaffen
28.11.16	48	17:50	18:30		0,67	0,67	Icinga2 auf Laptop installieren (da nicht mehr vorhanden; Grund: unbekannt); Ubuntu 14.04 System Fehler --> Neuinstallation von Ubuntu (sobald wie möglich tun) (gemeinsam mit Osman)
28.11.16	48	19:20	20:30		1,17	1,17	Ubuntu 14.04 auf Laptop installieren; Icinga2 und Icingaweb2 installieren; Master-Setup auf Laptop installieren; Versuchen Windows-Satellite als Satellite einzusetzen und zu überwachen --> Fehler: Windows Host kann den Master nicht finden, wahrscheinlich Fehler in der Icinga2 Node Wizard Konfiguration (gemeinsam mit Osman)



30.11.16	48	14:30	18:00		3,50	3,50	Icinga2 Fehler --> Icinga2 lässt sich nicht mehr starten, Fehler "Satellite is in more than one zone" obwohl Satellite gar nicht mehr angelegt ist --> Fehlersuche; Resultat: Icinga2 neu installieren (alle Dateien zurücksetzen); Icinga2 & Icingaweb2 & Icinga2 Node Wizard auf Laptop neu installieren; Währenddessen--> Ubuntu auf Stand-PC neu installieren da Virtualbox nicht angepingt werden kann; BIOS erkennt USB-Stick nicht --> Fehlersuche; Erfolg --> Ubuntu sowie Icinga2 & Icingaweb2 auf Stand-PC installieren (gemeinsam mit Osman)	
02.12.16	48	7:50	13:10	4,42	0,91	5,33	Installation Ubuntu 16.04 auf Schul-PC für verteilte Überwachung; Problem mit DNS -> Google DNS verwendet; Problem "apt-get update" hängt -> proxy2 verwenden und konfigurieren in /etc/environment	
05.12.16	49	15:00	17:15		2,25	2,25	Internetproblem Internat, Update und Upgrade, Installieren Icinga2	
16.12.16	50	7:50	13:10	4,42	0,91	5,33	Ubuntu auf Stand-PC installiert, Icinga2 auf Stand-PC installiert, mit SSH-Verbindung befasst um von Zuhause arbeiten zu können	
28.12.16	52	22:45	23:35		0,83	0,83	Update der Pakete, Installation Grafana auf Server (Port 3000 jetzt frei), Grafana nicht erreichbar	
01.01.17	52	16:00	17:30		1,50	1,50	verteilte Überwachung am Server; Master Konfiguration; Suche Grafana nicht erreichbar	
02.01.17	1	11:30	12:20		0,83	0,83	Installation Icinga; Problem: Repository	
09.01.17	2	13:00	14:20		1,33	1,33	Überprüfen der "sites-available", Grafana Port 3000 nicht erreichbar	
09.01.17	2	16:00	18:00		2,00	2,00	Komplettes Löschen Grafana und neu installieren (Fehler in Konfig vermutet)	
19.01.17	3	17:30	19:00		1,50	1,50	Präsentation	
20.01.17	3	7:50	13:10	4,42	0,91	5,33	Präsentation gehalten, Überarbeitung der Präsentation	
20.01.17	3	22:00	23:10		1,17	1,17	Template für Präsentationen, Bilder Aufgabenstellung/Projektteam	
21.01.17	3	14:00	15:20		1,33	1,33	PowerPoint für DA Präsentation	
23.01.17	4	14:15	16:00		1,75	1,75	Konfiguration Master und Windows Satellite	
23.01.17	4	18:00	19:30		1,50	1,50	Konfiguration Master und Windows Satellite	
24.01.17	4	12:20	16:30	4,17	0,00	4,17	PowerPoint Mujkanovic einbinden, Logos überarbeiten und in Folienmaster einfügen	
27.01.17	4	7:50	13:10	4,42	0,91	5,33	Optimierung Python Script, SNMP Theorie in PowerPoint einbinden	
27.01.17	4	13:30	14:30		1,00	1,00	Tag der offenen Tür Aufbau	
28.01.17	4	7:40	12:30		4,83	4,83	Diplomarbeit Präsentation Tag der offenen Tür	
31.01.17	5	15:45	17:35		1,83	1,83	Installation nach Server Reset	
01.02.17	5	14:00	16:30		2,50	2,50	Installation Influx & Grafana, Fehler bei Influx mit Paketen	
01.02.17	5	18:30	21:00		2,50	2,50	Installation easysnmp und net snmp	
02.02.17	5	18:30	21:15		2,75	2,75	Python Script nach Serverreset programmieren	



03.02.17	5	7:50	11:30	3,67	0,00	3,67	Snmp Installation Problem
03.02.17	5	11:30	13:10		1,67	1,67	Snmp Installation Problem
05.02.17	5	14:00	15:30		1,50	1,50	Linksammlung Doku
07.02.17	6	9:30	11:20		1,83	1,83	no module named "easysnmp" - Fehler
08.02.17	6	9:00	11:15		2,25	2,25	Dokumentation
10.02.17	6	13:30	15:30		2,00	2,00	Dokumentation
13.02.17	7	14:00	15:30		1,50	1,50	no module named "easysnmp" - Fehler behoben
14.02.17	7	17:00	19:45		2,75	2,75	snmpd error fixed, measurement fehler, retention policy database
15.02.17	7	12:35	14:10		1,58	1,58	Python script at startup
16.02.17	7	17:50	19:00		1,17	1,17	Grafana Dashboard
17.02.17	7	7:50	13:10	4,42	0,91	5,33	Besprechung, Doku
20.02.17	8	19:00	21:00		2,00	2,00	Dokumentation
28.02.17	9	18:00	21:00		3,00	3,00	Abstract
01.03.17	9	15:00	16:00		1,00	1,00	Besprechung Doku
01.03.17	9	16:15	18:15		2,00	2,00	Dokumentation
04.03.17	9	14:00	18:00		4,00	4,00	Abstract, Überwachung Linux PC Python
09.03.17	10	17:00	19:00		2,00	2,00	Blockschaltbild für Abstract, Formatierung und Übersetzung
10.03.17	10	7:50	13:10	4,42	0,91	5,33	Fertigstellen & Hochladen Abstract, RAM-Abfrage in Python Script hinzugefügt, Grafana Dashboard für RAM erstellt und konfiguriert
16.03.17	11	17:00	21:30		4,50	4,50	InfluxDB, Net-SNMP, EasySNMP Installation auf Satellite, Script zur Überwachung auf Satellite, Dokumentation
16.03.17	11	22:15	0:00		1,75	1,75	Dokumentation
17.03.17	11	7:50	13:10	4,42	0,91	5,33	SNMP Request timeout auf Satellite; SNMPD Daemon start failed auf Satellite; Dokumentation;
20.03.17	12	15:45	16:15		0,50	0,50	Dokumentation
21.03.17	12	19:00	21:15		2,25	2,25	SNMPD-Daemon error behoben (Neuinstallation), Dokumentation
23.03.17	12	9:00	10:30		1,50	1,50	Dokumentation
23.03.17	12	11:00	12:30		1,50	1,50	Dokumentation
23.03.17	12	13:00	13:55		0,92	0,92	Dokumentation
23.03.17	12	15:00	17:30		2,50	2,50	Dokumentation
23.03.17	12	19:40	21:30		1,83	1,83	Dokumentation
23.03.17	12	21:40	23:00		1,33	1,33	Dokumentation
24.03.17	12	8:30	12:00		3,50	3,50	Dokumentation
24.03.17	12	15:30	17:00		1,50	1,50	Dokumentation
24.03.17	12	20:15	22:00		1,75	1,75	Dokumentation
25.03.17	12	9:00	11:30		2,50	2,50	Dokumentation
25.03.17	12	20:20	22:30		2,17	2,17	Dokumentation
26.03.17	12	17:00	19:30		2,50	2,50	Dokumentation
27.03.17	13	17:00	19:00		2,00	2,00	Verbesserung Doku
27.03.17	13	20:00	21:40		1,67	1,67	Verbesserung Doku
29.03.17	13	15:00	16:45		1,75	1,75	Dokumentation



30.03.17	13	16:45	21:30		4,75	4,75	Doku zusammenfügen
31.03.17	13	7:40	13:10	4,42	1,08	5,50	Quellen aktualisieren, Doku
01.04.17	13	10:00	11:30			1,50	Doku
02.04.17	13	16:15	17:45			1,50	Doku
03.04.17	14	13:30	17:00		3,50	3,50	Doku fertigstellen

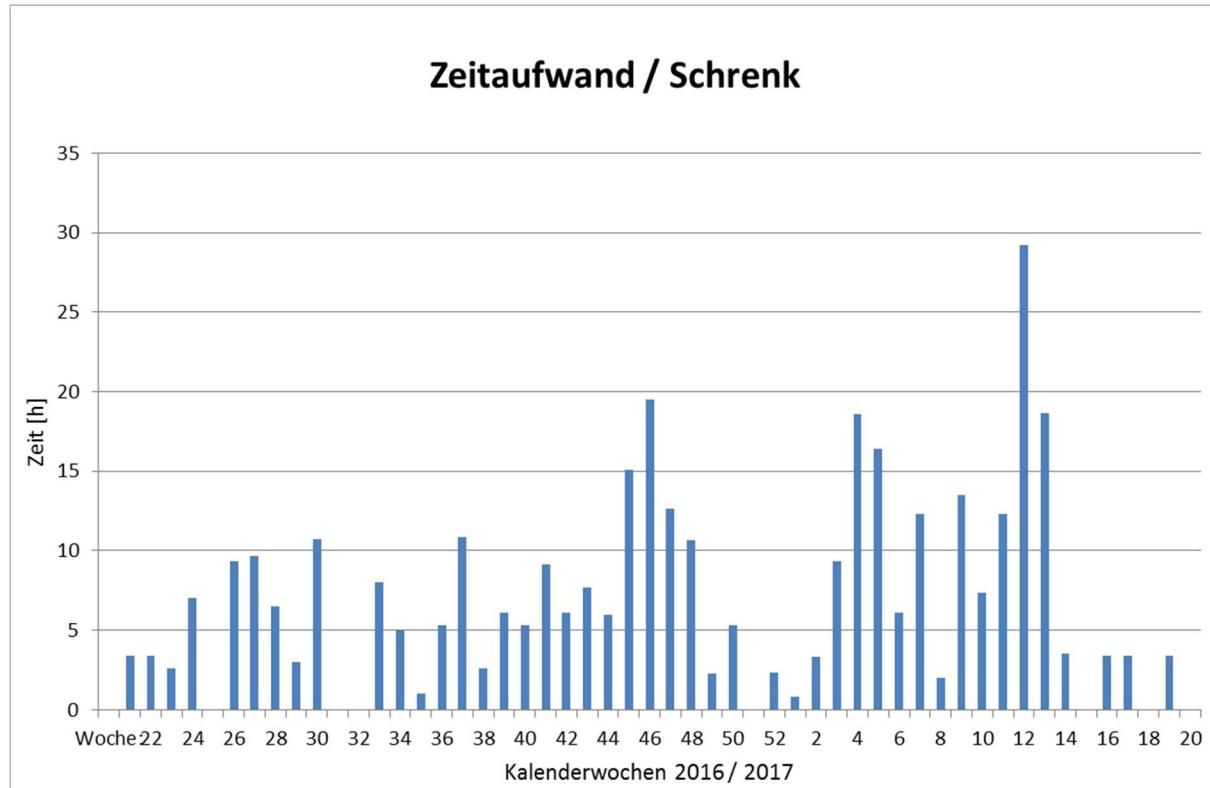


Abbildung 87 - Zeitaufwand Schrenk 7

Gesamt: 360,33h

Freizeit: 223,59h

Schule: 133,74h



7 Schlusswort

7.1 Icinga2

Schlussendlich kann nach den durchgearbeiteten Testszenarien gesagt werden, dass das Projekt „Netzwerküberwachung Windkraft Simonsfeld basierend auf Icinga2“ ein Erfolg ist. Das Programm ist sehr benutzerfreundlich und wirkt auch sehr modern im Gegensatz zum Vorgänger Icinga und zu anderen Monitoring Programmen. Das Einarbeiten erwies sich als sehr leicht da das Tool nicht allzu komplex ist und es für jede Funktion einen eigenen Punkt in der Dokumentation gibt. Darüber hinaus ist die eigene Konfigurationssprache sehr leicht zu erlernen und eine gute Idee seitens der Entwickler.

Icinga2 kann somit für jede Netzwerküberwachung empfohlen werden, vor allem für die der Windkraft Simonsfeld AG da es alle Anforderungen erfüllt.

Diese Dokumentation wird an die Windkraft Simonsfeld AG übergeben damit diese die Überwachung auf ihr Netzwerk eigenständig durchführen kann.

7.2 InfluxDB und Grafana

Abschließend kann gesagt werden, dass eine Netzwerküberwachung in einfacher Variante selbst zu programmieren durchaus in vernünftiger Zeit machbar ist. Der Vorteil der eigenen Programmierung ist der, dass eine Fehlersuche um einiges leichter ist, da es sich um einen selbst geschriebenen Code handelt.

Darüber hinaus kann man selbst festlegen, welche Funktionen man als wichtig empfindet und welche realisiert werden. Die Software ist also genau auf den Anwendungsfall abgestimmt und der Source-Code ist weniger kompliziert als bei fertigen Anwendungen.



8 Anhang

8.1 Icinga2 installieren

Icinga2 wurde mithilfe der Icinga2 Documentation installiert [DIC17].

Um Icinga2 zu installieren muss das Repository um das Icinga2 Repository erweitert werden:

```
# wget -O - http://packages.icinga.com/icinga.key | apt-key add -  
  
# echo 'deb http://packages.icinga.com/ubuntu icinga-xenial main' >/etc/apt/sources.list.d/icinga.list  
  
# apt-get update
```

Anschließend kann Icinga2 sowie die Nagios-Plugins installiert werden:

```
# apt-get install icinga2  
  
# apt-get install nagios-plugins
```

Um Icinga2 auf Funktion zu testen wird es gestartet:

```
/etc/init.d/icinga2  
  
{start|stop|restart|reload|checkconfig|status}
```

Mit der Kombination **/etc/init.d/icinga2 start** wird Icinga2 gestartet. Da das Tool ein Service ist kann es auch mit **service icinga2 start** gestartet werden.

Darauffolgend werden sowohl der mysql-server als auch der mysql-client installiert:

```
# apt-get install mysql-server mysql-client
```



Anschließend wird noch das icinga2-ido-mysql Modul installiert:

```
# apt-get install icinga2-ido-mysql
```

Nun kann für Icinga2 in MySQL eine eigene Datenbank angelegt werden:

```
# mysql -u root -p

mysql> CREATE DATABASE icinga;

mysql> GRANT SELECT, INSERT, UPDATE, DELETE, DROP, CREATE VIEW, INDEX, EXECUTE ON icinga.* TO 'icinga'@'localhost' IDENTIFIED BY 'icinga';

mysql> quit
```

Nun ist es wichtig das Icinga2 IDO Schema auf die erstellte Datenbank zu übertragen:

```
# mysql -u root -p icinga < /usr/share/icinga2-ido-mysql/schema/mysql.sql
```

Das IDO Schema ist ein eigenes Feature von Icinga2. Um dieses verwenden zu können muss es anschließend noch aktiviert werden:

```
# icinga2 feature enable ido-mysql
```

Wichtig: Icinga2 muss immer neu gestartet werden, wenn Änderungen durchgeführt werden um die Änderungen zu übernehmen und – falls vorhanden – mithilfe einer Fehlermeldung einen Fehler beheben.



8.2 Icingaweb2 installieren

Icingaweb2 wurde mithilfe der Website von Thomas Krenn installiert [TKC17].

Die Installation von Icinga2 ist somit abgeschlossen. Es können schon Services, Hosts usw. angelegt werden, jedoch wird eine GUI benötigt um all diese Daten auch zu sehen.

Icingaweb2 kann auf einem Webserver ausgeführt werden. Daher wird als Erstes Apache2 benötigt:

```
# apt-get install apache2
```

Icinga2 ist in der Lage Kommandos zur Weboberfläche über die Command Pipe zu schicken.

Diese muss aktiviert werden:

```
# icinga2 feature enable command
```

Anschließend wird Icingaweb2 installiert:

```
tk@icinga2:~$ sudo apt-get install icingaweb2
```

Anmerkung: Ab Ubuntu 16.04 muss das PHP7-Modul für Apache2 installiert werden.

```
apt-get -y install libapache2-mod-php7.0
```

Nun wird die MySQL Datenbank konfiguriert:

```
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, DROP, CREATE VIEW, INDEX, EXECUTE ON icinga2.* TO 'icingaweb'@'localhost' IDENTIFIED BY '<PASSWORD>';

Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;

Query OK, 0 rows affected (0.00 sec)

mysql> quit
```



Anschließend werden alle Dienste neu gestartet:

```
tk@ubuntuserver:~$ sudo service mysql restart && sudo service icinga2
restart
```

Danach kann das sogenannte Token, welcher für die Icingaweb2 Installation benötigt wird, angelegt werden:

```
$ sudo icingacli setup config directory --group icingaweb2
$ sudo icingacli setup token create
```

Zum Schluss kann Icingaweb2 über <http://IP/icingaweb2> konfiguriert werden. Für diese Konfiguration gibt es detaillierte Screenshots auf der Seite von Thomas Krenn.

8.3 Graphite installieren

Graphite wurde mithilfe einer Anleitung von linuxfrickeln.de installiert [LFD17].

Um Graphite zu installieren muss zunächst das Modul `perfdata` und das Modul `graphite` aktiviert werden:

```
icinga2 feature enable perfdata graphite
```

Danach sollte icinga2 neu gestartet werden.

Nun wird das Graphite Modul unter Icingaweb2 angelegt. Hierfür wird ein Projekt von Github zu den Modulen kopiert:

```
cd /usr/share/icingaweb2/modules/
git clone https://github.com/findmypast/icingaweb2-module-graphite.git
```



Anschließend muss der Verzeichnisname angepasst werden, da die GUI sonst die Daten des Moduls nicht findet:

```
mv icingaweb2-module-graphite graphite
```

Als Nächstes muss eine lokale Konfiguration für das Graphite-Modul der GUI erstellt werden. An diesem Socket werden die Performance Daten abgeholt und in der GUI entsprechend dargestellt:

```
Nano /etc/icingaweb2/modules/graphite/config.ini
```

```
[graphite]

metric_prefix = icinga

base_url = http://localhost/render?
```

Graphite benötigt zur Auswertung der Daten den Carbon-Cache, welcher standardmäßig zum Empfang von Daten auf Port 2003 an localhost lauscht. Dies ist richtig voreingestellt, daher kann die graphite.conf so belassen werden, wie sie ist.

Nun wird das erwähnte Carbon installiert:

```
apt-get install graphite-carbon
```

Damit die Aggregation der Daten in Carbon funktioniert müssen in der Datei storage-schemas.conf die Intervalle eingetragen werden, nach denen Carbon arbeiten soll.



Die Icinga2-Docs geben diese eigentlich schon vor, trotzdem sollte dies überprüft werden:

```
Nano /etc/carbon/storage-schemas.conf
```

```
[icinga_internals]

pattern = ^icinga\..*\.(max_check_attempts|reachable|current_attempt|execution_time|latency|state|state_type)

retentions = 5m:7d


[icinga_default]

pattern = ^icinga\.

retentions = 1m:2d,5m:10d,30m:90d,360m:4y
```

Anschließend wird der Carbon-Cache neu gestartet:

```
systemctl restart carbon-cache.service
```

Danach sollte Icinga2 und Apache2 neu gestartet werden um die vorherigen Schritte auf einen Konflikt mit den Programmen zu prüfen.

```
systemctl restart icinga2

systemctl restart apache2
```

Nun kann graphite-web installiert werden:

```
apt-get install graphite-web
```

Für die Oberfläche muss eine kleine Konfiguration vorgenommen werden.



```
Nano /etc/graphite/local_settings.py
```

Unter "SECRET_KEY" wird ein selbst gewählter Key eingegeben:

```
#SECRET_KEY = 'Hmwjnfoasndo24234djfsdlds'
```

Abbildung 88 – Secret_Key Graphite

Achtung: Wird dies nicht getan erscheint ständig eine WARNING beim Neustart der Carbon-Cache!

Danach müssen – der Datenbank entsprechend – folgende Einstellungen übernommen werden:

```
DATABASES = {
    'default': {
        'NAME': 'graphite',
        'ENGINE': 'django.db.backends.mysql',
        'USER': 'graphite',
        'PASSWORD': 'Oth1Muri1',
        'HOST': '127.0.0.1',
        'PORT': '3306'
    }
}
```

Abbildung 89 – Datenbankeinstellung Graphite

Anschließend wird die Graphite Seite für den Webserver aktiviert und das Python-WSGI-Interface installiert:

```
a2ensite apache2-graphite.conf
apt-get install libapache2-mod-wsgi
```

Zu guter Letzt wird Python synchronisiert:

```
python /usr/lib/python2.7/dist-packages/graphite/manage.py syncdb
```



Jetzt kann unter <https://IP> die Graphite-Web Oberfläche betrachtet werden. Sollen die Graphen auch in Icingaweb2 betrachtet werden können muss unter „Modules“ das Graphite Modul aktiviert werden – nach einem Neustart sind alle Graphen beim entsprechenden Service vorhanden.

8.4 Verteiltes Netzwerk aufbauen

Der Aufbau eines Verteilten Netzwerkes in Icinga2 wurde mithilfe einer Anleitung aus der Icinga2 Documentation durchgeführt [DIC17b].

8.4.1 Allgemeines

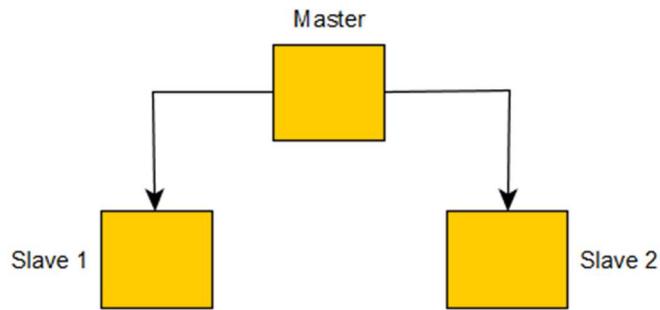


Abbildung 90 - Verteiltes Netzwerk: Hiararchie

Die verteilte Überwachung von Icinga2 basiert auf einem **Master-Slave Prinzip**. Der Master kann seinen Slaves Aufgaben zuteilen (**Top Down Prinzip**) oder die Slaves führen ihre Arbeit ständig durch und schicken die Ergebnisse zum Master (**Bottom Up Prinzip**).

In unserem Fall wird das Bottom Up Prinzip benutzt da ein Slave seine Checks auch dann durchführen soll, wenn die Verbindung zum Master getrennt wird. Anschließend soll nach einer wiederhergestellten Verbindung die Ergebnisse der Checks an den Master übermittelt werden.



Jeder Slave kann weitere Slaves besitzen, diese werden dann Clients genannt.

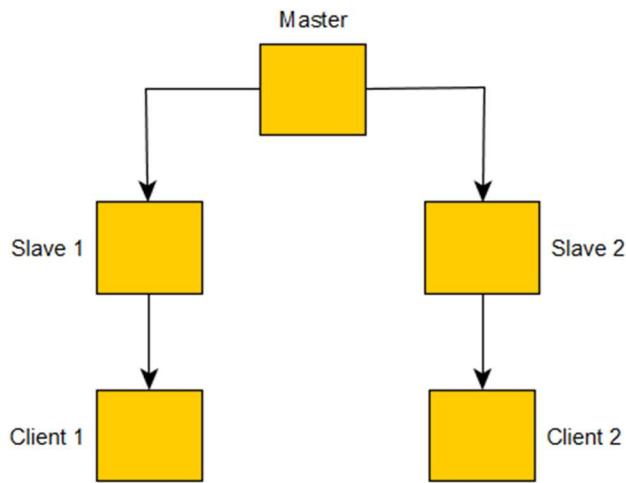


Abbildung 91 - Verteiltes Netzwerk: Erweiterte Hierarchie

8.4.2 Master Setup

Als erstes muss auf dem Master Icinga2 sowie Icingaweb2 installiert werden.

Anschließend muss in der Datei **zones.conf** unter **/etc/icinga2** die Definition von Master und Slaves vorgenommen werden:

```
object Endpoint "icinga2-master1.localdomain" {
    host = "192.168.56.101"
}

object Endpoint "icinga2-satellite1.localdomain" {
    host = "192.168.56.105"
}

object Zone "master" {
    endpoints = [ "icinga2-master1.localdomain" ]
}
```



```
object Zone "satellite" {  
  
    endpoints = [ "icinga2-satellite1.localdomain" ]  
  
    parent = "master"  
  
}
```

Nun sind der Master und ein Slave mit **ihrer IP-Adresse und einem Namen definiert** (IP-Adressen sowie Namen können frei gewählt werden, müssen jedoch gemerkt werden → Tipp: leichte Namen wählen wie z.B.: Windows, Ubuntu, Slave 1, Slave 37 usw.).

Als nächstes muss der API-Listener aktiviert werden:

```
icinga2 feature enable api
```

Bei einem Neustart von Icinga2 kommt es nun zu einem Fehler welcher ignoriert werden kann da er im nächsten Schritt behoben wird.

Nun folgt die Master Konfiguration. Hierfür wird der Befehl **icinga2 node wizard** ausgeführt (der Icinga2 Node Wizard übernimmt die manuelle Konfiguration und führt mit Abfragen diese selber durch, siehe Beispiel).

```
[root@icinga2-master1.localdomain /]# icinga2 node wizard  
  
Welcome to the Icinga 2 Setup Wizard!  
  
We'll guide you through all required configuration details.  
  
Please specify if this is a satellite setup ('n' installs a master setup)  
[Y/n]: n  
  
Starting the Master setup routine...  
  
Please specifiy the common name (CN) [icinga2-master1.localdomain]: icinga  
2-master1.localdomain
```



```
Checking for existing certificates for common name 'icinga2-master1.localdomain'...
```

```
Certificates not yet generated. Running 'api setup' now.
```

```
information/cli: Generating new CA.
```

```
information/base: Writing private key to '/var/lib/icinga2/ca/ca.key'.
```

```
information/base: Writing X509 certificate to '/var/lib/icinga2/ca/ca.crt'
```

```
.
```

```
information/cli: Generating new CSR in '/etc/icinga2/pki/icinga2-master1.localdomain.csr'.
```

```
information/base: Writing private key to '/etc/icinga2/pki/icinga2-master1.localdomain.key'.
```

```
information/base: Writing certificate signing request to '/etc/icinga2/pki/icinga2-master1.localdomain.csr'.
```

```
information/cli: Signing CSR with CA and writing certificate to '/etc/icinga2/pki/icinga2-master1.localdomain.crt'.
```

```
information/cli: Copying CA certificate to '/etc/icinga2/pki/ca.crt'.
```

```
Generating master configuration for Icinga 2.
```

```
information/cli: Adding new ApiUser 'root' in '/etc/icinga2/conf.d/api-users.conf'.
```

```
information/cli: Enabling the 'api' feature.
```

```
Enabling feature api. Make sure to restart Icinga 2 for these changes to take effect.
```

```
information/cli: Dumping config items to file '/etc/icinga2/zones.conf'.
```

```
information/cli: Created backup file '/etc/icinga2/zones.conf.orig'.
```

```
Please specify the API bind host/port (optional):
```

```
Bind Host []:
```

```
Bind Port []:
```



```
information/cli: Created backup file '/etc/icinga2/features-available/api.conf.orig'.
```

```
information/cli: Updating constants.conf.
```

```
information/cli: Created backup file '/etc/icinga2/constants.conf.orig'.
```

```
information/cli: Updating constants file '/etc/icinga2/constants.conf'.
```

```
information/cli: Updating constants file '/etc/icinga2/constants.conf'.
```

```
information/cli: Updating constants file '/etc/icinga2/constants.conf'.
```

Done.

Now restart your Icinga 2 daemon to finish the installation!

Wenn jetzt Icinga2 neu gestartet wird ist die Fehlermeldung von vorher nicht mehr vorhanden.

Hiermit ist der Host als ein Master definiert worden.

8.4.3 Satellite Setup

Die Satellites werden mithilfe von Tickets angelegt. Diese sollen helfen die Satellites unterscheidbar zu machen.

Zunächst wird ein Ticket auf dem Master für den Client generiert:

```
[root@icinga2-master1.localdomain /]# icinga2 pki ticket --cn icinga2-client1.localdomain
```

Anschließend wird ein neuer API-User angelegt:

```
[root@icinga2-master1.localdomain /]# nano /etc/icinga2/conf.d/api-users.conf
```

```
object ApiUser "client-pki-ticket" {
    password = "bea11beb7b810ea9ce6ea" //change this
    permissions = [ "actions/generate-ticket" ]
}
```



Anmerkung: Den bestehenden API-User nicht löschen! Für jeden Satellite muss ein neuer API-User angelegt werden!

Nun wird Icinga2 neu gestartet und ein Befehl ausgeführt:

```
[root@icinga2-master1.localdomain /]# curl -k -s -u client-pki-ticket:bea11beb7b810ea9ce6ea -H 'Accept: application/json' \  
-X POST 'https://icinga2-master1.localdomain:5665/v1/actions/generate-  
ticket' -d '{ "cn": "icinga2-client1.localdomain" }'
```

8.4.3.1 Windows Satellite

Der Windows Satellite wird mit einem Programm installiert:

<http://packages.icinga.org/windows/>. Dieses Programm wird installiert und ausgeführt.

Anschließend öffnet sich eine grafische Oberfläche.

Unter **Instance Name** wird der Name des Satellite eingegeben und unter **Setup Ticket** die Ticket Nummer welche am Master generiert wurde. Danach wird unter **Add** zuerst der **Instance Name**, danach die **IP-Adresse des Masters** und der **Port 5665** eingegeben. Unter Advanced Options noch die 3 Häkchen setzen und auf **Next** klicken bis die Meldung **The Icinga2 Client was set up successfully** erscheint.

In der Eingabeaufforderung (welche mit Administrator Rechten ausgeführt wird) wird noch mit **net stop icinga2** und **net start icinga2** das Programm neu gestartet.



8.4.3.2 Ubuntu Satellite

Zuerst lassen wir uns das Ticket unseres Satellites erneut zeigen:

```
[root@icinga2-master1.localdomain /]# icinga2 pki ticket --cn icinga2-client1.localdomain  
4f75d2ecd253575fe9180938ebff7cbca262f96e
```

Nun wird der Befehl **icinga2 node wizard** ausgeführt:

```
[root@icinga2-client1.localdomain /]# icinga2 node wizard  
Welcome to the Icinga 2 Setup Wizard!  
  
We'll guide you through all required configuration details.  
  
Please specify if this is a satellite setup ('n' installs a master setup)  
[Y/n] :  
  
Starting the Node setup routine...  
  
Please specify the common name (CN) [icinga2-client1.localdomain]:  
icinga2-client1.localdomain  
  
Please specify the master endpoint(s) this node should connect to:  
  
Master Common Name (CN from your master setup): icinga2-master1.localdomain  
  
Do you want to establish a connection to the master from this node? [Y/n] :  
  
Please fill out the master connection information:  
  
Master endpoint host (Your master's IP address or FQDN): 192.168.56.101  
  
Master endpoint port [5665]:  
  
Add more master endpoints? [y/N] :  
  
Please specify the master connection for CSR auto-signing (defaults to  
master endpoint host):  
  
Host [192.168.56.101]: 192.168.2.101  
  
Port [5665]:  
  
information/base: Writing private key to '/etc/icinga2/pki/icinga2-client1.localdomain.key'.  
  
information/base: Writing X509 certificate to '/etc/icinga2/pki/icinga2-client1.localdomain.crt'.
```



```
information/cli: Fetching public certificate from master (192.168.56.101, 5665):
```

```
Certificate information:
```

```
Subject: CN = icinga2-master1.localdomain
Issuer: CN = Icinga CA
Valid From: Feb 23 14:45:32 2016 GMT
Valid Until: Feb 19 14:45:32 2031 GMT
Fingerprint: AC 99 8B 2B 3D B0 01 00 E5 21 FA 05 2E EC D5 A9 EF 9E AA E3
```

```
Is this information correct? [y/N]: y
```

```
information/cli: Received trusted master certificate.
```

```
Please specify the request ticket generated on your Icinga 2 master.
```

```
(Hint: # icinga2 pki ticket --cn 'icinga2-client1.localdomain'): 4f75d2ecd253575fe9180938ebff7cbca262f96e
```

```
information/cli: Requesting certificate with ticket '4f75d2ecd253575fe9180938ebff7cbca262f96e'.
```

```
information/cli: Created backup file '/etc/icinga2/pki/icinga2-client1.localdomain.crt.orig'.
```

```
information/cli: Writing signed certificate to file '/etc/icinga2/pki/icinga2-client1.localdomain.crt'.
```

```
information/cli: Writing CA certificate to file '/etc/icinga2/pki/ca.crt'.
```

```
Please specify the API bind host/port (optional):
```

```
Bind Host []:
```

```
Bind Port []:
```

```
Accept config from master? [y/N]: y
```

```
Accept commands from master? [y/N]: y
```

```
information/cli: Disabling the Notification feature.
```

```
Disabling feature notification. Make sure to restart Icinga 2 for these changes to take effect.
```



```
information/cli: Enabling the Apilistener feature.  
information/cli: Generating local zones.conf.  
information/cli: Dumping config items to file '/etc/icinga2/zones.conf'.  
information/cli: Updating constants.conf.  
information/cli: Updating constants file '/etc/icinga2/constants.conf'.  
information/cli: Updating constants file '/etc/icinga2/constants.conf'.  
Done.
```

Now restart your Icinga 2 daemon to finish the installation!

8.4.3.3 Einfügen der Satellites in Icingaweb2

Die Satellite-PCs sind nun im Icinga2 vermerkt aber noch nicht eingebunden. Icinga weiß bisher nur, dass die Satellites existieren, nicht dass es diese überwachen soll.

Um die Satellite-PCs zu überwachen müssen folgende Befehle ausgeführt werden:

```
[root@icinga2-master1.localdomain /]# icinga2 node list
```

Hier werden anschließend alle Satellites angezeigt welche eingebunden worden sind.

```
[root@icinga2-master1.localdomain /]# icinga2 node update-config
```

Mit diesem Befehl werden die Satellites in Icinga2 & Icingaweb2 eingebunden und sollten (wenn alles richtiggemacht wurde) im Icingaweb2 zu sehen sein. Die Satellites sind zuerst im Pending Modus (so nennt man den Zustand wenn ein neuer Host eingebunden wird, dauert 3 Checks → kann beschleunigt werden durch drücken auf den Host und 3 mal den Check manuell ausführen).



8.5 Installation Net-SNMP

8.5.1 Allgemeines

Net-SNMP wird benötigt um generell eine Überwachung zu ermöglichen, da diese bei dem von uns verwendeten Betriebssystem nicht vorinstalliert ist.

8.5.2 Anleitung

Zu Beginn ist es nötig das zu verwendende Paket herunterzuladen, dieses wird zum Beispiel auf der offiziellen Seite <http://www.net-snmp.org/download.html> heruntergeladen werden. Dies ist aber nur dann möglich, wenn eine grafische Oberfläche vorhanden ist. Da diese aber in unserem Fall nicht zur Verfügung steht ist es nötig das Paket über das Terminal mit folgendem Befehl herunterzuladen:

[SCF17]

```
wget http://sourceforge.net/projects/net-snmp/files/net-snmp/5.7.3/net-snmp-5.7.3.tar.gz
```

Nach der Eingabe dieses Befehls wird der Download durchgeführt und bei Erfolg am Ende mit folgender Ausgabe am Bildschirm bestätigt:

```
2017-03-17 12:40:14 (1,69 MB/s) - 'net-snmp-5.7.3.tar.gz.1' saved [6382428/6382428]
```

Abbildung 92 – Download Net-SNMP erfolgreich

Um dieses Paket verwenden zu können ist es nötig mit folgendem Befehl das „Perl development package“ zu installieren:

```
sudo apt-get install libperl-dev
```

Hierbei dient das „sudo“ dazu um den Befehl als Administrator auszuführen, was die erneute Eingabe des root-Passworts erfordert!



Nun ist es erforderlich das eben heruntergeladene Paket zu entpacken, um es anschließend installieren zu können. Dazu verwendet man den „tar“-Befehl wie folgt:

```
tar -xvzf net-snmp-5.7.3.tar.gz
```

Zu achten ist auf die Richtigkeit der Version, welche sich im Dateinamen befindet und in diesem Fall „5.7.3“ ist. Diese Versionsnummer muss mit dem heruntergeladenen Paket übereinstimmen.

Nachdem das Paket entpackt wurde ist es nötig in den entpackten Ordner „net-snmp-5.7.3“ zu wechseln.

Dazu wird folgender Befehl verwendet:

```
cd net-snmp-5.7.3/
```

„cd“ gibt in Linux immer einen Wechsel des Verzeichnisses an.

In diesem Ordner kann dann das Konfigurationsfile gestartet werden. Hierzu ist es nötig mit Administratorrechten das File aufzurufen. Dies funktioniert mit folgendem Befehl:

```
sudo ./configure
```

„./“ gibt an, dass das nachfolgende File gestartet werden soll. In diesem Fall handelt es sich um die Datei mit dem Namen „configure“.

Die Abarbeitung dieser Datei erfordert einige Benutzereingaben, welche aber einfach mit der Enter-Taste übersprungen werden, bis eine Übersicht der Konfiguration auf dem Bildschirm zu sehen ist und die Konsole auf weitere Eingaben wartet.



Die nächste Eingabe ist wieder als Administrator auszuführen. Die Verbindung mit „&&“ gibt an, dass zuerst der linke Befehl und nachdem dieser abgeschlossen ist sofort der rechte Befehl abzuarbeiten ist. Dies erleichtert die Eingabe, da die Abarbeitung des nächsten Befehls von selbst erledigt wird.

```
sudo make && sudo make install
```

Nun ist es nötig die MIB-Files auf dem System zu installieren. Hierzu wird folgender Befehl ausgeführt:

```
sudo apt-get install snmp-mibs-downloader
```

Als nächste Operation wird noch der SNMPD-Daemon installiert, welcher nötig ist um Funktionen der Netzwerk-Überwachung durchführen zu können. Dies kann über folgende Eingabe ausgeführt werden:

```
sudo apt-get install snmpd
```

8.6 Installation Easy-SNMP

8.6.1 Allgemeines

Um eine SNMP-Abfrage mittels Python zu ermöglichen ist es nötig eine geeignete Library zu installieren. Diese Library dient dazu die SNMP-Befehle in der verwendeten Programmiersprache verwenden zu können, da diese in diesem Fall nicht standardmäßig verwendbar sind. Hier wurde EasySNMP verwendet. [EAS17b]



8.6.2 Anleitung

Um EasySNMP korrekt installieren und verwenden zu können ist es nötig zuerst Net-SNMP wie oben beschrieben zu installieren. Hierbei ist es wichtig, dass eine Version 5.7.X von Net-SNMP installiert ist, da EasySNMP auf Systemen läuft, auf denen auch diese Versionen laufen.

Ist dies sichergestellt und die Installation abgeschlossen kann mit dieser Installation fortgefahren werden.

Zuerst wird der gcc-Compiler in Verbindung mit Python benötigt, welcher folgendermaßen installiert wird:

```
sudo apt-get install gcc python-dev
```

8.6.3 Verwendung

In diesem Fall wird die Verwendung des Python-Moduls beschrieben, da die Library zu diesem Zweck installiert wurde.

Um die mitgelieferten Befehle verwenden zu können ist es zu Beginn des Scripts nötig das Modul zu importieren. Dies funktioniert mit folgender Zeile:

```
from easysnmp import snmp_get
```

In diesem Fall wird nur der Befehl um einzelne Abfragen zu machen benötigt, weshalb aus dem Modul lediglich der Befehl „snmp_get“ importiert wird.

Um nun die SNMP-Daten eines Geräts zu erhalten ist es nötig sowohl die IP-Adresse als auch den Community-Namen zu kennen.

```
snmp_get('OID', hostname='IP-Adresse', community='Community-Name',  
version=1)
```



Der folgende Code zeigt beispielhaft die Verwendung des obigen Befehls für den Temperatursensor:

```
temp = snmp_get('1.3.6.1.4.1.21796.4.1.3.1.4.1',  
hostname='10.48.1.201', community='public', version=1)
```

Dieser Befehl liefert sämtliche SNMP-Werte des Temperatursensors zurück, da aber in diesem Fall nur der aktuelle Temperaturwert benötigt wird muss auf das richtige Objekt zugegriffen werden:

```
val = temp.value
```

Der Punkt signalisiert den Zugriff auf das Objekt „value“, in welchem der vom Sensor aufgenommene Temperaturwert gespeichert ist.

8.7 Installation InfluxDB

8.7.1 Allgemeines

InfluxDB liefert standardmäßig mit der Installation das benötigte Python Modul zur Verwendung mit. In diesem Fall wird die Installation der Datenbank auf Ubuntu 16.04 LTS erklärt. Bei anderen Systemen, wie zum Beispiel Debian, CentOS, openSUSE und so weiter werden andere Befehle benötigt. Diese können unter docs.influxdata.com unter „Installation“ gefunden werden. [DIN17b]

Bei der folgenden Anleitung ist es nötig sich mit `sudo -i` und dem zugehörigen Passwort des root-Users anzumelden oder vor sämtliche Befehle, die diese Rechte benötigen, ein `sudo` zu setzen, da dieser Teil des Befehls hier weggelassen wurde, da ohnehin mit root-Rechten gearbeitet wurde.



8.7.2 Anleitung

Zu Beginn ist es nötig das Repository von InfluxData zur Paketliste hinzuzufügen, dies wird mit folgenden Befehlen erledigt:

```
curl -sL https://repos.influxdata.com/influxdb.key | sudo apt-key add -  
source /etc/lsb-release  
echo "deb https://repos.influxdata.com/${DISTRIB_ID,,} ${DISTRIB_CODENAME}  
stable" | sudo tee /etc/apt/sources.list.d/influxdb.list
```

Wurde die Paketquelle erfolgreich zur Paketliste hinzugefügt ist es nötig ein Update der Liste durchzuführen, was mit folgendem Befehl erledigt wird:

```
apt update
```

Alternativ kann auch der bisher geläufigere Befehl `apt-get update` verwendet werden, da die Kurzform mit `apt` lediglich ein Ersatz für diesen Befehl ist.

Ist die Paketliste nun aktuell kann mit der eigentlichen Installation der Datenbank fortgesetzt werden. Hierzu wird wieder entweder der Befehl `apt-get install influxdb` oder der Befehl `apt install influxdb` verwendet.

Wurde die Datenbank erfolgreich installiert kann der Service mit folgendem Befehl gestartet werden:

```
service influxdb start
```

Wenn das System „systemd“ verwendet wird der Service mit `systemctl start influxdb` gestartet.



Wurde der Service gestartet so kann mit der Konfiguration begonnen werden.

Diese kann entweder mittels `influx` oder mittels `influx -config /etc/influxdb/influxdb.conf` durchgeführt werden. Hierbei wird aber eher die zweite Variante bevorzugt, da direkt beim Befehl mit dem Parameter „-config“ die zu verwendende Konfigurationsdatei angegeben wird. Dies ist vor allem wichtig, wenn mehrere Konfigurationsdateien vorhanden sind.

8.7.3 Verwendung

Um die Datenbank generell zu verwenden wird zuerst der Befehl `influx` ausgeführt.

Danach ergibt sich folgendes Bild:

```
root@icinga:~# influx
Connected to http://localhost:8086 version 1.2.1
InfluxDB shell version: 1.2.1
> 
```

Abbildung 93 – InfluxDB Start

Man erhält Informationen über den Host des Datenbankmanagementsystems, welcher in diesem Fall der Localhost auf Port 8086 ist, und über die aktuell installierte Version, welche in diesem Fall 1.2.1 ist.

Nach dieser Ausgabe erkennt man an dem Pfeil nach rechts „>“, dass die Konsole nun auf die Eingabe des Benutzers wartet.

Da es sich um eine SQL-ähnliche Sprache handelt können sehr viele der klassischen SQL-Befehle verwendet werden.

Zum Erstellen einer Datenbank kann simpel `create database name` verwendet werden, wie er aus der Sprache SQL bereits bekannt ist.



Weiters können die existierenden Datenbanken zum Beispiel mittels `show databases` angezeigt werden, was auch mit den sogenannten Measurements funktioniert.

Ein Measurement ist in InfluxDB vergleichbar wie ein Table bei SQL-Datenbanken. In diesen Measurements werden sämtliche Werte abgespeichert.

```
> show databases
name: databases
name
-----
internal
diplomarbeit
test
```

Abbildung 94 – Anzeige Datenbanken

Hier sieht man die Ausgabe des obigen Befehls. Es werden sämtliche vorhandenen Datenbanken angezeigt. In unserem Fall sind das die bereits nach der Installation vorhandene namens „internal“ sowie einmal die Datenbank, in der sämtliche Werte der Überwachung gespeichert werden namens „diplomarbeit“ und eine, die für Testzwecke verwendet wird, um keine Fehler an der Haupt-Datenbank zu machen namens „test“.

Der Befehl `use dbname` kann ebenfalls von SQL übernommen werden.

Das Einfügen von Werten sieht bei InfluxDB folgendermaßen aus:

```
insert measurement,tag=name,tag=name key=value
```



Der obige Befehl kann nun an folgendem Beispiel für das händische Einfügen von Daten erklärt werden:

```
> show measurements
name: measurements
name
-----
RAM_satellite
temp
testmeasure

> insert test,host=Sensorname,region=standort value=25
> select * from test
name: test
time                  host      region  value
-----                ----      -----  -----
1490227227222348105 Sensorname  standort 25
```

Abbildung 95 – Anzeige Measurements

Zuerst werden alle Measurements, welche bei InfluxDB mit den Tables von SQL-Datenbanken verglichen werden können, aufgelistet und man sieht, dass kein Measurement namens „test“ vorhanden ist. Wird nun der insert-Befehl für ein Measurement ausgeführt, welches nicht existiert, so wird dieses angelegt.

„host“ ist in diesem Fall der erste Tag und „region“ ist der zweite Tag. Diese Tags sind indiziert und können sowohl Buchstaben als auch Zahlen enthalten, sind aber nicht unbedingt benötigt. Tags werden immer mit einem Beistrich voneinander getrennt und dem Tag wird ein Wert über ein „=“ zugewiesen.

Value ist in diesem Fall ein Field, welche nicht indiziert sind und vorhanden sein müssen.

Weiters sieht man, dass diese Datenbank standardmäßig einen Zeitstempel hinzufügt, welcher die Anzeige in Kurven stark erleichtert.

Mit dem Befehl `drop` können zum Beispiel Datenbanken, Measurements oder einzelne Einträge in einem Measurement gelöscht werden.



Um einen Eintrag zu löschen wird folgender Befehl verwendet:

```
drop series from test where region='standort'
```

Damit werden alle Einträge aus dem Measurement „test“ gelöscht, wo bei Region „standort“ angegeben wurde.

```
> select * from test
name: test
time          4 hallo host      region  value
----          - ----- ----      -----  -----
1490227227222348105      Sensorname standort 25
1490227619714309960      Sensorname 7      25
1490227635794309378  4      Sensorname 7      25
1490227647241970039 3 5  Sensorname 7      25
1490227686454324011 3 3  Sensorname 7      25

> drop series from test where region='standort'
> select * from test
name: test
time          4 hallo host      region  value
----          - ----- ----      -----  -----
1490227619714309960      Sensorname 7      25
1490227635794309378  4      Sensorname 7      25
1490227647241970039 3 5  Sensorname 7      25
1490227686454324011 3 3  Sensorname 7      25
```

Abbildung 96 – Einträge löschen

Man sieht, dass nach diesem Befehl genau die eine Zeile, in der zuvor die Region als „standort“ definiert war, gelöscht wurde.



Es ist ebenso möglich Daten anhand ihres Zeitstempels zu löschen, dabei wird aber anstatt `drop series` der Befehl `delete` verwendet. In diesem Fall werden alle Daten gelöscht, deren Stempel kleiner als die angegebene Zahl ist:

```
> select * from test
name: test
time          host  region value
----          ----  -----  -----
1490229554162267479 name1 name2  7
1490229558061327506 name1 name2  8
1490229560773342392 name1 name2  9
1490229564421517500 name1 name2 10
1490229570073598144 name1 name2 11
1490229572853289313 name1 name2 12
1490229574773361255 name1 name2 13

> delete from test where time < 1490229560773342392
> select * from test
name: test
time          host  region value
----          ----  -----  -----
1490229560773342392 name1 name2  9
1490229564421517500 name1 name2 10
1490229570073598144 name1 name2 11
1490229572853289313 name1 name2 12
1490229574773361255 name1 name2 13
```

Abbildung 97 – löschen nach Zeit

In diesem Beispiel wurde genau der Zeitstempel des Eintrags mit einem „value“ von 9 gewählt. Man sieht also, dass die beiden Einträge, die älter als dieser sind, gelöscht werden.

Um nun InfluxDB in Python verwenden zu können ist es zuerst nötig entweder mittels `pip install influxdb` oder bei Debian/Ubuntu mit `apt install python-influxdb` die Datenbank für Python zu installieren. Danach muss das richtige Modul importiert werden. Das Importieren des gesamten Moduls funktioniert mittels `import influxdb`. In diesem Fall wird aber nur der Client benötigt, weshalb mittels `from influxdb import InfluxDBClient` dieser importiert wird.



Um nun das Programm mit der Datenbank zu verbinden wird folgender Befehl verwendet:

```
#Accessing database
client = InfluxDBClient('localhost', 8086, 'root', 'root', 'test')
```

Abbildung 98 – Zugriff auf Datenbank

Localhost wird in diesem Fall verwendet, weil die Datenbank auf dem lokalen System installiert wurde.

Der Befehl ist folgendermaßen aufgebaut:

```
client = InfluxDBClient('host', port, 'user', 'password', 'dbname')
```

In diesem Fall wird also zuerst in das Datenbankmanagementsystem mittels Benutzernamen und Passwort „root“ eingeloggt und dann die Datenbank „test“ verwendet.

Das Schreiben der Werte in die Datenbank funktioniert über eine Struktur, welche in diesem Fall als JSON-Body definiert wurde. Diese Definition wird später genauer erläutert.

Der verwendete Befehl `client.write_points(json_body)` greift auf das Attribut „write_points“ der Instanz „client“ zu und übergibt als Parameter den Namen der Struktur.

Bei „json_body“ ist der im Programmcode definierte Name der Struktur anzugeben.

```
json_body = []
{}
```

Abbildung 99 – Name JSON Body

Dieser lautet in unserem Fall „json_body“.



8.8 Installation Grafana

8.8.1 Allgemeines

Diese Installation wurde als root-User durchgeführt um sämtliche Befehle ohne „sudo“ ausführen zu können. Um in diesen permanenten root-Modus zu kommen ist es nötig den Befehl `sudo -i` auszuführen und mit dem root-Passwort zu bestätigen.

In diesem Fall bezieht sich die untenstehende Anleitung auf die Installation auf Ubuntu 16.04 LTS bzw. Debian. Die Anweisungen zur Installation unter anderen Betriebssystemen wie zum Beispiel CentOS, Windows und so weiter kann unter docs.grafana.org gefunden werden.

[DGR17b]

8.8.2 Anleitung

Die Installation von Grafana kann entweder händisch oder mittels `apt-get` durchgeführt werden.

8.8.2.1 händische Installation

Bei der händischen Installation wird zuerst das Paket von Grafana heruntergeladen, was mit folgendem Befehl gemacht wird:

```
wget https://s3-us-west-2.amazonaws.com/grafana-releases/release/grafana_4.2.0_amd64.deb
```

Nach dem Herunterladen des Pakets ist es nötig mit den folgenden Befehlen fortzufahren:

```
apt install -y adduser libfontconfig && dpkg -i grafana_4.2.0_amd64.deb
```

Der erste Befehl installiert zuerst die Schrift-Library und legt einen neuen Benutzer an und der zweite Befehl konfiguriert das Paket von Grafana.

Hierbei ist darauf zu achten, dass die im Befehl angegebene Version von Grafana, welche in diesem Fall 4.2.0 lautet, mit dem Paket, das heruntergeladen wurde, übereinstimmt.



8.8.2.2 Installation mit apt-get

Hierbei wird unter /etc/apt/sources.list der folgende Eintrag zur Paketliste hinzugefügt:

```
deb https://packagecloud.io/grafana/stable/debian/ jessie main
```

Dies fügt die Quelle des Pakets hinzu und dient dazu, dass das Paket über apt-get install gefunden und installiert werden kann.

Würde die Quelle nicht hinzugefügt werden gäbe es später das Problem beim Installieren, dass das Advanced Packaging Tool das Paket nicht finden könnte.

Sollte es gewünscht sein Beta-Versionen der Software zu verwenden so ist im obigen Link das „stable“ gegen ein „testing“ zu tauschen, auf dies wurde in unserem Fall aber verzichtet.

Um sogenannte „signed packages“ zu installieren wird mit folgendem Befehl ein „Package Cloud Key“ hinzugefügt:

```
curl https://packagecloud.io/gpg.key | sudo apt-key add -
```

Um nun die eigentliche Installation durchführen zu können ist es nötig zuerst die Quellen zu aktualisieren und danach Grafana zu installieren. Hierzu wurden die folgenden beiden Befehle per „&&“ verbunden:

```
apt update && apt install Grafana
```

Anschließend wird das Service noch gestartet, was über service grafana-server start funktioniert. Optional kann Grafana auch direkt beim Start des Systems gestartet werden, was über update-rc.d grafana-server defaults aktiviert wird.



Um den Start der Software über „systemd“ zu ermöglichen werden folgende Befehle benötigt:

```
systemctl daemon-reload  
systemctl start grafana-server  
systemctl status grafana-server
```

Der letzte Befehl zeigt lediglich nach dem Start den Status an, um den erfolgreichen Start zu überprüfen.

Das Hinzufügen zum Autostart funktioniert in diesem Fall über `systemctl enable grafana-server.service`.

8.8.3 Verwendung

Grafana besetzt eine grafische Oberfläche, welche über den Webbrowser über Port 3000 erreichbar ist. Mit Hilfe dieses Interfaces kann das ganze Programm eingerichtet werden.

In unserem Fall ist Grafana unter `icinga.dmz.wksimonsfeld.at:3000` aufrufbar.

Beim ersten Aufruf dieser Seite erscheint folgendes Login-Fenster:

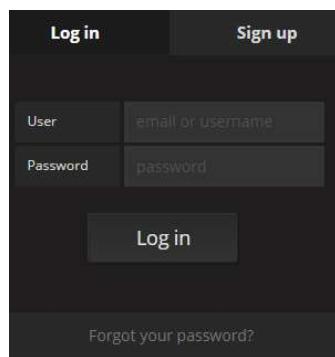


Abbildung 100 – Grafana-Login



Hierbei kann man sich entweder anmelden, wenn schon ein Konto vorhanden ist, oder ein Konto erstellen indem man auf „Sign up“ klickt. Ist es nötig ein Konto zu erstellen so muss zuerst lediglich die zu verwendende E-Mail-Adresse eingegeben werden. Nach dieser Eingabe erscheint folgendes Feld:

You're almost there.
We just need a couple of more bits of information to finish creating your account.

Your email: [REDACTED]

Your name (optional)

Username [REDACTED]

Password [REDACTED]

Continue

Abbildung 101 – Grafana-Registration

Hier werden nun optional ein Benutzername und verpflichtend ein Passwort vergeben. Nach diesem Schritt ist die Registration abgeschlossen und man kann mit dem Login fortsetzen.

Als ersten Konfigurationsschritt soll eine Datenquelle angegeben werden. Dazu wird in der linken oberen Ecke auf das Symbol von Grafana geklickt, wodurch sich folgendes Drop-Down-Menü öffnet:

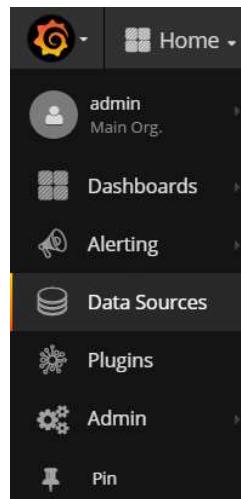


Abbildung 102 – Menü-Grafana



Um nun eine Datenbank als Quelle hinzuzufügen wird auf das im Bild grau hinterlegt Feld „Data Sources“ geklickt. Danach erscheint ein Fenster, in dem folgender Button zu sehen ist:



Abbildung 103 – Datenquelle hinzufügen

Nachdem auf „Add data source“ geklickt wurde ergibt sich folgendes Bild:

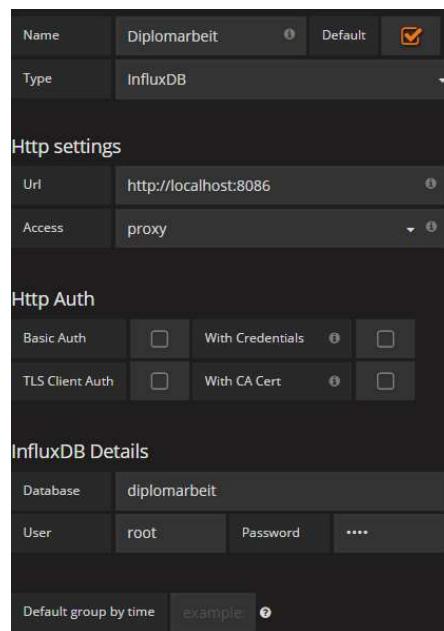


Abbildung 104 – Quelle konfigurieren

Hier ist es wichtig bei „Type“ den Typen „InfluxDB“ auszuwählen, bei der Url die des localhosts anzugeben und in unserem Fall bei „Access“ auf „proxy“ zu stellen.

Bei „direct“ würde Grafana die URL direkt vom Browser verwenden, bei „proxy“ wird die Anfrage durch einen Proxy gefiltert.

Weiters muss ganz unten bei „InfluxDB Details“ der Name der Datenbank sowie Benutzername und Passwort eingegeben werden.



Wenn Grafana zur Datenbank eine Verbindung herstellen kann wird dies mit folgender Ausgabe am Bildschirm bestätigt:



Abbildung 105 – Hinzufügen erfolgreich

Sobald die Datenquelle hinzugefügt wurde kann das erste Dashboard zur Anzeige erstellt werden. Dazu klickt man wieder auf das Grafana-Symbol und wählt dann unter Dashboard-> New

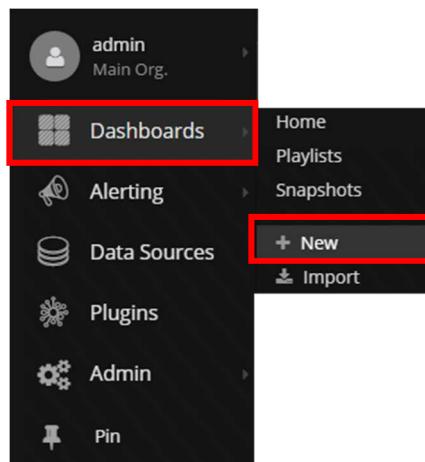


Abbildung 106 – Dashboard erstellen

Ist das Dashboard nun angelegt wird eine Reihe von Elementen vorgeschlagen, welche hinzugefügt werden können. In diesem Fall wird „Graph“ gewählt, da die Daten aus der Datenbank in einem Diagramm dargestellt werden sollen.

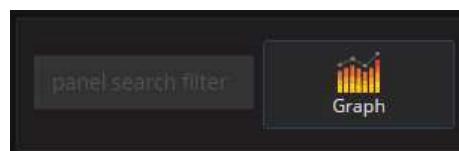


Abbildung 107 – Graph einfügen



Nach dem Klick auf Graph wird ein leeres Diagramm erstellt. Um dieses nun zu konfigurieren wird auf „Panel Title“ geklickt, wodurch sich folgende Möglichkeiten ergeben:

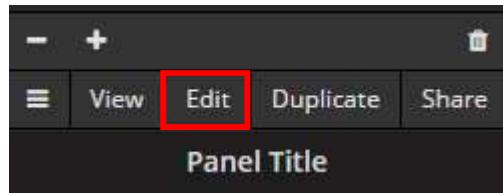


Abbildung 108 – Panel bearbeiten

Über „Edit“ kann das aktuelle Diagramm bearbeitet werden, im Reiter „General“ kann unter „Title“ der Diagrammtitel hinzugefügt werden. Dieser erscheint zentriert über dem Graphen und gibt diesem einen Namen.

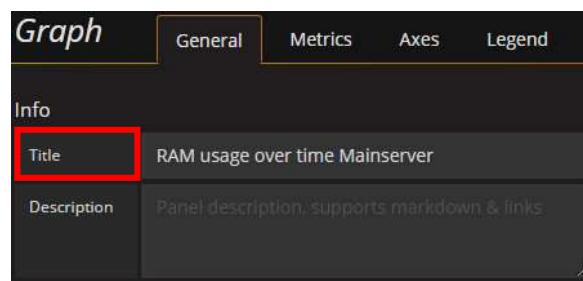


Abbildung 109 – Graph benennen

Unter „Description“ kann eine Beschreibung über das Diagramm hinzugefügt werden, diese wird angezeigt, wenn man den Mauszeiger über das Rufzeichen in der linken oberen Ecke des „Panels“ stellt.

Um nun die Daten aus der Datenbank auszuwählen wird auf den Reiter „Metrics“ gewechselt. Im folgenden Bereich können dann diverse Einstellungen zur Auswahl der Werte getroffen werden. Hierbei entspricht „temp_lager“ dem Measurement, aus welchem die Daten angezeigt werden sollen. Die „WHERE“-Klausel basiert auf demselben Prinzip wie bei InfluxQL, dadurch werden nur die Werte angezeigt, wo der Host „Lager“ und die Region „Ernstbrunn“ heißen.



The screenshot shows the Grafana interface with the 'Metrics' tab selected. The 'WHERE' section is highlighted with a red box. The 'SELECT' section is also highlighted with a red box, showing the 'value' field selected.

Abbildung 110 – Auswahl der Daten

Anschließend ist der anzuzeigende Wert auszuwählen, in diesem Fall stehen zwei Möglichkeiten zur Auswahl. Unter der „SELECT“-Klausel stehen die Fields zur Auswahl, welche das Measurement enthält. Hier wird also nochmal der Unterschied zwischen Tag und Field verdeutlicht. Gewählt wird „value“, da dies der Wert der Temperatur ist und eine Temperaturkurve angezeigt werden soll. Der zweite Wert ist die Systemzeit, welche aus Zwecken der Übersicht zur Datenbank hinzugefügt wurde und lediglich für den Menschen wichtig ist, da das System ohnehin mit dem Zeitstempel arbeitet.

The screenshot shows the Grafana interface with the 'SELECT' section highlighted. The 'value' field is highlighted with a red box.

Abbildung 111 – Auswahl der Werte

Die Einstellungen der Achsen werden gleich neben „Metrics“ unter „Axes“ gefunden. Im Fall der RAM-Kurve ist es nötig einen Wertebereich der Achse sowie einen passenden Datentyp einzustellen.

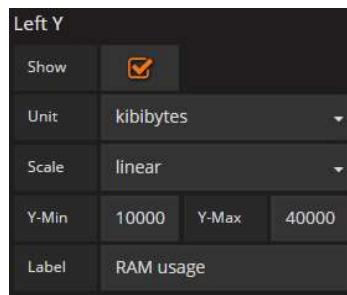


Abbildung 112 – Achseneinstellungen

Unter „Unit“ können viele verschiedene Einheiten gewählt werden, in unserem Fall wurden „kibibytes“ gewählt, da das Script den benutzten RAM in dieser Größe in die Datenbank schreibt. Weiters wurde unter „Y-Min“ und „Y-Max“ ein fixer Wertebereich definiert, da der maximal verfügbare RAM bekannt ist und nicht überschritten werden kann. Dieser Bereich beginnt bei 1GiB RAM, da dieser Wert erfahrungsgemäß nicht unterschritten wird.

Unter „Label“ kann die Achsenbeschriftung hinzugefügt werden, welche in diesem Fall „RAM usage“ heißt.

Darüberhinaus können in der Legende diverse Werte wie zum Beispiel Min, Max, Durchschnitt und so weiter angezeigt werden.

Diese können im Reiter „Legend“ festgelegt werden, welcher folgendermaßen aussieht:

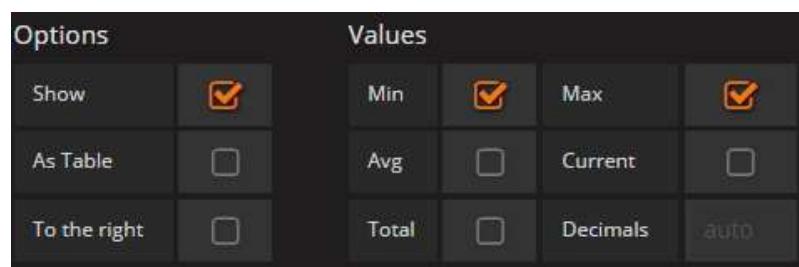


Abbildung 113 – Einstellung Legende

Bei „Options“ kann die Legende selbst aktiviert oder deaktiviert werden, indem man entweder den Haken bei „Show“ setzt oder nicht. „As Table“ bewirkt simpel eine leicht abgeänderte Anzeige der gewählten Werte und „To the right“ verschiebt die Legende von links unten rechts neben das Diagramm.



8.9 Python-Script

8.9.1 Allgemeines

Dieses Script übernimmt die Abfrage der Clients per SNMP, greift anschließend auf den Wert zu und speichert die Daten in unterschiedlichen Measurements ab.

Der prinzipielle Aufbau des Scripts kann folgendermaßen dargestellt werden:

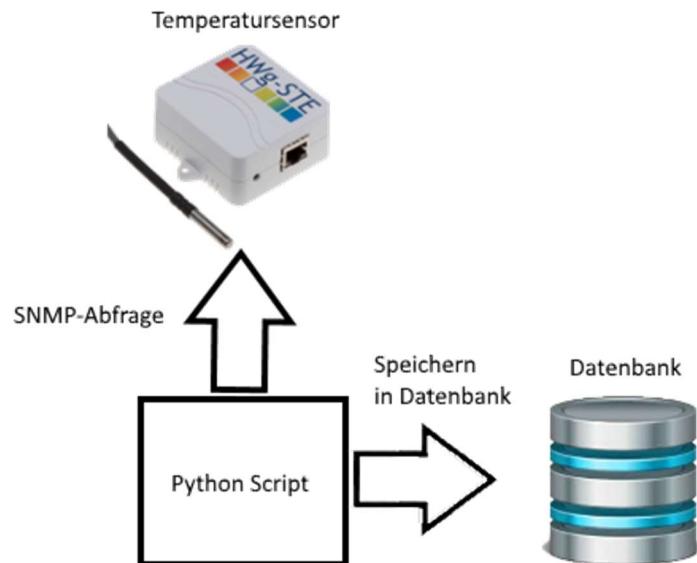


Abbildung 114 – Blockschaltbild Script

Dieses Bild zeigt den Ablauf der Abfrage des Temperatursensors. Zuerst wird die SNMP-Abfrage durch den Aufruf einer Klasse durchgeführt, welche diverse Werte als Instanz zurückliefert. Diese Instanz enthält sämtliche Werte über das Gerät. Bevor nun das Speichern durchgeführt werden kann, müssen die richtigen Daten gefiltert werden. In diesem Fall wird lediglich der Temperaturwert benötigt, da der Client bekannt ist. Zum Schluss speichert das Script sämtliche relevanten Daten in die Datenbank.



8.9.2 Code

Begonnen wird ein Script in der ersten Zeile immer mit der Angabe des zu verwendenden Interpreters. In diesem Fall wird der Python-Interpreter verwendet, welcher sich im Verzeichnis „/usr/bin“ befindet. Direkt danach folgt meist eine Reihe von Kommentaren, welche den Autor des Scripts, die Version, Kontaktdaten, den Zweck und diverse andere Informationen enthalten.

```
#!/usr/bin/python
#by Stephan Schrenk, V2.0
#contact: stephan@schenknet.at
#monitoring of clients
```

Abbildung 115 – Header Script

Die Angabe des Interpreters kann anhand der Zeichenfolge von „#!“ erkannt werden, die darunterliegenden Zeilen, welche lediglich ein Raute-Symbol am Beginn der Zeile stehen haben, sind Kommentare.

Mittels `pydoc` kann die Dokumentation einzelner Module über die Shell aufgerufen werden. Die Dokumentation über das verwendete Modul „time“ kann zum Beispiel mit `pydoc time` am Bildschirm ausgegeben werden. Hier findet man sämtliche Informationen über das verwendete Modul, was zur Fehlersuche sehr hilfreich sein kann. [DPY17]

Direkt unter diesen Informationen steht das Importieren der verwendeten Module. Diese werden folgendermaßen eingefügt:

```
import time
from influxdb import InfluxDBClient
from easysnmp import snmp_get
```

Abbildung 116 – Imports

Da aus den Modulen „influxdb“ und „easysnmp“ jeweils nicht das gesamte Modul benötigt wird wurde hier die Form `from modul import name` verwendet.



Anschließend werden in diesem Fall sämtliche verwendeten Variablen definiert und gleichzeitig als Wert „0“ zugewiesen, da andernfalls ein zufälliger Wert enthalten wäre, welcher mit den tatsächlichen Werten verwechselt werden könnte. Durch diese Zuweisung wird aber erkannt, ob die Variable verändert wurde oder nicht und somit auch, ob die Abfrage und der Zugriff stimmen oder nicht.

```
formtime = 0
temp = 0
val = 0
ramall = 0
ram = 0
```

Abbildung 117 – Definition Variablen

Nach dem Anlegen der Variablen wird der Login in die Datenbank durchgeführt, da dieser nur einmalig beim Start des Scripts durchgeführt werden muss.

```
#Accessing database
client = InfluxDBClient('localhost', 8086, 'root', 'root', 'diplomarbeit')
```

Abbildung 118 – Zugriff auf Datenbank

Nach diesem Teil sind die einmalig abzuarbeitenden Befehle fertig und die eigentliche Abfrage beginnt.

Diese wird in einer Endlosschleife ausgeführt und beginnt mit dem Erfassen der aktuellen Systemzeit, welche für den Benutzer lesbar abgespeichert wird.

```
#getting formatted time of the server
formtime= time.asctime(time.localtime(time.time()))
```

Abbildung 119 – Zeiterfassung



Danach wird an jedes implementierte Gerät eine SNMP-Anfrage gesendet und die zurückgelieferten Daten gespeichert. In diesem Fall handelt es sich dabei um zwei Abfragen, wovon eine die aktuelle Temperatur abfragen soll und die andere den momentan verwendeten Arbeitsspeicher des root-Servers.

```
#get snmp data of HWg-STE
temp = snmp_get('1.3.6.1.4.1.21796.4.1.3.1.4.1', hostname='10.48.1.201', community='public', version=1)

#SNMP request root-Server
ramall = snmp_get('.1.3.6.1.4.1.2021.4.6.0', hostname='10.48.10.13', community='public', version=1)
```

Abbildung 120 – SNMP-Abfrage

Hierbei ist es wichtig zu wissen, dass durch den Aufruf der Klasse „snmp_get“ eine Objektinstanz „temp“ dieser Klasse angelegt wird, in welcher unter anderem das Attribut „value“ vorhanden ist, auf welches im folgenden Code zugegriffen wird.

```
#save only the needed value into val by accessing the value object
val = temp.value
ram = ramall.value
```

Abbildung 121 – Zugriff auf Attribut

Um die Daten später in die Datenbank schreiben zu können ist es nötig einen JSON-Body als Struktur zu definieren. Hierbei ist sowohl das Measurement, in das die Daten geschrieben werden sollen, als auch diverse Tags und Fields anzugeben.

JavaScript Object Notation ist ein einfach lesbares Dateiformat, welches zum Datenaustausch zwischen Anwendungen verwendet wird. Prinzipiell ist JSON aber ein von der Programmiersprache unabhängiges Dateiformat und kann beispielsweise auch in diesem Fall mit Python verwendet werden.

Meist wird dieses Dateiformat zur Übertragung zwischen Client und Server, häufig aber auch bei Webanwendungen oder mobilen Apps verwendet. [WIK17h]



Die Syntax der Struktur sieht folgendermaßen aus:

```
json_body = [
    {
        "measurement": "RAM_sat",
        "tags": {
            "host": "Test1",
            "region": "Test2"
        },
        "fields": {
            "SysTime": formtime,
            "RAM": ram
        }
    }
]
```

Abbildung 122 – Struktur-JSON

Als letzte Operation der Abfrage werden noch die Daten in die Datenbank geschrieben, in die zu Beginn des Scripts eingeloggt wird. Hierbei werden sowohl die oben angegebenen Measurements, als auch die Tags und Fields erstellt, falls sie noch nicht existieren.

```
#writing points into database using json format
client.write_points(json_body)

time.sleep(60)
```

Abbildung 123 – Daten schreiben, warten

Am Ende des Scripts wird für 60 Sekunden gewartet, da die Abfrage einmal pro Minute durchgeführt werden soll und die Durchlaufzeit des Programms vernachlässigbar gering ist.



9 Abbildungsverzeichnis

Abbildung 1 - prinzipielle Struktur Netzwerk.....	6
Abbildung 2 – Ping unter Ubuntu.....	8
Abbildung 3 – Begrenzter Ping unter Ubuntu.....	9
Abbildung 4 – Ping unter Windows.....	9
Abbildung 5 – Ping Funktionsweise.....	10
Abbildung 6 – Ping Zeitüberschreitung der Anforderung.....	11
Abbildung 7 – Ping in Icinga2	12
Abbildung 8 – SNMP-Paket-Header	15
Abbildung 9 – SNMP-PDU-Header	16
Abbildung 10 – SNMP-PDU-Body	16
Abbildung 11 – MIB-Baum	17
Abbildung 12 – OID-Notation	19
Abbildung 13 – OID-Referenzen.....	20
Abbildung 14 – Beispiel OIDs	20
Abbildung 15 – Python-Logo	21
Abbildung 16 – InfluxDB-Logo	23
Abbildung 17 – Grafana-Logo.....	25
Abbildung 18 – Graphen Überwachung.....	25
Abbildung 19 – Net-SNMP-Logo.....	26
Abbildung 20 – PuTTY-Logo.....	27
Abbildung 21 – Oberfläche PuTTY.....	28
Abbildung 22 – Icinga2	29
Abbildung 23 – Icinga2 unter Ubuntu	30
Abbildung 24 – Icinga2 unter Windows	30
Abbildung 25 – Datei conf.d.....	31
Abbildung 26 – Datei features-enabled	31
Abbildung 27 – Datei-features available	32
Abbildung 28 – Datei repository.d	32
Abbildung 29 – Datei scripts.....	32
Abbildung 30 – Service "ping6" unter Ubuntu.....	33
Abbildung 31 – Service "ping6" unter Windows.....	33



Abbildung 32 – PC überwacht sich selbst	34
Abbildung 33 – PC überwacht Sensor	34
Abbildung 34 – PC überwacht sich selbst und einen Sensor	35
Abbildung 35 – Zonen	35
Abbildung 36 – Beispiel Host.....	36
Abbildung 37 – Beispiel Standardservice	37
Abbildung 38 – Beispiel eigener Service	37
Abbildung 39 – Code für die E-Mail-Benachrichtigung	39
Abbildung 40 – Konfiguration E-Mail-Adresse	39
Abbildung 41 – Von Icinga2 geschickte E-Mails	39
Abbildung 42 – Inhalt von Icinga2 geschickter E-Mail	40
Abbildung 43 – Icingaweb2 Dashboard.....	41
Abbildung 44 – Icingaweb2 Dashboard Erklärung	41
Abbildung 45 – Icingaweb2-Overview	42
Abbildung 46 – Graphite-Web	43
Abbildung 47 – Graphite-Plugin in Icingaweb2.....	43
Abbildung 48 – Check_MK	44
Abbildung 49 – Check_MK Dashboard.....	45
Abbildung 50 – Check_MK WATO	46
Abbildung 51 – Check_MK Host tags	47
Abbildung 52 – Check_MK Manuelle Servicekonfiguration.....	48
Abbildung 53 – Check_MK Hostzustände	48
Abbildung 54 – Check_MK Servicezustände	49
Abbildung 55 – Check_MK Graphing-Dashboard.....	52
Abbildung 56 – Check_MK Graphing Beispiel	53
Abbildung 57 – Check_MK Graphing Messwerte.....	53
Abbildung 58 - TeamViewer	54
Abbildung 59 – TeamViewer Konfiguration	55
Abbildung 60 – TeamViewer Verbindungsaufbau	56
Abbildung 61 – TeamViewer Remote Ansicht.....	56
Abbildung 62 - Testumgebung 1	57
Abbildung 63 – Putty Verbindung zu Hauptserver.....	58



Abbildung 64 – SSH-Verbindung zu Hauptserver (Konsole)	58
Abbildung 65 – Temperatursensor.....	59
Abbildung 66 – Hostdefinition Temperatursensor	59
Abbildung 67 – Servicedefinition Temperatursensor	61
Abbildung 68-Graph Temperatursensor	62
Abbildung 69 – E-Mail-Benachrichtigung Code Milestone 1	63
Abbildung 70 – Servicedefinition Temperatursensor komplett.....	63
Abbildung 71 – E-Mail-Benachrichtigung Test	64
Abbildung 72 – E-Mail-Benachrichtigung Temperatursensor.....	64
Abbildung 73 - Testumgebung 2	65
Abbildung 74 – Verteiltes Netzwerk Vorprojekt	66
Abbildung 75 – Services Verteiltes Netzwerk Vorprojekt.....	66
Abbildung 76 – Verteiltes Netzwerk Milestone 2	68
Abbildung 77 – APC UPS Network Management Card 2.....	68
Abbildung 78 – CISCO WAP321.....	69
Abbildung 79 – Hostdefinition Milestone 2	69
Abbildung 80 – Servicedefiniton Milestone 2.....	70
Abbildung 81 – Hostdefinition Milestone 2 Windows	70
Abbildung 82 – Overview Hosts Verteiltes Netzwerk	71
Abbildung 83 – Overview Services Verteiltes Netzwerk	71
Abbildung 84 – Overview Services Verteiltes Netzwerk (2).....	72
Abbildung 85 – Hostansicht eines Client Verteiltes Netzwerk.....	72
Abbildung 98 - Zeitaufwand Mujkanovic 13	96
Abbildung 105 - Zeitaufwand Schrenk 7	102
Abbildung 106 – Secret_Key Graphite	110
Abbildung 107 – Datenbankeinstellung Graphite.....	110
Abbildung 108 - Verteiltes Netzwerk: Hiararchie	111
Abbildung 109 - Verteiltes Netzwerk: Erweiterte Hierarchie	112
Abbildung 110 – Download Net-SNMP erfolgreich.....	120
Abbildung 111 – InfluxDB Start.....	126
Abbildung 112 – Anzeige Datenbanken	127
Abbildung 113 – Anzeige Measurements	128



Abbildung 114 – Einträge löschen.....	129
Abbildung 115 – löschen nach Zeit	130
Abbildung 116 – Zugriff auf Datenbank	131
Abbildung 117 – Name JSON Body.....	131
Abbildung 118 – Grafana-Login.....	134
Abbildung 119 – Grafana-Registration.....	135
Abbildung 120 – Menü-Grafana.....	135
Abbildung 121 – Datenquelle hinzufügen.....	136
Abbildung 122 – Quelle konfigurieren	136
Abbildung 123 – Hinzufügen erfolgreich.....	137
Abbildung 124 – Dashboard erstellen.....	137
Abbildung 125 – Graph einfügen	137
Abbildung 126 – Panel bearbeiten.....	138
Abbildung 127 – Graph benennen	138
Abbildung 128 – Auswahl der Daten.....	139
Abbildung 129 – Auswahl der Werte	139
Abbildung 130 – Achseneinstellungen	140
Abbildung 131 – Einstellung Legende	140
Abbildung 132 – Blockschaltbild Script	141
Abbildung 133 – Header Script.....	142
Abbildung 134 – Imports	142
Abbildung 135 – Definition Variablen	143
Abbildung 136 – Zugriff auf Datenbank	143
Abbildung 137 – Zeiterfassung.....	143
Abbildung 138 – SNMP-Abfrage.....	144
Abbildung 139 – Zugriff auf Attribut	144
Abbildung 140 – Struktur-JSON.....	145
Abbildung 141 – Daten schreiben, warten.....	145



10 Quellenverzeichnis

[WIK17a]:

https://de.wikipedia.org/wiki/Simple_Network_Management_Protocol/

Letzter Aufruf am 29.03.2017.

[WIK17b]:

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

Letzter Aufruf am 03.04.2017.

[WIK17c]:

https://de.wikipedia.org/wiki/Management_Information_Base/

Letzter Aufruf am 25.03.2017.

[WIK17d]:

https://de.wikipedia.org/wiki/Internet_Assigned_Numbers_Authority/

Letzter Aufruf am 25.03.2017.

[WIK17e]:

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

Letzter Aufruf am 10.02.2017.

[WIK17f]:

<https://de.wikipedia.org/wiki/CRUD/>

Letzter Aufruf am 28.03.2017.



[WIK17g]:

<https://de.wikipedia.org/wiki/PuTTY/>

Letzter Aufruf am 23.03.2017.

[WIK17h]:

https://de.wikipedia.org/wiki/JavaScript_Object_Notation/

Letzter Aufruf am 03.04.2017.

[WIK17i]:

[https://de.wikipedia.org/wiki/Ping_\(Daten%C3%BCbertragung\)](https://de.wikipedia.org/wiki/Ping_(Daten%C3%BCbertragung))

Letzter Aufruf am 04.03.2017.

[WIK17j]:

https://de.wikipedia.org/wiki/Round_Trip_Delay

Letzter Aufruf am 04.03.2017.

[WIK17k]:

https://de.wikipedia.org/wiki/Object_Identifier

Letzter Aufruf am 11.03.2017.

[WIK17l]:

<https://de.wikipedia.org/wiki/TeamViewer>

Letzter Aufruf am 11.03.2017



[PYT17]:

<https://www.python.org/doc/essays/blurb/>

Letzter Aufruf am 08.02.2017.

[MIH17]:

http://www.michael-holzapfel.de/progs/python/python_vorteile.htm/

Letzter Aufruf am 08.02.2017.

[DIN17a]:

<https://docs.influxdata.com/influxdb/v0.9/concepts/crosswalk/>

Letzter Aufruf am 23.03.2017.

[DIN17b]:

<https://docs.influxdata.com/influxdb/v0.9/introduction/installation/>

Letzter Aufruf am 03.04.2017.

[DBE17]:

<http://db-engines.com/de/system/InfluxDB/>

Letzter Aufruf am 21.03.2017.

[IPD17]:

<http://influxdb-python.readthedocs.io/en/latest/examples.html/>

Letzter Aufruf am 23.03.2017.



[YTB17]:

<https://www.youtube.com/watch?v=86cOdXXvjhA>

Letzter Aufruf am 17.02.2017.

[DGR17a]:

<http://docs.grafana.org/installation/configuration/>

Letzter Aufruf am 28.03.2017.

[DGR17b]:

<http://docs.grafana.org/installation/>

Letzter Aufruf am 03.04.2017.

[NET17]:

<http://www.net-snmp.org/>

Letzter Aufruf am 03.04.2017

[EAS17a]

<http://easysnmp.readthedocs.io/en/latest/>

Letzter Aufruf am 25.03.2017.

[EAS17b]:

<http://easysnmp.readthedocs.io/en/latest/#installation/>

Letzter Aufruf am 03.04.2017.



[DPY17]:

<https://docs.python.org/3/library/pydoc.html>

Letzter Aufruf am 03.04.2017.

[SCF17]:

http://net-snmp.sourceforge.net/wiki/index.php/Net-Snmp_on_Ubuntu

Letzter Aufruf am 03.04.2017.

[EPE17]:

Icinga2: Ein praktischer Einstieg ins Monitoring

Autoren: Lennart Betz, Thomas Widhalm

Letzte Verwendung am 11.03.2017.

[MKD17]:

https://mathias-kettner.de/check_mk.html

Letzter Aufruf am 11.03.2017

[DIC17a]:

<https://docs.icinga.com/icinga2/latest/doc/module/icinga2/toc>

Letzter Aufruf am 25.03.2017

[DIC17b]

<https://docs.icinga.com/icinga2/latest/doc/module/icinga2/chapter/distributed-monitoring>

Letzter Aufruf am 25.03.2017



[TKC17]:

https://www.thomas-krenn.com/de/wiki/Icinga_Web_2_mit_Icinga_2_verwenden

Letzter Aufruf am 25.03.2017

[LFD17]:

<https://www.linuxfrickeln.de/visualisierung-von-performancedaten-in-der-icingaweb2-gui/>

Letzter Aufruf am 25.03.2017