

# Monitoring Large-scale OpenACS Applications



Thomas Renner  
Learn@WU Systemmanager

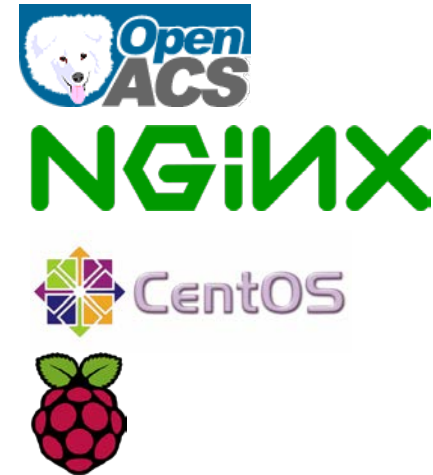
European OpenACS and Tcl Conference,  
Vienna 2022

JUNE 2022



# How to monitor Large-scale OpenACS Applications?

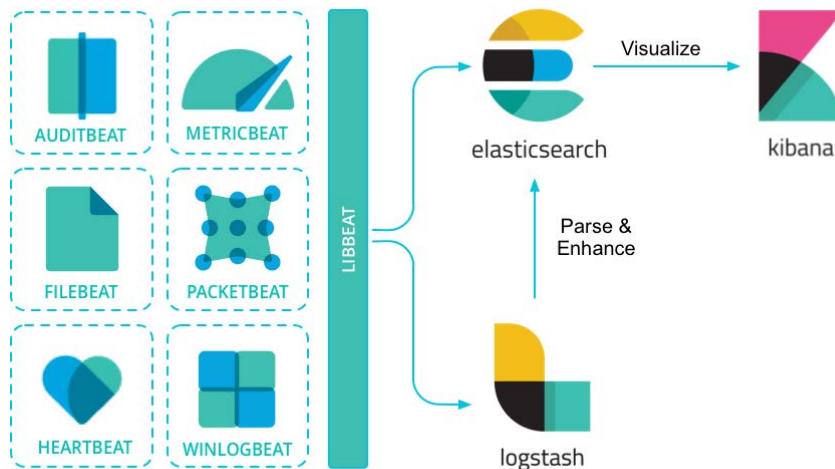
- Goal is to have a live overview of the whole „system/service“ and detect potential flaws before they are reaching the customer
- A „system/service“ consists of various components
  - Web application (LEARN) (OpenACS instance)
  - Proxy server nginx
  - KVM hosts
  - Various virtual servers
  - Network
  - Lecture casting / live streaming system
    - Touch panel (Raspberry PI)
    - Streaming cluster
  - ...
- Single point of log data
  - Diverse data has to be aggregated



# Elastic Search Stack

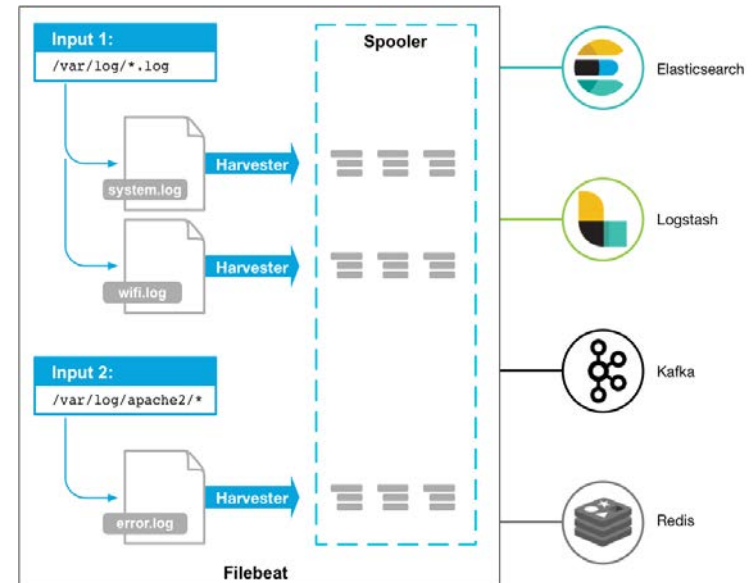


- Beats: Data collection
  - Filebeat for logfiles
  - Metricbeat with many optional modules
    - System: CPU, load, memory, network, IO,...
- Logstash: Data aggregation & processing
- Elasticsearch: Indexing & storage
- Kibana: Analysis & visualisation



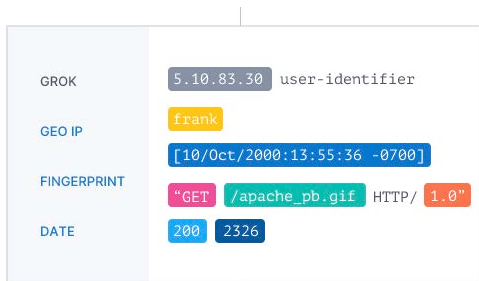
<https://www.elastic.co/guide/en/beats/libbeat/current/beats-reference.html>

- Lightweight shipper for forwarding and centralizing log data
- For each log file filebeat starts a harvester
- Each harvester reads a single log for new content and sends the new log data to the libbeat framework
- Libbeat aggregates the events and sends the aggregated data to the output that is configured for filebeat



<https://www.elastic.co/guide/en/beats/filebeat/8.2/filebeat-overview.html>

- Data processing pipeline
  - Structuring the data
  - Anonymizing the data
- Multitude of sources (inputs)
- Transforming of data (filters)
- Sending it (outputs)



<https://www.elastic.co/de/logstash/>

```

input {
  beats {
    port => 5044
    ssl => true
    ssl_certificate => "/etc/pki/tls/certs/logstash-forwarder.crt"
    ssl_key => "/etc/pki/tls/private/logstash-forwarder.key"
  }
}

filter {
  if [type] == "syslog" {
    grok {
      match => { "message" => "%{SYSLOGTIMESTAMP:syslog_timestamp} %{SYSLOGHOST:syslog_host} %{SYSLOGMESSAGE:message}" }
      add_field => [ "received_at", "%{@timestamp}" ]
      add_field => [ "received_from", "%{host}" ]
    }
    syslog_pri { }
    date {
      match => [ "syslog_timestamp", "MMM d HH:mm:ss", "MMM dd HH:mm:ss" ]
    }
  }
}

output {
  elasticsearch {
    hosts => ["localhost:9200"]
    sniffing => true
    manage_template => false
    index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM}"
    document_type => "%{[@metadata][type]}"
    user => "elastic"
    password => "changeme"
  }
}
    
```

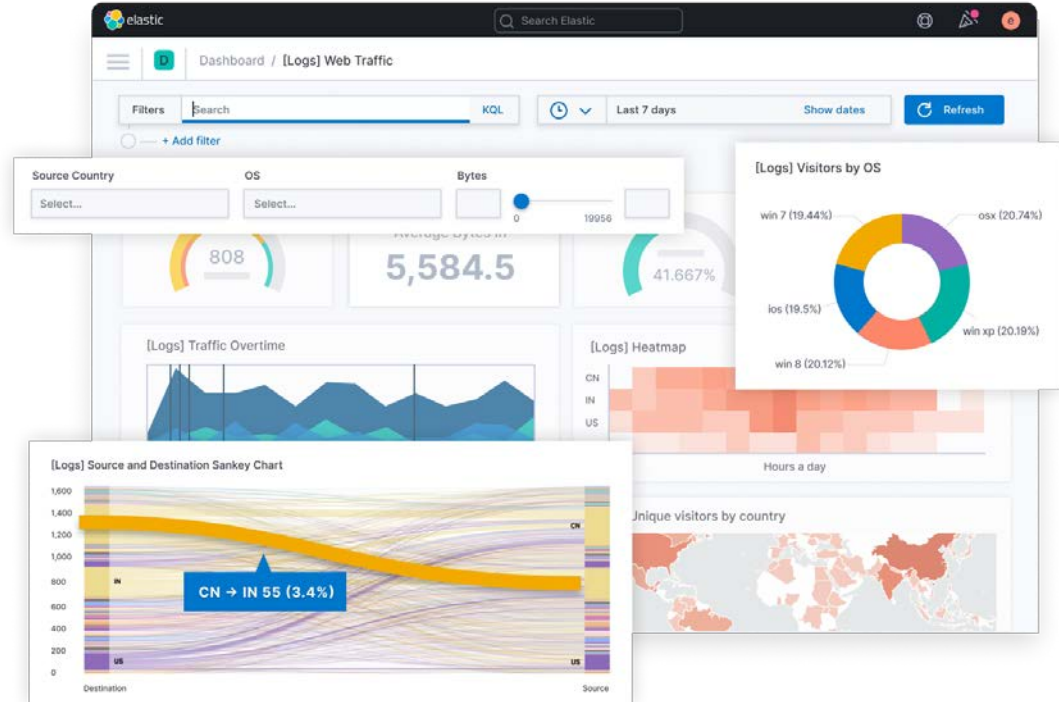
# Elasticsearch Cluster Management



- Cluster consists of one or multiple nodes
- Data is stored in indexes
- An index can have one or more replications
- An index consist of one or multiple shards
- Automatically distributed through the cluster
- Search:
  - RESTful search and analytics engine
  - Perform and combine many types
- Resiliency:
  - Cross-cluster replication
  - Hot backup



- Visualizing:
  - Histograms
  - Line graphs
  - Pie charts
  - ...
- Analyzing
- Machine Learning
- Graphs and networks



<https://www.elastic.co/de/kibana/>



- Open-source platform for monitoring and observability
  - Querying, visualizing, alerting
  - Mixed various data sources: e.g. elastic search
  - Alerting:
    - Visualizing
    - Notifications (Email, Microsoft Teams,....)





# Monitoring screen in LEARN's office

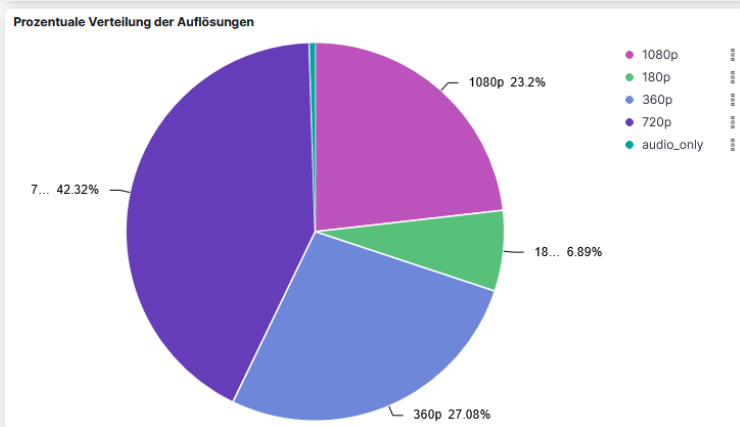
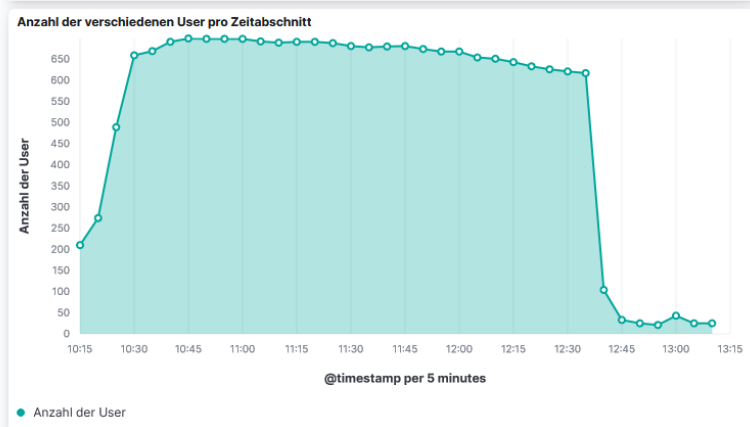
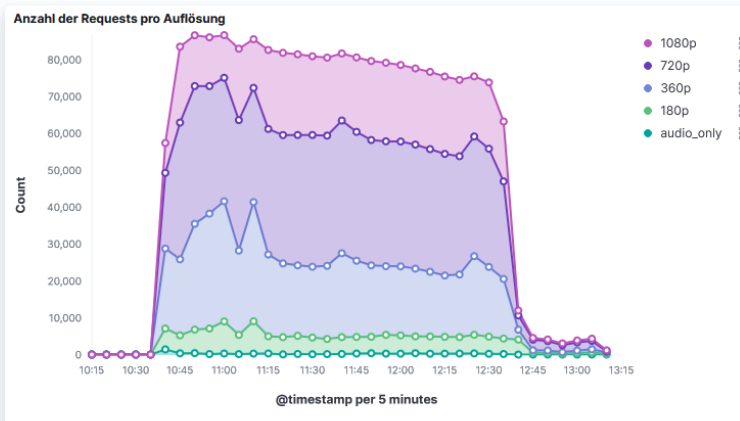
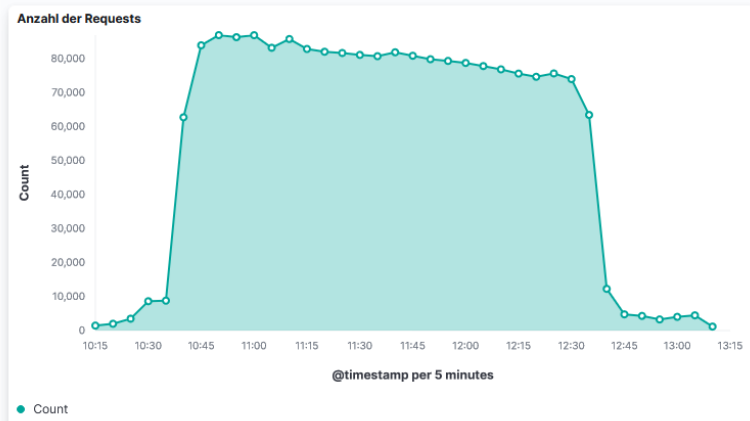


# Kibana: Filtering POST and GET requests for IP address

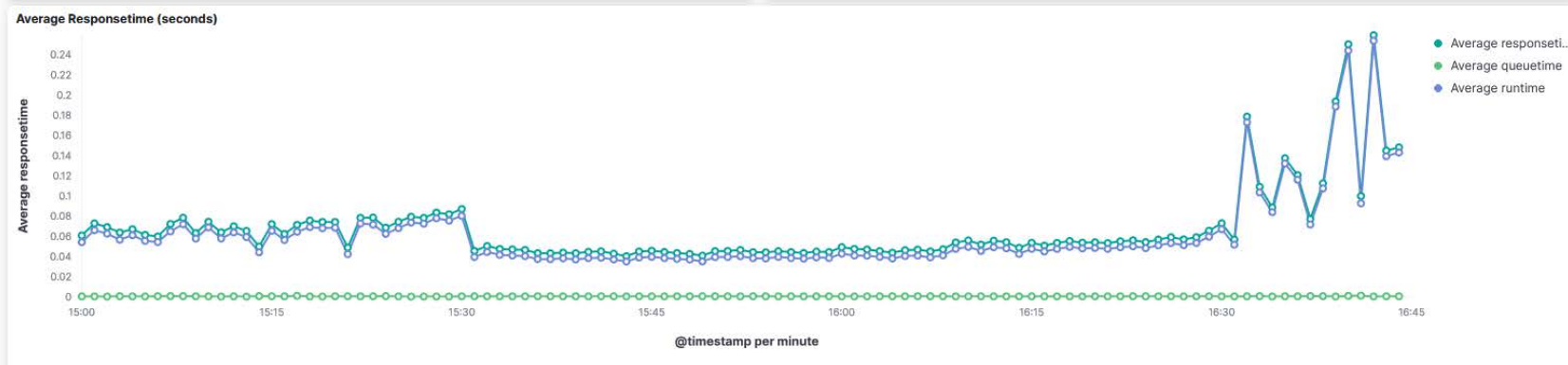
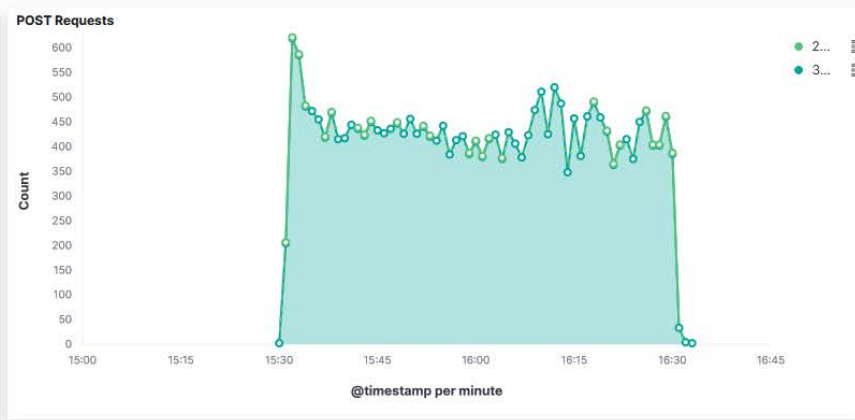
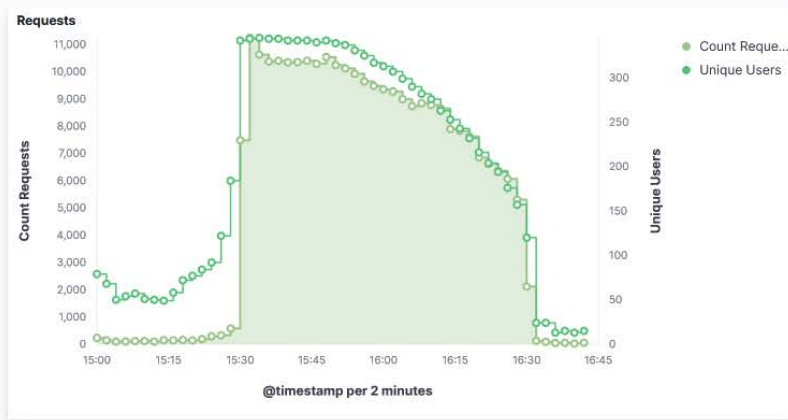
The screenshot displays the Kibana search interface. At the top, the search bar contains the query `client.geo.ip: 137.208.114.28` and `userid: exists`, which are circled in red. The interface shows 2,361 hits for the 'filebeat-\*' index pattern. A bar chart visualizes the request frequency over time, with a peak around 11:00. Below the chart is a table of search results.

Time ↑	verb	request	response	userid
> Jun 22, 2022 @ 09:21:18.000	GET	/	200	"14664491"
> Jun 22, 2022 @ 09:22:37.000	GET	/dotlrn/	200	"14664491"
> Jun 22, 2022 @ 09:22:39.000	GET	/dotlrn/classes/tlf/	302	"14664491"
> Jun 22, 2022 @ 09:22:40.000	GET	/dotlrn/classes/tlf/one-community?page_num=0	200	"14664491"
> Jun 22, 2022 @ 09:22:40.000	GET	/dotlrn/calendar/resources/ical12x12.gif	304	"14664491"
> Jun 22, 2022 @ 09:23:33.000	GET	/dotlrn/	200	"14664491"
> Jun 22, 2022 @ 09:23:39.000	GET	/dotlrn/org/wu/od/3805/3819/	302	"14664491"
> Jun 22, 2022 @ 09:23:39.000	GET	/dotlrn/org/wu/od/3805/3819/one-community?page_num=0	200	"14664491"
> Jun 22, 2022 @ 09:23:44.000	GET	/dotlrn/classes/bw1/	302	"14664491"
> Jun 22, 2022 @ 09:23:44.000	GET	/dotlrn/classes/bw1/one-community?page_num=0	200	"14664491"
> Jun 22, 2022 @ 09:23:44.000	POST	/dotlrn/quick_class	302	"14664491"
> Jun 22, 2022 @ 09:23:53.000	GET	/dotlrn/classes/tlf/	302	"14664491"

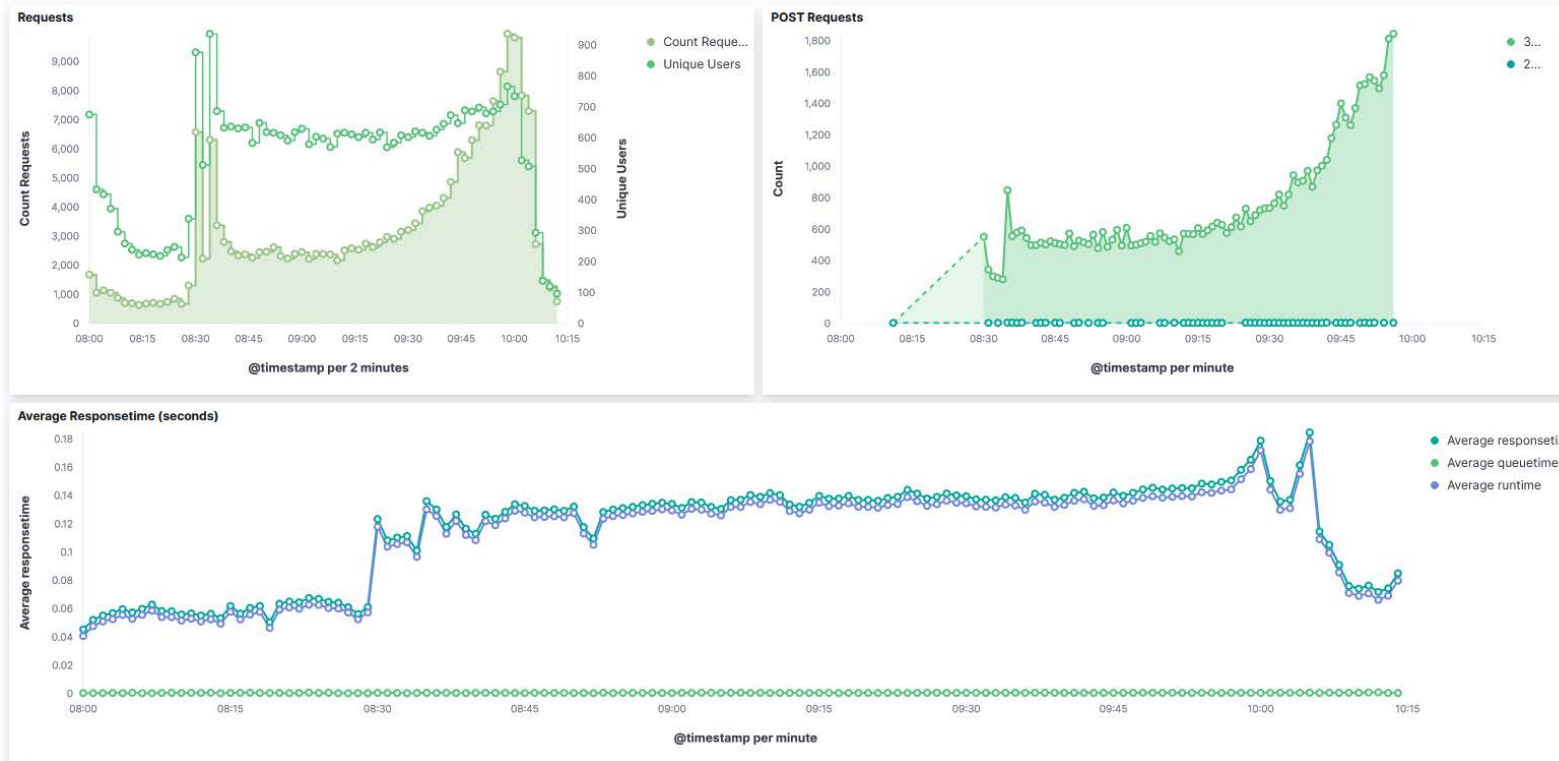
# Kibana Dashboard: Web Streaming



# Kibana Dashboard: Exam 1



# Kibana Dashboard: Exam 2



# Live Demo

# Questions?