Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences

Linux Monitoring

Friday, 14th October 2016 14:00-16:00

BFH – Conference Room C.O.R.A. (Dammweg 3, 1st. Floor, Bern)

Andreas Kreuzer <andreas.kreuzer@bfh.ch>

Daniel Baumann <daniel.baumann@bfh.ch>

David Kunz <david.kunz@bfh.ch>

Philipp Pluess <philipp.pluess@bfh.ch>

Sakirnth Nagarasa <sakirnth.nagarasa@bfh.ch>

Overview

- **Introduction**
  - **Linux News**
  - **Linux Container**
- **Monitoring**
  - **Realtime with Netdata**
  - **Blackbox with Icinga**
  - **Whitebox with Graphite/Grafana**
- **Break**
- **Logging**
  - **Centralized with Elastic Stack**
- **Future (2017)**
  - **Ceph, OpenStack, and Ansible**

Linux News

Linux News

- **archive-key rollover**
- **dropping jessie support, sort of**

Linux News

- **archive-key rollover**
- **dropping jessie support, sort of**

- **Winter is comming.. or.. remember the 5th of November:**
  - **Debian 9 (stretch) is freezing and we are ready**
  - **container-server images end of October**
  - **gnome-desktop images end of November**

Linux News

- **archive-key rollover**
- **dropping jessie support, sort of**

- **Winter is comming.. or.. remember the 5th of November:**
  - **Debian 9 (stretch) is freezing and we are ready**
  - **container-server images end of October**
  - **gnome-desktop images end of November**

- **Forced upgrades to stretch for all Linux systems in 2017 (with help from us, of course)**

Linux Container

container-shell

Demo

- **Roadmap**
  - **maintaining container-tools LTS branch for stretch**
  - **Publishing container-tools configs as (generic) appliance images starting November**
  - **with freeze of stretch, supporting systemd-networkd only in newer container-tools**

Realtime Monitoring with Netdata

Netdata

- **Simple…**
  - **C, (almost) no depends**
  - **5'000 graphs by default, no configuration needed**
  - **python or nodejs for plugins**
- **Realtime**
- **Distributed**
- **Registry**

Netdata Demo

Blackbox Monitoring with Icinga

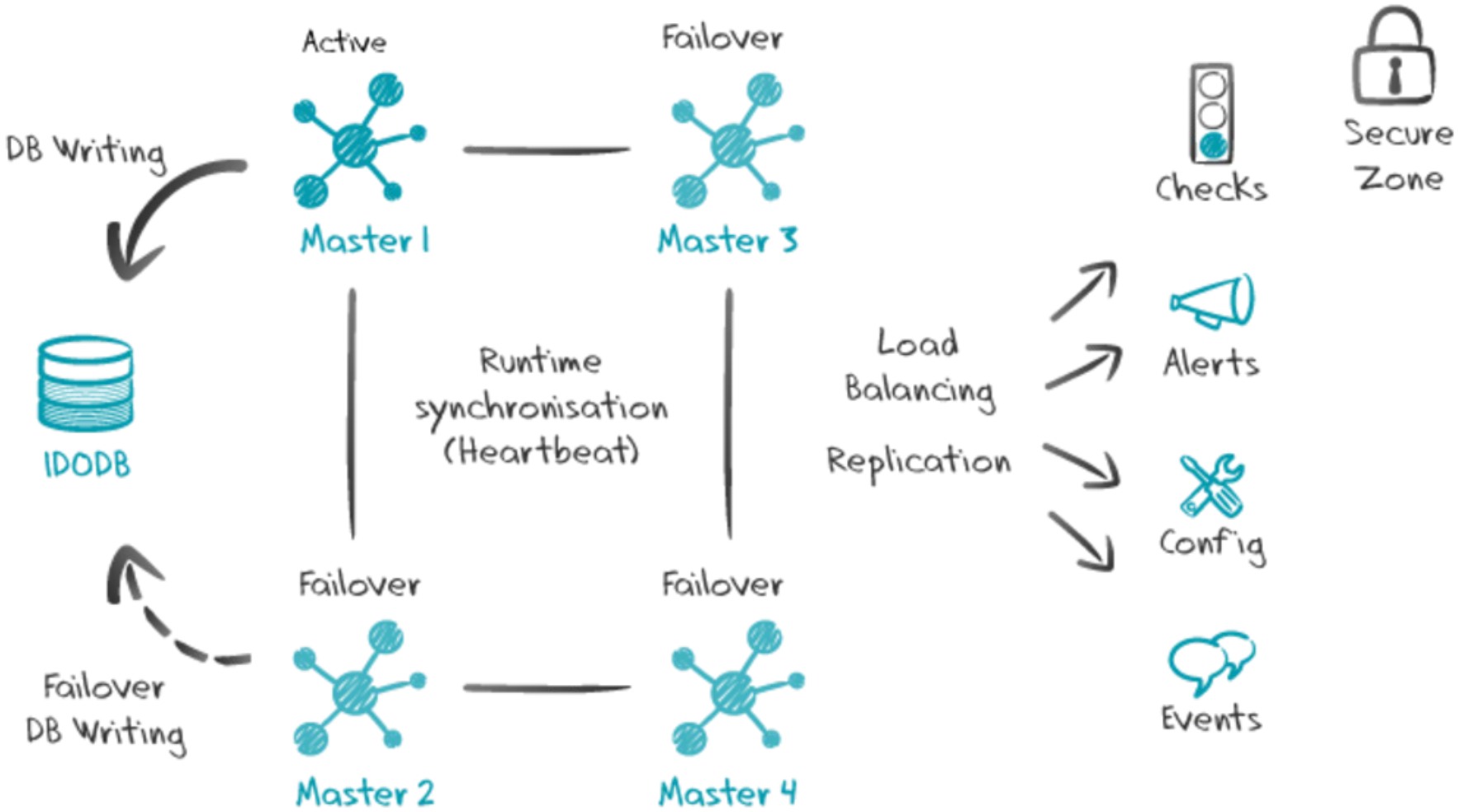
Different Monitoring

- **Blackbox monitoring**
  - **Object is controlled „indrectly“, e.g. http-check, ping, SNMP**
  - **Results like: apache is running/is not running**
  - **Software like Icinga**
- **Whitebox monitoring**
  - **Object is checked „directly“, e.g. Service Checks**
  - **Results like: apache is (too) slow**
  - **Software like Graphite or Grafana**
- **We use whitebox and blackbox for longtime monitoring**

Icinga 2: News since Icinga 1

- **Powerful (very much compared to Nagios)**
- **Scalable**
- **High availability**
- **Checks Object-bound or global**
- **Icinga Web 2**
- **Business process integration**
- **And much more...**

How it works


Active
Failover
DB Writing
Master 1
Master 3
Checks
Secure Zone
IDODB
Runtime synchronisation (Heartbeat)
Load Balancing
Replication
Alerts
Config
Events
Failover
Failover
Failover DB Writing
Master 2
Master 4

Status Monitoring

- **Possible States**

- **Host**
  - **Up**
  - **Down**
  - **Unreachable (is calculated from dependencies)**
- **Service**
  - **OK**
  - **Warning**
  - **Critical**
  - **Unknown**

Key Features and Operation

- **Icinga Core**
- **Icinga IDO**
- **Icinga DB (PostgreSQL/MySQL)**
- **Frond End (Icinga ClassicUI, Icinga Web 2)**
- **API**
- **Client (Agent)**

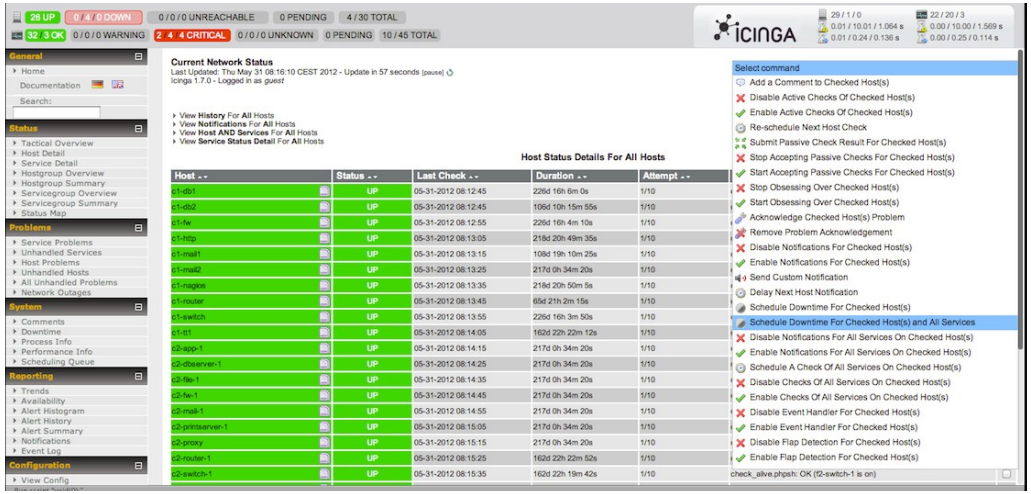
Icinga Client

- **Client Operating Systems**
  - **Linux**
  - **Windows**
- **Monitoring possible by...**
  - **Icinga Agent (Linux, Windows); Windows from 2008/Vista and up with .NET 2.0**
- **NRPE**
- **NSClient++ (Windows)**
- **SSH**
- **SNMP**

Front Ends

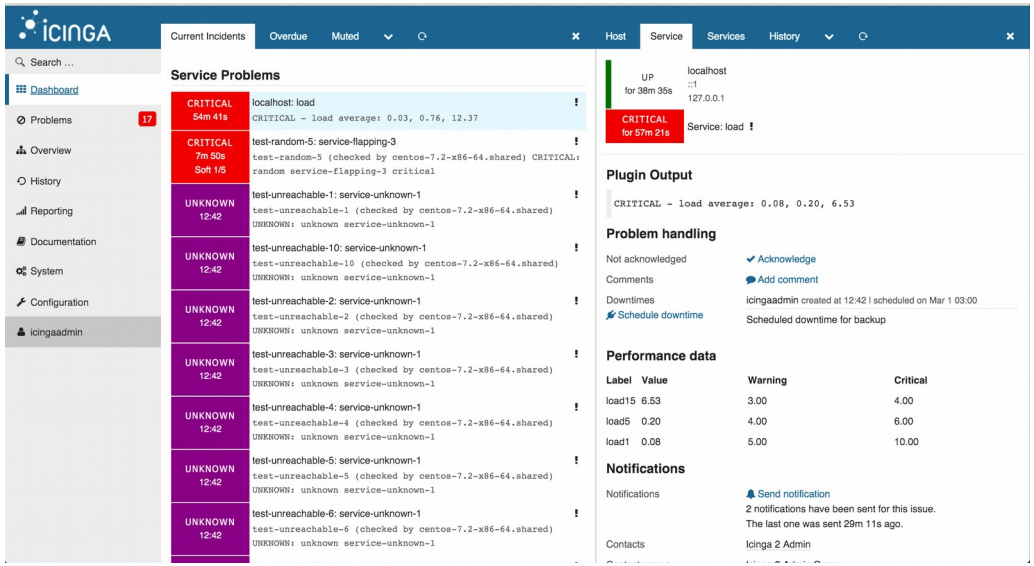
- **Icinga 2 ClassicUI**

  (Basic Authentication)



- **Icinga Web 2**

  (LDAP Authentication)



Extensions

- **Plugins**
- **Add-on**
  - **notify-poseidon-sms**
  - **isms**
- **Modules for Icinga Web 2**
  - **Business Process**
  - **Generic TTS**
  - **PNP**
  - **NagVis**
  - **Boxydash**
  - **Director**

Host Configuration

template.conf

```
object HostGroup "linux-servers" {
    display_name = "Linux Servers"
    assign where host.vars.os == "Linux"
}
```

host.conf

```
object Host icinga-client.bfh.ch {
    import "generic-host"
    address = "147.87.250.24"
    address6 = " 2001:0db8:85a3:08d3:1319:8a2e:0370:7344 "
    vars.os = "Linux"
}
```

Service Configuration

template.conf

```
template Service "generic-service" {
    max_check_attempts = 5
    check_interval = 1m
    retry_interval = 30s
}
```

service.conf (global)

```
apply Service "ping4" {
    import "generic-service"
    check_command = "ping4"
    assign where host.address
}
```

service.conf (wrt/ client)

```
apply Service "ping4" {
    import "generic-service"
    check_command = "ping4"
    host_name = "icina-client.bfh.ch"
}
```

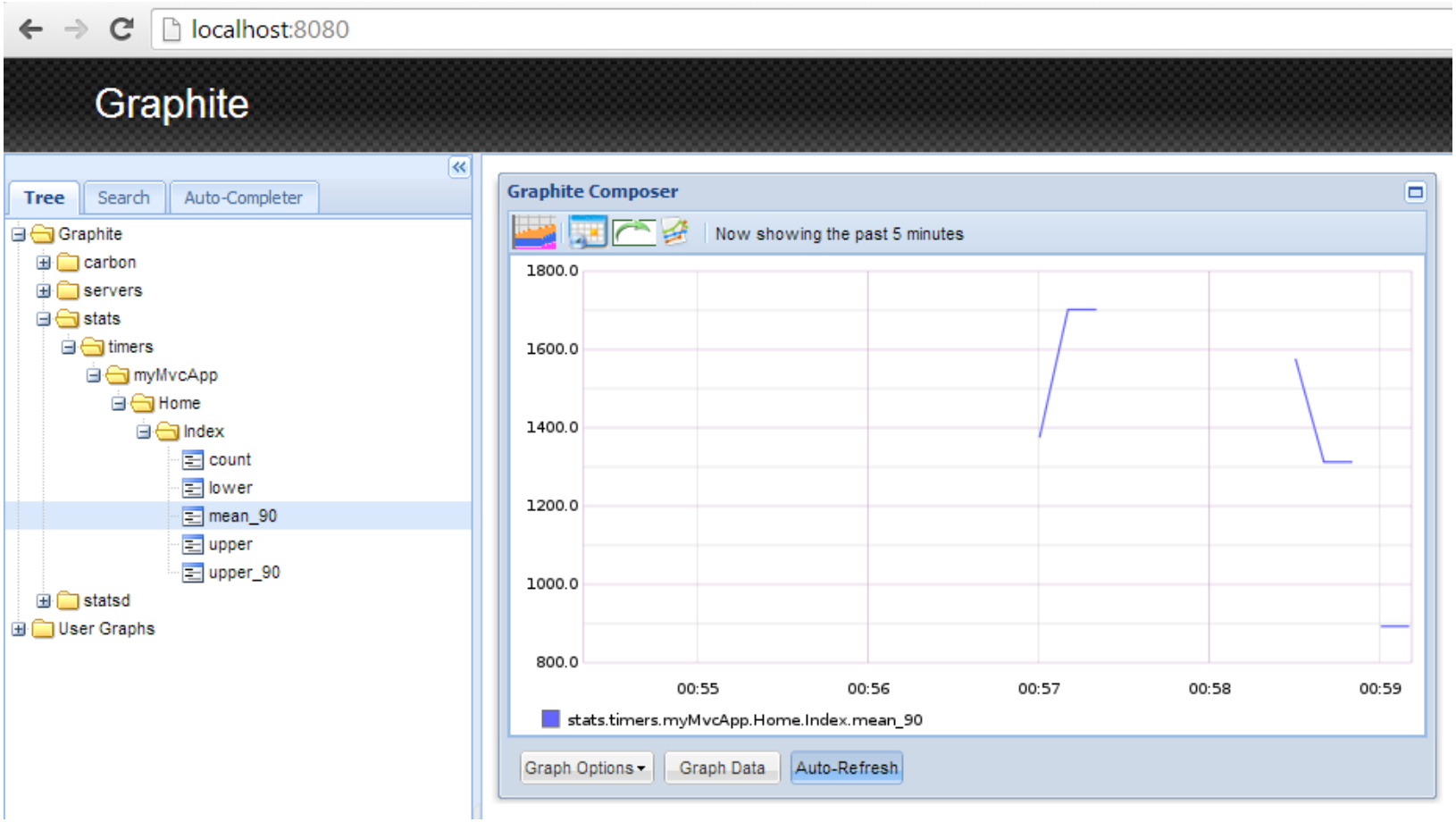
Client Configuration

- **Further configuration possible**
  - `vars.sla = „7x24“`
  - `enable_flapping = 1`
- **Variable** `vars.xy`
- **Downtime**
  - `„2016-10-14“ = „04:29-12:00, 15:00-24:00“`
  - `„december 25“ = „00:00-24:00“`
  - `„monday 2 february – november 8 / 3“ = „00:00-24:00“`
- **Notification**

Icinga Demo

Whitebox Monitoring with Graphite/Grafana

Graphite


localhost:8080
Graphite
Tree
Search
Auto-Completer
Graphite
carbon
servers
stats
timers
myMvcApp
Home
Index
count
lower
mean_90
upper
upper_90
statsd
User Graphs
Graphite Composer
Now showing the past 5 minutes
1800.0
1600.0
1400.0
1200.0
1000.0
800.0
00:55
00:56
00:57
00:58
00:59
stats.timers.myMvcApp.Home.Index.mean_90
Graph Options
Graph Data
Auto-Refresh

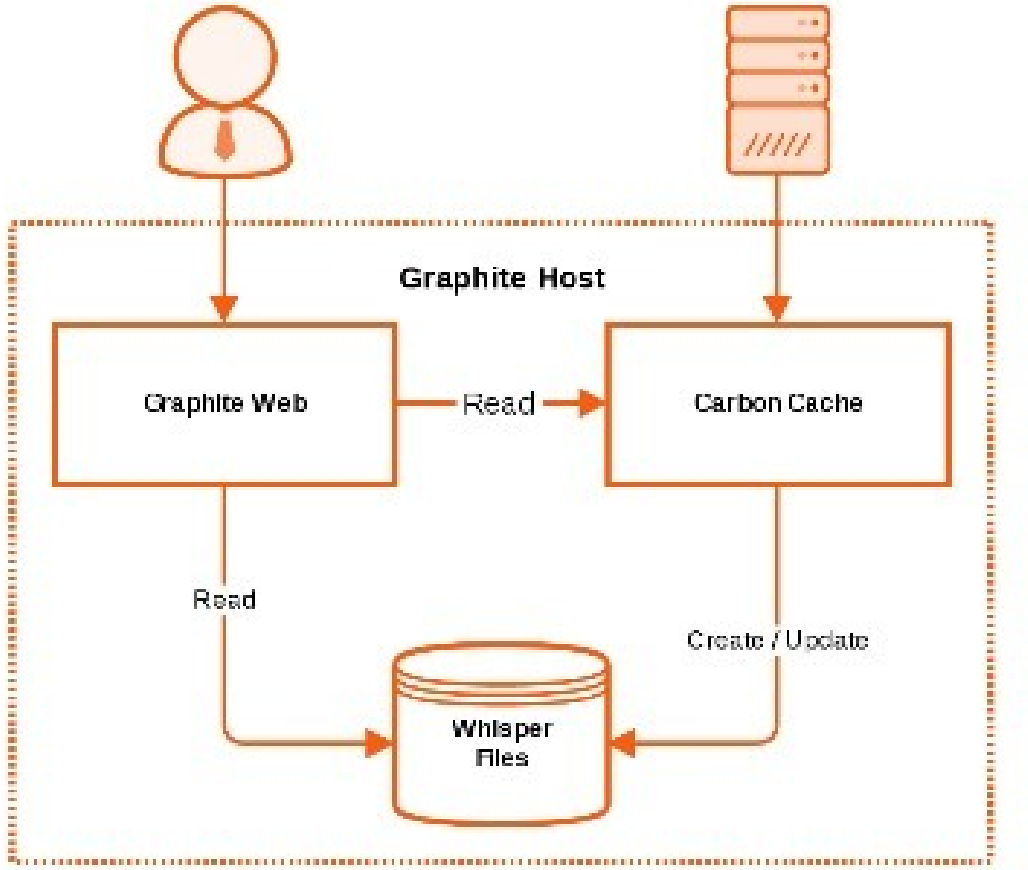
Graphite

- **Graphite is a monitoring tool for the vizualization of graphs**
- **Open Source since 2008**

- **Three main components:**
  - **Carbon: a Daemon who listens on time-series data**
  - **Whisper: a simple Database for storing time-series data**
  - **Graphite: Webapp, a Django Webapp, that depicts graphs using Cairo in realtime**

How it works

Graphite



Graphite Configuration

- **Port input 2003**
- **Output http(s)://**

- **Storage Cycles**

```
/etc/arbon/storage-schemas.conf
  [carbon]
  pattern = ^carbon\.
  retentions = 60:90d

  [default_1min_for_1day]
  pattern = .*
  retentions = 60s:1d
```

Graphite Demo

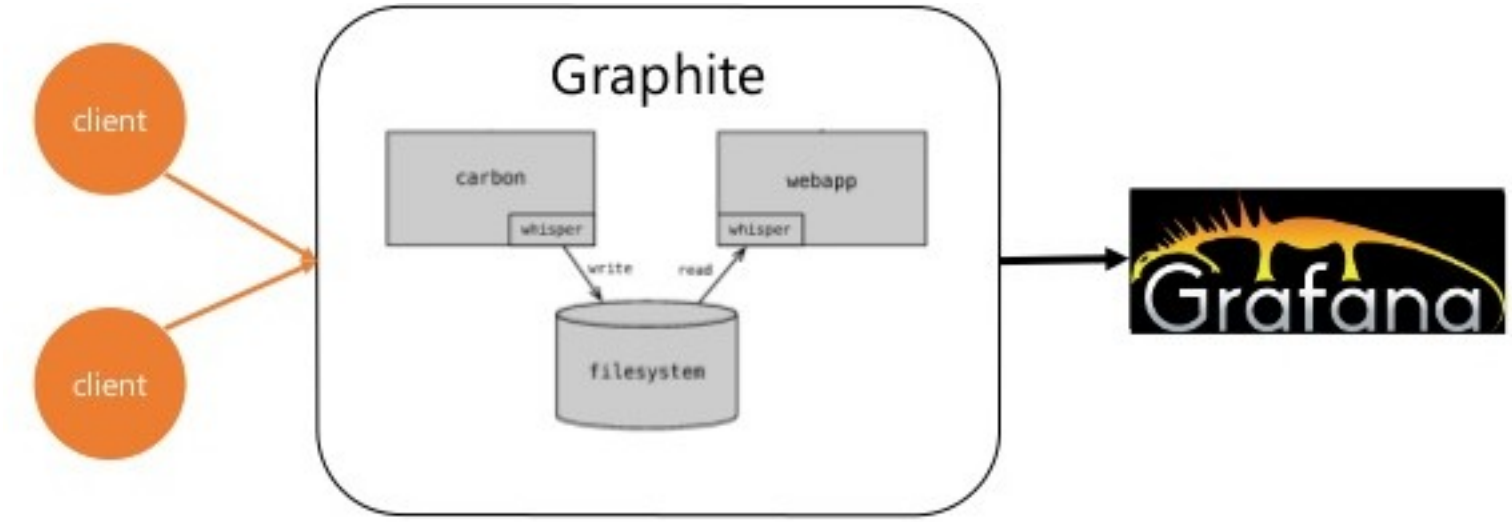
Grafana


worldPing Endpoint: |--- W...
Zoom Out
now-3h to now-1m
Endpoint: imgur_com
Probe(s): All
Protocol: http
Outliers: 100
Endpoint Dashboards
All endpoints > imgur.com
DNS:
Ping:
HTTP:
HTTPS:
Details
Configure
WEB HEALTH (LAST)
OK
WEB UPTIME
100.00%
WEB RESPONSE
125 ms
WEB THROUGHPUT
15.3 Mbps
WEB ERRORS (ALL SELECTED PROBES)
current total
Errors
10:40 10:50 11:00 11:10 11:20 11:30 11:40 11:50 12:00 12:10 12:20 12:30 12:40 12:50 13:00 13:10 13:20 13:30
WEB RESPONSE (ALL SELECTED PROBES)
150 100 50 0
11:00 11:30 12:00 12:30 13:00 13:30
dns connect send wait recv
WEB THROUGHPUT (ALL SELECTED PROBES)
20.00 kbps 19.95 kbps 19.90 kbps 19.85 kbps 19.80 kbps 19.75 kbps 19.70 kbps
57 MiB 48 MiB 38 MiB 29 MiB 19 MiB 10 MiB 0 B
11:00 11:30 12:00 12:30 13:00 13:30
content size
throughput
WEB PERFORMANCE (EACH INDIVIDUAL PROBE)
700 ms 600 ms 500 ms 400 ms 300 ms
latency
avg current
singapore 564.8 ms 548.2 ms
taipei 415.2 ms 404.6 ms
tokyo 344.5 ms 279.1 ms
sydney 8.9 ms 209.3 ms
omaha 114.5 ms 108.3 ms
seattle 64.9 ms 90.6 ms
new-jersey 36.9 ms 63.5 ms
portland 76.1 ms 56.3 ms
frankfurt 57.8 ms 55.3 ms
los-angeles 18.8 ms 40.5 ms
amsterdam 27.3 ms 27.3 ms
paris 36.4 ms 26.3 ms
new-york 56.3 ms 21.5 ms
silicon-valley 9.8 ms 21.1 ms

Features

- **Data Sources**
  - **Supports Graphite, Elasticsearch, Prometheus, InfluxDB, OpenTSDB, and KairosDB out of the box**
- **Authentication**
  - **LDAP, Basic Auth, and Auth Proxy**
- **Dashboards**
  - **scripted (static) vs. database (dynamic)**
- **Snapshot Sharing**
  - **Create and share a fully-interactive graph in 1-click and share with only your team or the world**

How it works


client
client
Graphite
carbon
whisper
webapp
whisper
write
read
filesystem
Grafana

Grafana Demo

Let's take a break...

Centralized Logging with Elastic Stack

Overview

What?

- Centralizing Logs from different sources

How?

- Logstash
- Elasticsearch
- Kibana

Show me some!

- Let‘s find some infos
- And draw some graphs

Centralized Logging

What do we want?

- Gather logfiles from any source
- Put them on a centralized place
- Parse, filter and display the recieved data

How?

- Elastic Stack fits all the needs
- Is open source
- Large userbase

Logstash

What is Logstash?

- Data pipeline to help process logs
- Normalize data
- Extend to custom log formats
- Enrich pipeline with additional plugins
- Write result to multiple output

How do we use it?

- With Logstash we are able to process any data from any source and put it on multiple locations
- At this time we use tcp and udp input as well as winlogbeat and send it to elasticsearch
- https://git.bfh.ch/gitweb/?p=services/it/infrastructure/linux/services/logstash.linux.bfh.ch.git

Elasticsearch

What is Elasticsearch?

- Real-time data
- Massively distributed (not yet used in our case)
- High available
- Full-text search (Apache Lucene)
- Document-oriented (json)
- RESTful API

How do we use it?

- Single node cluster
- Secured behind apache2 (ldap-auth)
- https://elasticsearch.linux.bfh.ch/_plugin/kopf
- https://kibana.linux.bfh.ch/app/sense

Kibana

What is Kibana?

- Flexible analytics and visualization platform
- Real-time summary and charting of streaming data
- Intuitive interface
- Sharing and embedding of dashboards

How do we use it?

- You Know, for Search! :-)
- Let the admin choose what, when, and how

Show me some!

Where can I get it?

- https://elastic.io/products

Find data

- Use discovery in kibana

Draw charts

- Use visualize in kibana

- https://kibana.linux.bfh.ch

Future (2017)

Ceph (1)

- **Facts**
  - **Easy scale-out**
  - **No single point of failure (If properly configured)**
  - **Replica**
  - **Self-healing**

Ceph (1)

- **Facts**
  - **Easy scale-out**
  - **No single point of failure (If properly configured)**
  - **Replica**
  - **Self-healing**

- **Distributed Storage vs. Traditional Storage**
  - **Data is ''distributed''**
  - **Faster when scale out**
  - **Not dependend on one supplier (e.g. NetApp)**

Ceph (2)

- **Daemons**
  - **ceph-mds**
  - **ceph-mon**
  - **ceph-osd**
  - **ceph-rgw**

Ceph (2)

- **Daemons**
  - **ceph-mds**
  - **ceph-mon**
  - **ceph-osd**
  - **ceph-rgw**
- **Storage Types**
  - **Object Storage**
  - **Block Storage**
  - **Filesystem**

OpenStack (1)

- **Facts**
  - **Cloud Environment**
  - **IaaS**
  - **Modulare Architecture**
  - **Self Service**

OpenStack (1)

- **Facts**
  - **Cloud Environment**
  - **IaaS**
  - **Modulare Architecture**
  - **Self Service**
- **Core Services**
  - **Block Storage**
  - **Compute**
  - **Identity**
  - **Image**
  - **Networking**

OpenStack (2)

- **Advantages**
  - **Dashboard is not complicated**
  - **Fast growing comunity**
  - **Fits perfect in an educational instituation (for reasearch)**
  - **Many components & features**
  - **No license fee**
  - **Open Source**
  - **„Self Service“**

Ansible (1)

- **Facts**
  - **Configuring and managing computers**
  - **Free Software platform**
  - **Manage nodes over SSH**

Ansible (1)

- **Facts**
  - **Configuring and managing computers**
  - **Free Software platform**
  - **Manage nodes over SSH**

- **How it works**
  - **Manage over SSH**
  - **No daemon is running**
  - **Single controlling machine**

Ansible (2)

- **Advantages**
  - **Easy to learn**
  - **No agent**
  - **No daemon**
  - **Use SSH for deploy**

Thank You for Your Attention.

❤ Source Code is freely available

git clone git://git.bfh.ch/git/staff/bad9/other/talks.git