Monitoring Large-scale OpenACS Applications

Thomas Renner

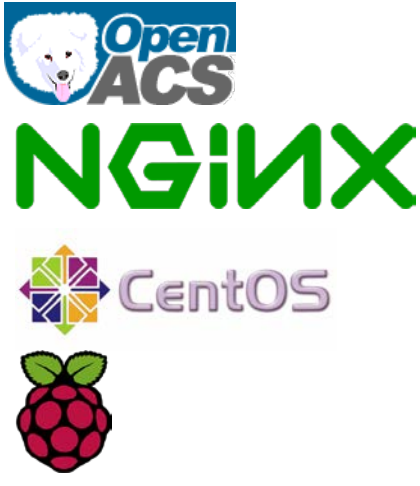
Learn@WU Systemmanager

European OpenACS and Tcl Conference, Vienna 2022

JUNE 2022

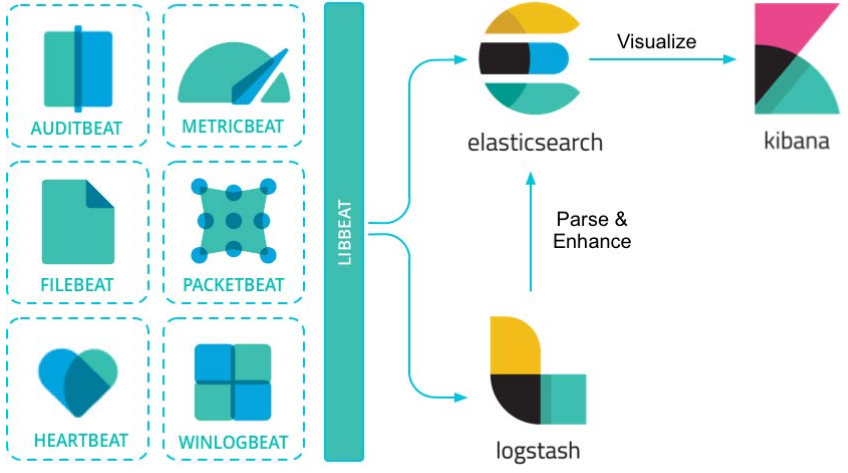
How to monitor Large-scale OpenACS Applications?

- 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

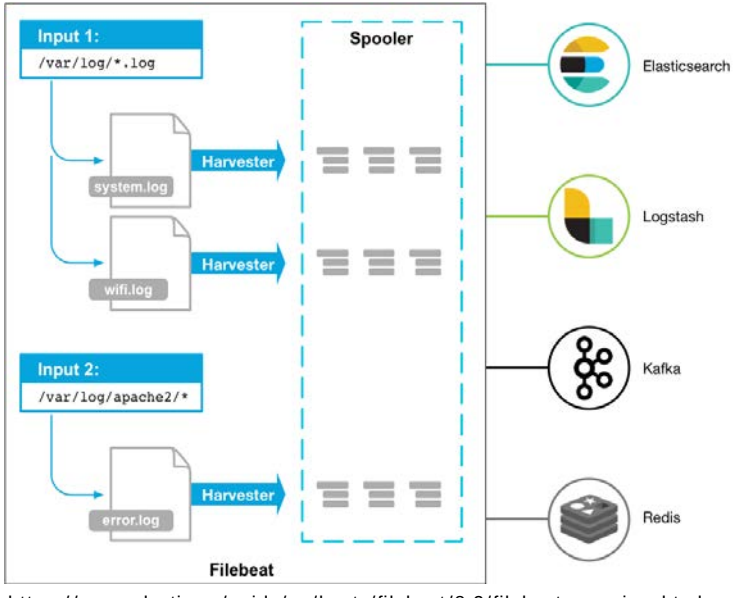
- Beats: Data collection
  - Filebeat for logfiles
  - Metricbeat with many optional modules
    - System: CPU, load, memory, network, IO,...
- Logstash: Data aggregation & processing
- Elasticsearch: Indexing & storage
- Kibana: Analysis & visualisation



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

filebeat

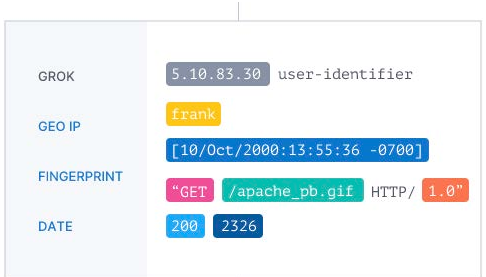
- Lightweight shipper for forwarding and centralizing log data
- For each log file filebeat starts a harvester
- Each harvester reads a single log for new content and sends the new log data to the libbeat framework
- Libbeat aggregates the events and sends the aggregated data to the output that is configured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



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

logstash

- 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/

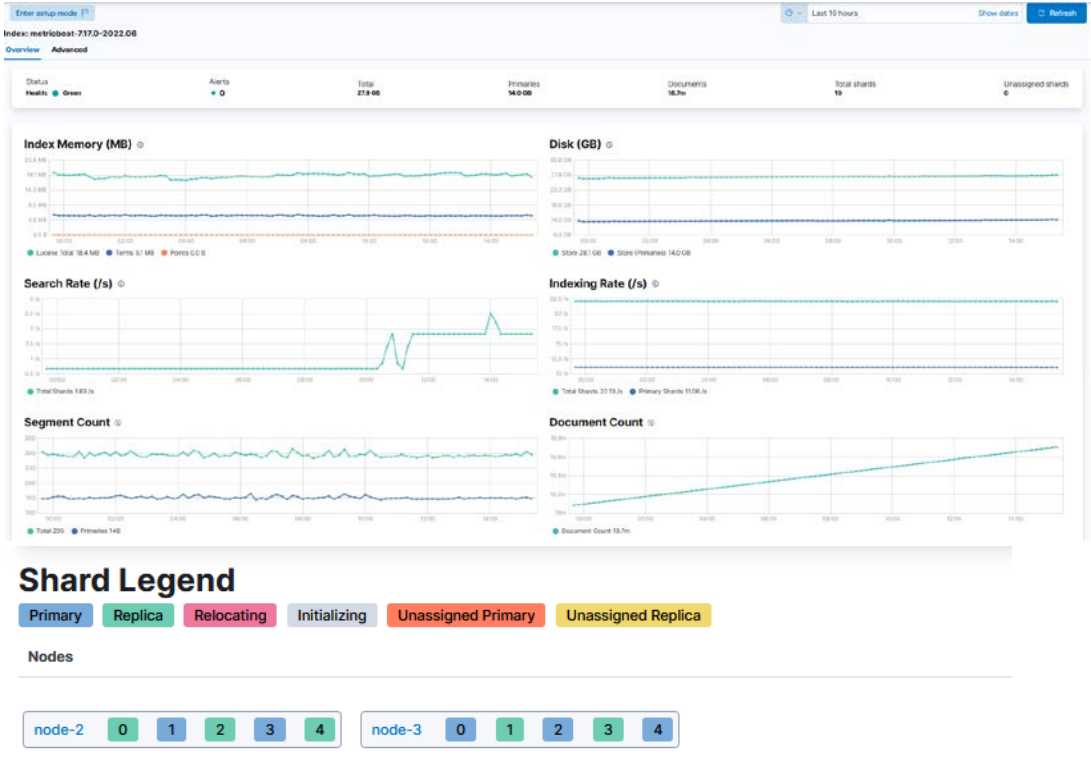
```
input {
  beats {
    port => 5044
    ssl => true
    ssl_certificate => "/etc/pki/tls/certs/logstash-forwarder.crt"
    ssl_key => "/etc/pki/tls/private/logstash-forwarder.key"
  }
}

filter {
  if [type] == "syslog" {
    grok {
      match => { "message" => "%{SYSLOGTIMESTAMP:syslog_timestamp} %{SYSLOGHC
      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 => ["                :9200"]
    sniffing => true
    manage_template => false
    index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM}"
    document_type => "%{[@metadata][type]}"
    user => "                  "
    password => "                "
  }
}
```

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

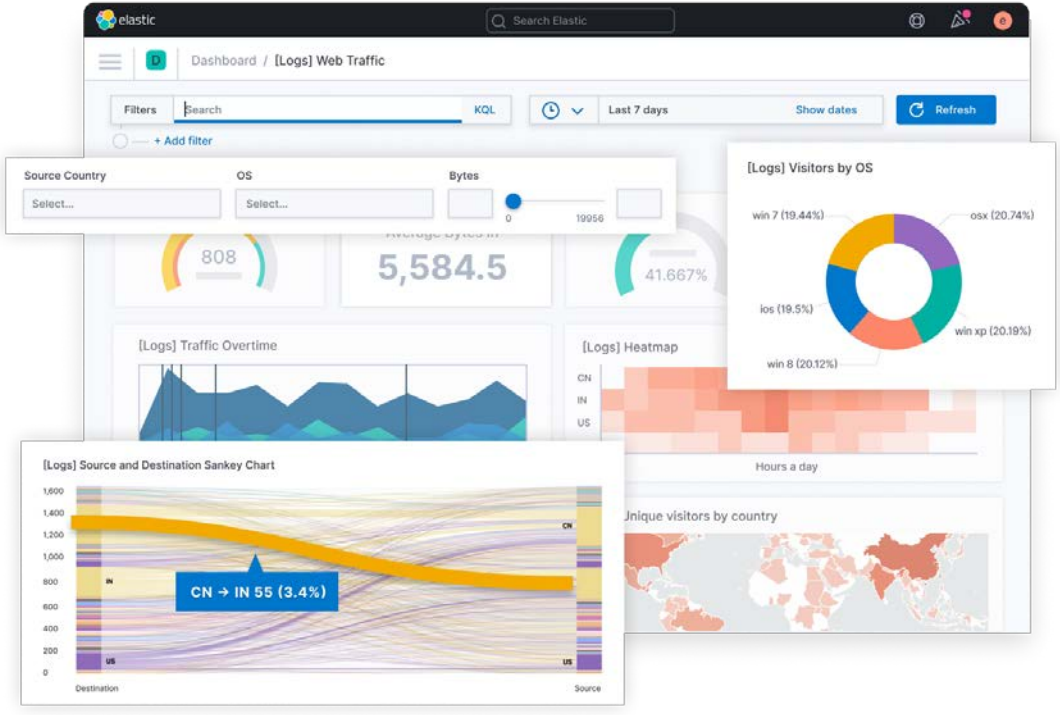
- Search:
  - RESTful search and analytics engine
  - Perform and combine many types

- Resiliency:
  - Cross-cluster replication
  - Hot backup



Kibana

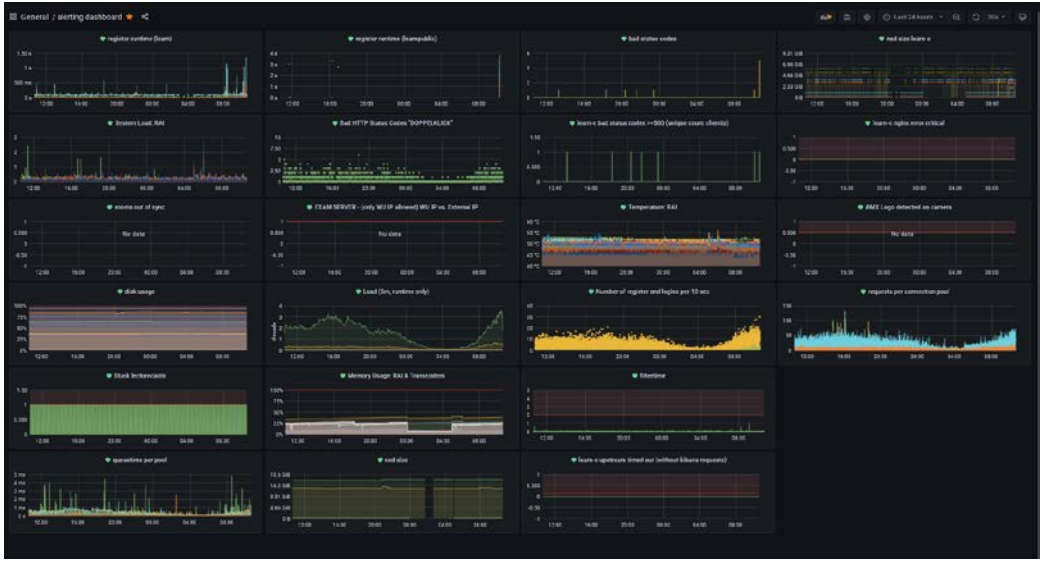
- Visualizing:
  - Histograms
  - Line graphs
  - Pie charts
  - …
- Analyzing
- Machine Learning
- Graphs and networks



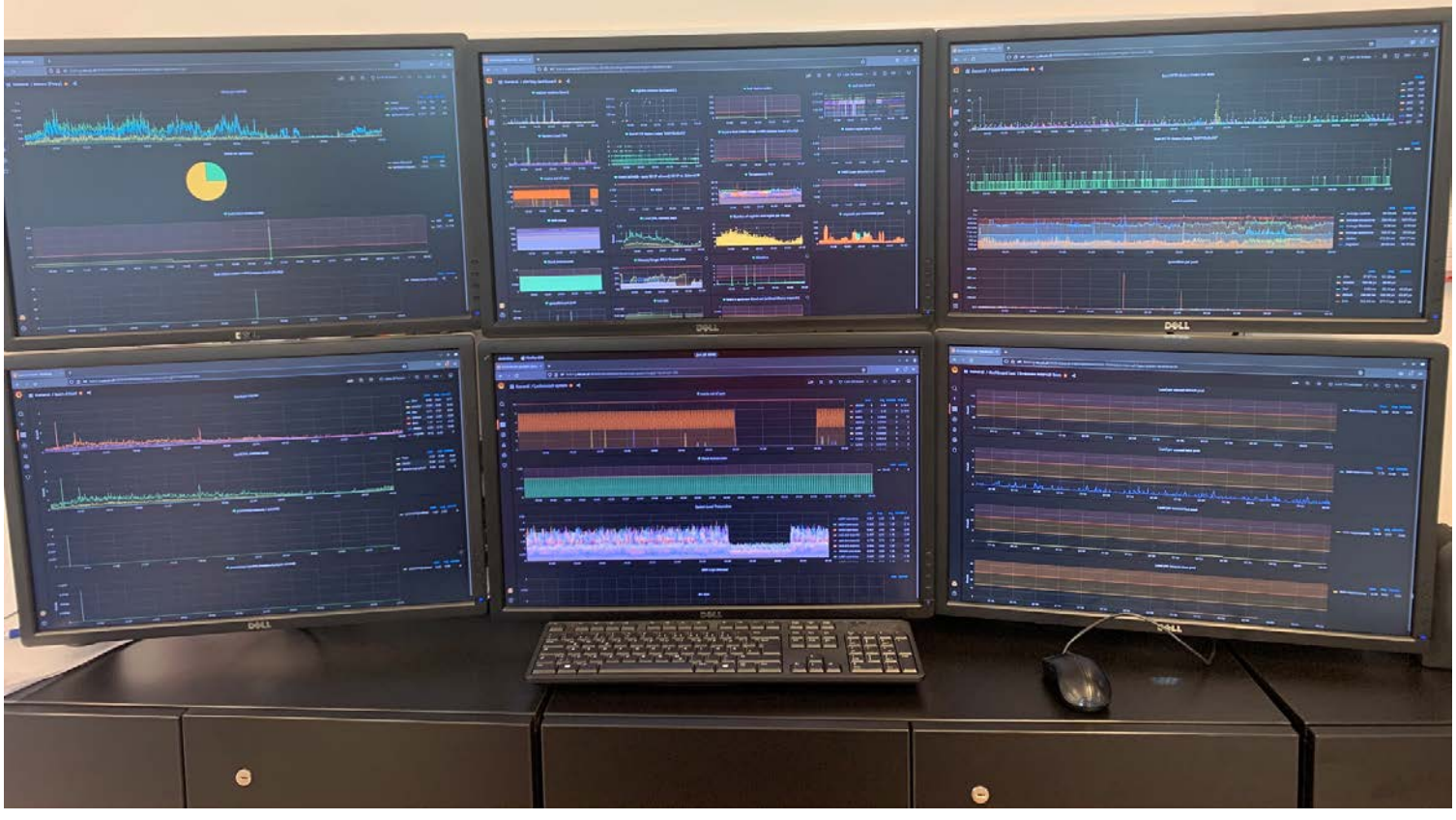
https://www.elastic.co/de/kibana/

Grafana

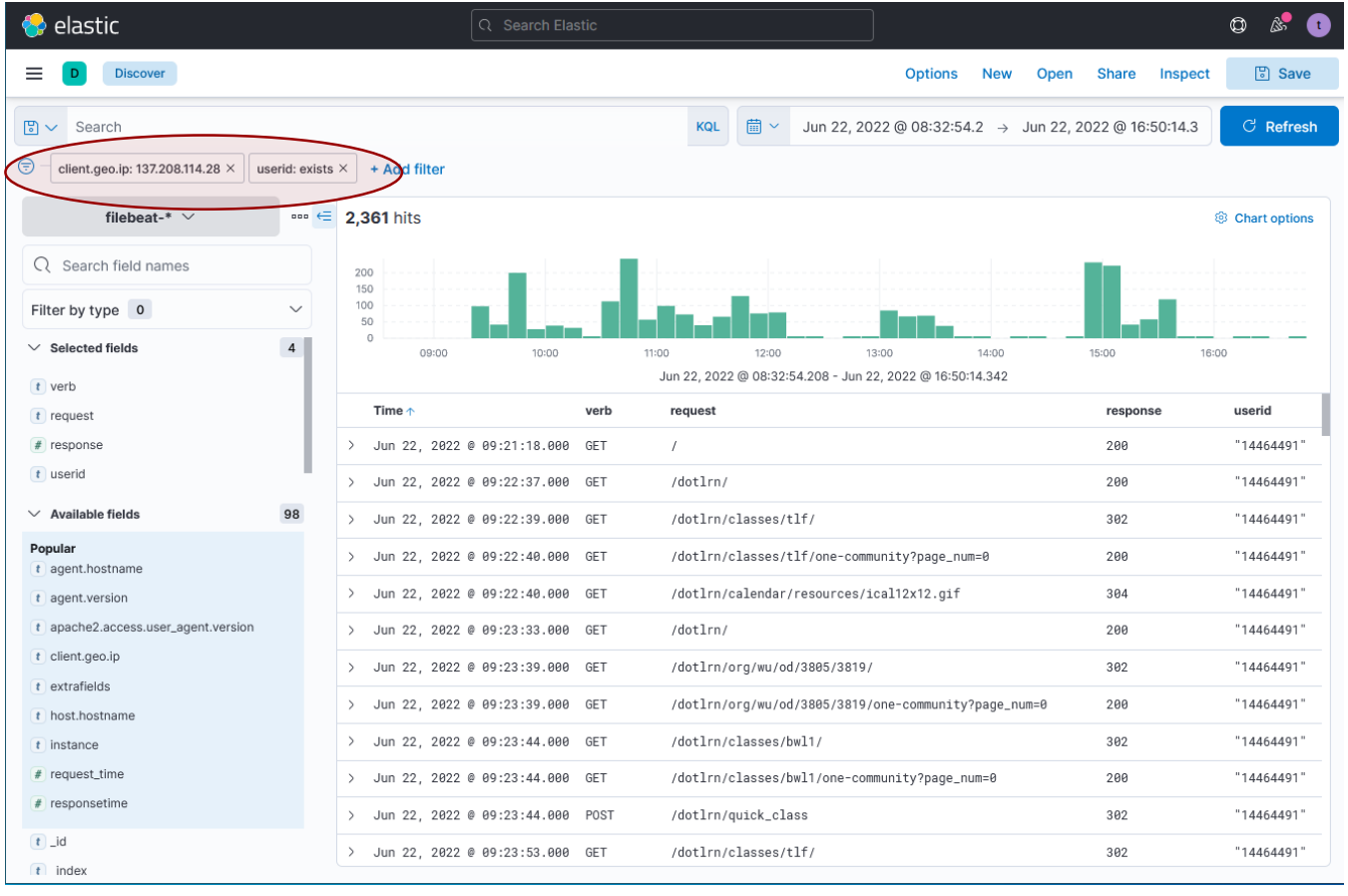
- Open-source platform for monitoring and observability
  - Querying, visualizing, alerting
  - Mixed various data sources: e.g. elastic search
  - Alerting:
    - Visualizing
    - Notifications (Email, Microsoft Teams,….)



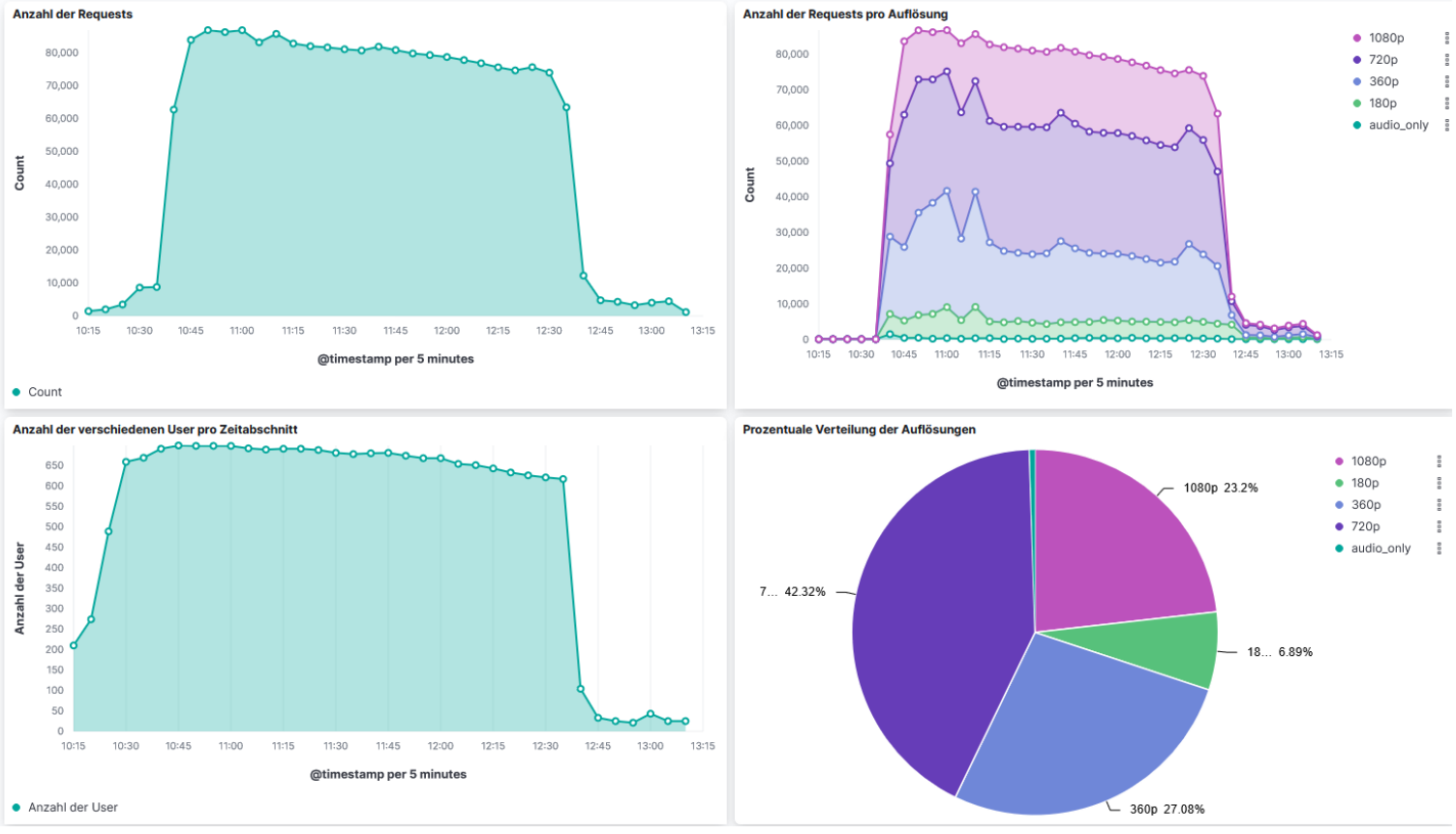
Monitoring screen in LEARN's office



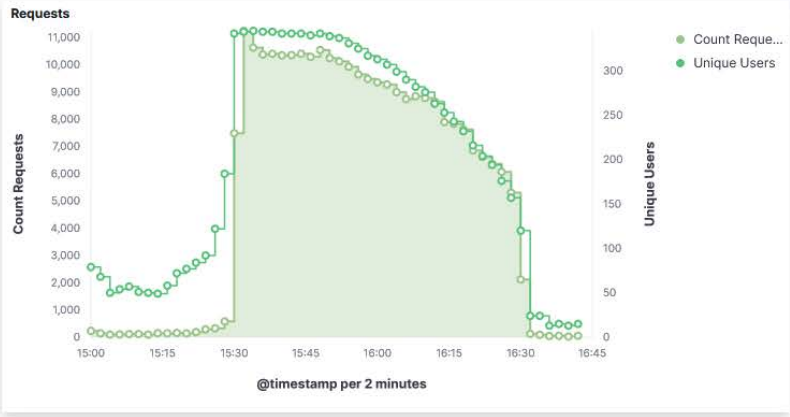
Kibana: Filtering POST and GET requests for IP address

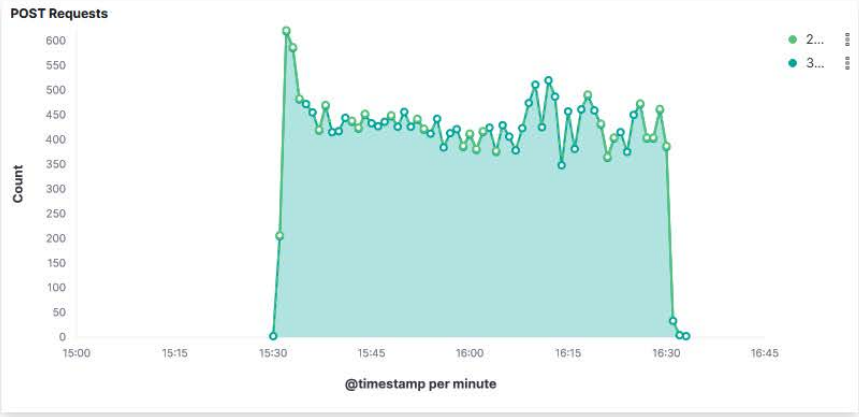


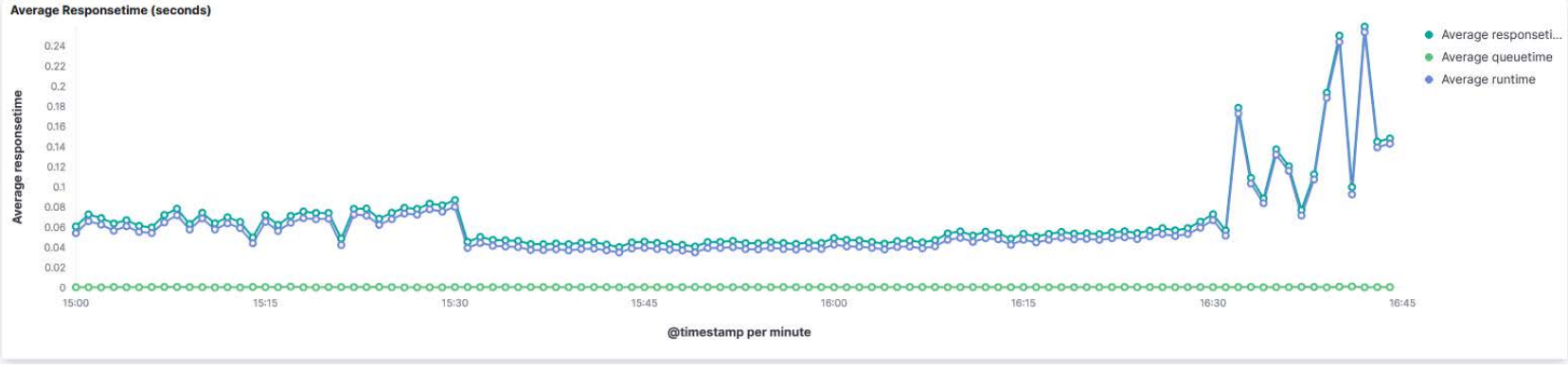
Kibana Dashboard: Web Streaming



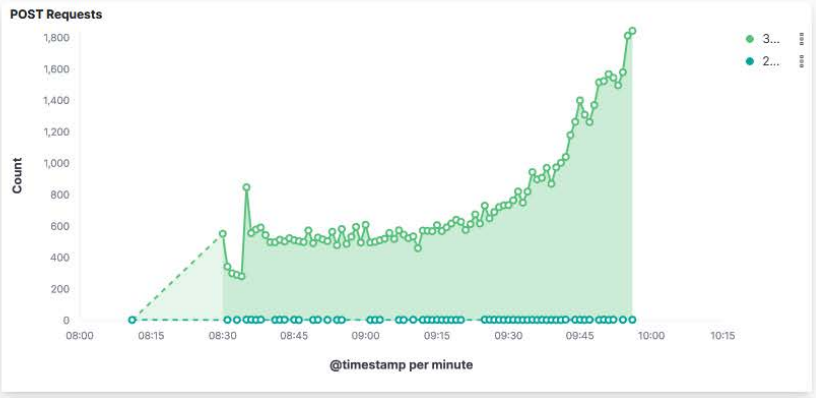
Kibana Dashboard: Exam 1

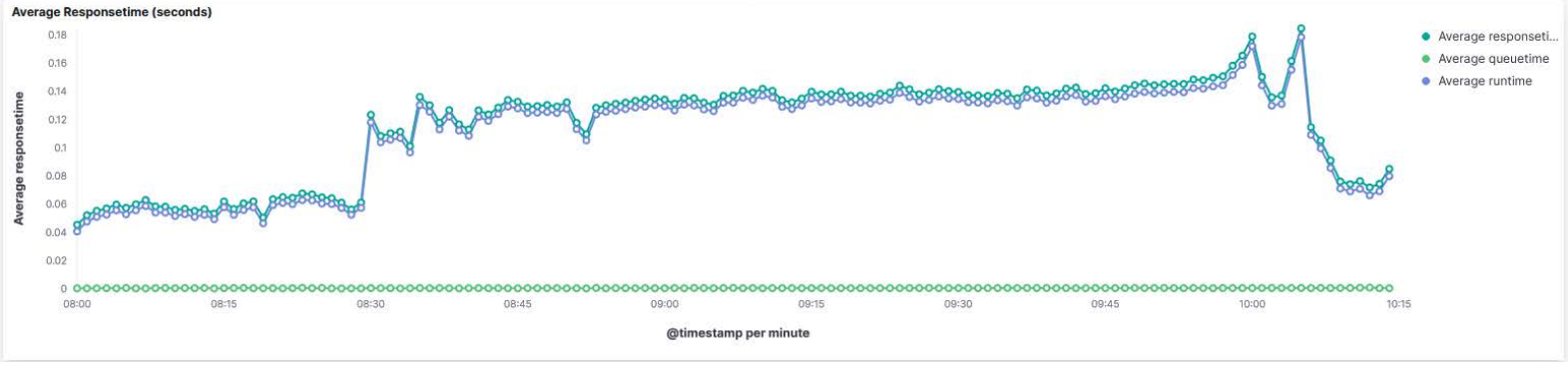






Kibana Dashboard: Exam 2





Live Demo

Questions?