IdP Cluster Strukturen und zentrales Logging

IT Center | RWTH AACHEN UNIVERSITY

Agenda

Agenda

- Kurzvorstellung RWTH Aachen
- Shibboleth Kennzahlen an der RWTH Aachen
- IdP Cluster Strukturen an der RWTH Aachen
- Aufbau IdP Cluster
- Kurzvorstellung Graylog
- Einbinden der Shibboleth Logs in Graylog

Kurzvorstellung RWTH Aachen

- Die RWTH Aachen ist eine technische Hochschule
- Studium
  - 44.517 Studierende im Wintersemester 2016/2017
  - 152 Studiengänge
- Personal
  - 540 Professorinnen und Professoren zum 31. Dezember 2016
  - 5.373 Wissenschaftliche Mitarbeiterinnen und Mitarbeiter – einschließlich Drittmittelpersonal
  - 2.679 Mitarbeiterinnen und Mitarbeiter in Technik und Verwaltung – einschließlich Drittmittelpersonal
  - 672 Auszubildende, Praktikantinnen und Praktikanten
- Einrichtungen
  - 9 Fakultäten
  - 260 Institute

Shibboleth Kennzahlen an der RWTH Aachen

IT Center | RWTH AACHEN UNIVERSITY

Kennzahlen an der RWTH Aachen

- ca. 48000 Studierende
- ca. 5000 Logins/Tag am Shibboleth System
- 126 lokal angebundene SPs

IdP Cluster Strukturen an der RWTH Aachen

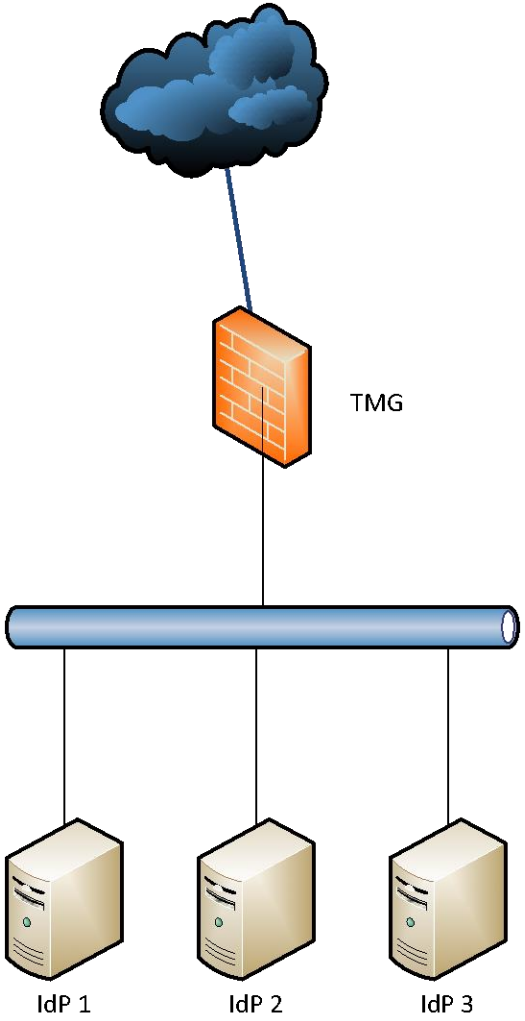
Theoretischer Aufbau

IdP Cluster Strukturen an der RWTH Aachen 1/3

- Produktiv: Hinter Forefront Thread Management Gateway (TMG)
- Test: Hinter BIG IP f5
- Jeweils 3 IdP Server
- Session Storage über MariaDB (Galera Cluster)
- Replay Cache memcached
- Artifact memcached

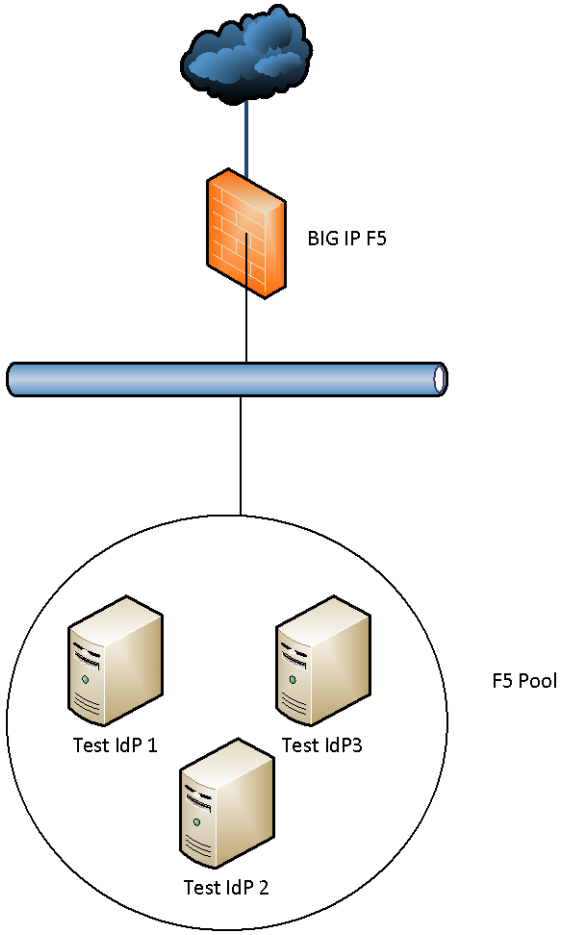
IdP Cluster Strukturen an der RWTH Aachen 2/3

- DNS: sso.rwth-aachen.de
- TMG Listener mit SSL Zertifikat für sso.rwth-aachen.de (IPv4 only)
- Verbindet mit einem der drei Server, legt auf Client Seite einen Cookie ab
- Alle drei Server sind in einem Cluster
- Metadaten über https://sso.rwth-aachen.de/metadata/rwth.metadata.xml abrufbar



IdP Cluster Strukturen an der RWTH Aachen 3/3

- DNS: sso-test.rwth-aachen.de
- Virtueller F5 Server, der SSL terminiert und vermittelt (IPv4 und IPv6)
- Verbindung zum Shibboleth-Pool
- Ein Server wird ausgesucht
- Alle drei Server laufen im Cluster
- Metadaten über https://sso-test.rwth-aachen.de/metadata/rwth.metadata.xml abrufbar



Aufbau IdP Cluster

Technische Umsetzung

Aufbau IdP Cluster 1/8

Verwendete Software

- Shibboleth Identity Provider 3.3.1
- Apache Tomcat 8
- openJDK 8
- MariaDB 10
- memcached
- apache2

Datenquellen:

- MS SQL Datenbank (Microsoft Identity Manager)
- Shibboleth LDAP

Aufbau IdP Cluster 2/8

memcached Cluster (Artifact und ReplayCache Storage)

- In der Datei `idp.properties`:
  - `idp.artifact.StorageService = shibboleth.MemcachedStorageService`
  - `idp.replayCache.StorageService = shibboleth.MemcachedStorageService`
- MemcachedStorageService muss zuerst in der `global.xml` konfiguriert werden:

```
<bean id="shibboleth.MemcachedStorageService"
    class="org.opensaml.storage.impl.memcached.MemcachedStorageService"
    c:timeout="2">
    <constructor-arg name="client">
        <bean class="net.spy.memcached.spring.MemcachedClientFactoryBean"
              p:servers="<SERVER1>:11211,<SERVER2>:11211"
              p:protocol="BINARY"
              p:locatorType="CONSISTENT"
              p:failureMode="Redistribute">
            <property name="hashAlg">
                <util:constant static-field="net.spy.memcached.DefaultHashAlgorithm.FNV1_64_HASH" />
            </property>
            <property name="transcoder">
                <!-- DO NOT MODIFY THIS PROPERTY -->
                <bean class="org.opensaml.storage.impl.memcached.StorageRecordTranscoder" />
            </property>
        </bean>
    </constructor-arg>
</bean>
```

Aufbau IdP Cluster 3/8

MariaDB Galera Cluster (User Consent und Session Storage)

- MariaDB aus den offiziellen Quellen installieren
- Auf dem initialen MariaDB Master
  - MariaDB starten
  - mysql_secure_installation ausführen
  - `server.cnf` anpassen

```
[galera]
# Mandatory settings
wsrep_on=ON
wsrep_provider=/usr/lib64/galera/libgalera_smm.so
wsrep_cluster_address='gcomm://'
wsrep_cluster_name='<Cluster Name>'
wsrep_node_address='<IP des Servers>'
wsrep_node_name='<Node Name>'
wsrep_sst_method=rsync
binlog_format=row
default_storage_engine=InnoDB
innodb_autoinc_lock_mode=2
bind-address=0.0.0.0
```

Aufbau IdP Cluster 4/8

- MariaDB stoppen
- Cluster mittels `galera_new_cluster` initialisieren
- Anschließend in der `server.cnf` die `wsrep_cluster_address` anpassen
  - `wsrep_cluster_address='gcomm://IdP2,IdP3`
- In der Firewall müssen die Ports 3306 (MySQL), 4567 (rsync Port), 4568 (IST Port) und 4444 TCP freigegeben sein
- Auf jedem Node gleiche Konfiguration wie Master
- Test, ob der Cluster läuft

```
mysql -u root -p -e "SHOW STATUS LIKE 'wsrep_cluster_size'"
```

Aufbau IdP Cluster 5/8

Shibboleth Consent in MariaDB schreiben

- MariaDB Connector nach `edit-webapp/WEB-INF/lib` speichern
  - Version 1.5.5: https://downloads.mariadb.com/Connectors/java/connector-java-1.5.5/mariadb-java-client-1.5.5-javadoc.jar
- `global.xml` um MySQL Beans erweitern

Aufbau IdP Cluster 6/8

```
<bean id="shibboleth.MySQLDataSource"
      class="%{mysql.class}"
      p:driverClassName="org.mariadb.jdbc.Driver"
      p:url="%{mysql.url}"
      p:username="%{mysql.username}"
      p:password="%{mysql.password}"
      p:maxWait="15000"
      p:validationQuery="select 1" />

<bean id="shibboleth.JPAStorageService"
      class="org.opensaml.storage.impl.JPAStorageService"
      p:cleanupInterval="%{idp.storage.cleanupInterval:PT10M}"
      c:factory-ref="shibboleth.JPAStorageService.EntityManagerFactory" />

<bean id="shibboleth.JPAStorageService.EntityManagerFactory"
      class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean">
    <property name="packagesToScan" value="org.opensaml.storage.impl"/>
    <property name="dataSource" ref="shibboleth.MySQLDataSource"/>
    <property name="jpaVendorAdapter" ref="shibboleth.JPAStorageService.JPAVendorAdapter"/>
    <property name="jpaDialect">
        <bean class="org.springframework.orm.jpa.vendor.HibernateJpaDialect" />
    </property>
</bean>

<bean id="shibboleth.JPAStorageService.JPAVendorAdapter"
      class="org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter"
      p:generateDdl="true"
      p:database="MYSQL"
      p:databasePlatform="org.hibernate.dialect.MySQL5Dialect" />
```

Aufbau IdP Cluster 7/8

- Konfiguration in der `idp.properties`:
  - `mysql.class = org.apache.tomcat.jdbc.pool.DataSource`
  - `mysql.url = jdbc:mariadb://localhost:3306/shibboleth`
  - `mysql.username = shibboleth`
  - `mysql.password = GeHEIM007`
- Der Wert `mysql.url` kann auch mehrere Server Adressen enthalten.
  - Beispiel für eine failover Konfiguration:
    `mysql.url = jdbc:mariadb:failover://server1:3306,server02:3306,server03:3306/shibboleth`

Aufbau IdP Cluster 8/8

Datenbank und Nutzer anlegen

```
mysql -uroot -p
 Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 19
Server version: 10.1.19-MariaDB MariaDB Server
Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]> CREATE DATABASE shibboleth;
MariaDB [(none)]> USE shibboleth;
MariaDB [shibboleth]> CREATE TABLE `StorageRecords` (
  `context` varchar(255) NOT NULL,
  `id` varchar(255) NOT NULL,
  `expires` bigint(20) DEFAULT NULL,
  `value` longtext NOT NULL,
  `version` bigint(20) NOT NULL,
  PRIMARY KEY (`context`,`id`)
);
MariaDB [shibboleth]> CREATE USER 'shibboleth'@'%' IDENTIFIED BY 'GeHEIM007';
MariaDB [shibboleth]> GRANT ALL PRIVILEGES ON shibboleth.* TO 'shibboleth'@'%';
```

Kurzvorstellung Graylog

Zentrales Logging

Kurzvorstellung Graylog

- Basiert auf dem ELK-Stack
- Nutzt anstelle von Kibana eine eigene Weboberfläche
- Integrierte Benutzerverwaltung
- Authentifizierung per LDAP, Shibboleth, lokal
- API
- Metriken lassen sich abbilden
- Zentraler Ort um Logs zu durchsuchen

Einbinden der Shibboleth Logs in Graylog

Wie kommen die Shibboleth Logs nach Graylog?

Einbinden der Shibboleth Logs in Graylog 1/3

- Elasticsearch arbeitet mit sogenannten Beats
- Filebeat und Journalbeat
- Installation von Filebeat über die offiziellen Elasticsearch Repositories
  - https://www.elastic.co/de/products/beats/filebeat
- Journalbeat muss manuell kompiliert werden und bringt alle journald Logs ins zentrale Logging
  - https://github.com/mheese/journalbeat

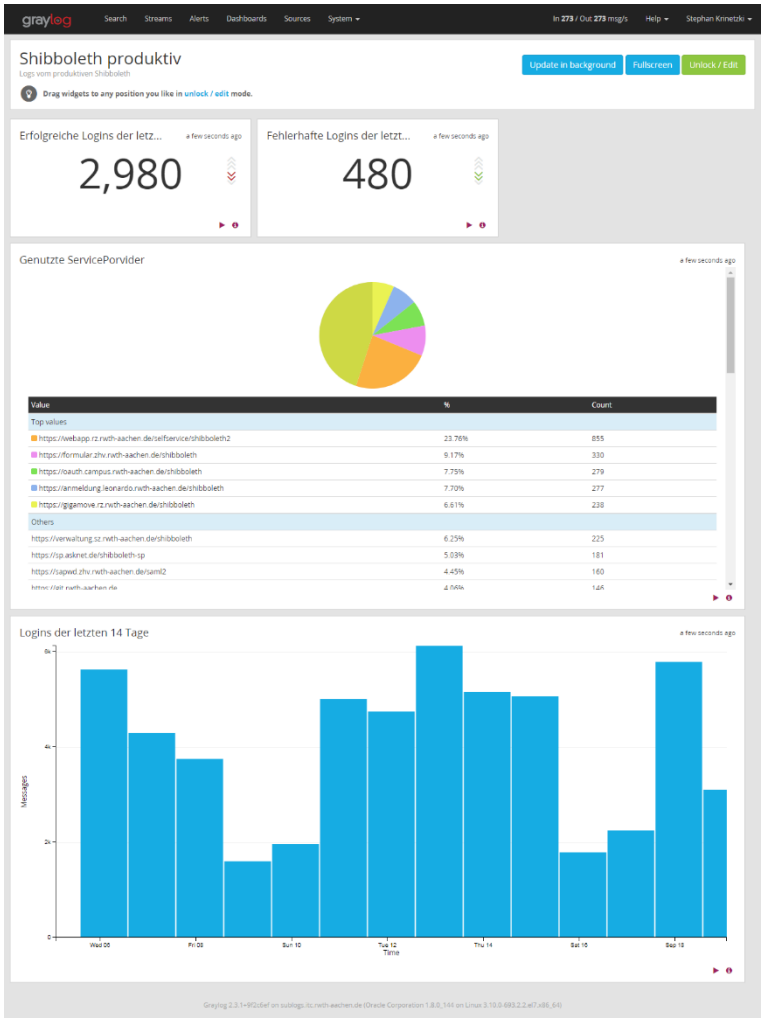
Einbinden der Shibboleth Logs in Graylog 2/3

Einrichten von Filebeat

- Konfiguriert wird `filebeat` über die `filebeat.yml` (meist `/etc/filebeat/filebeat.yml`)
- Beispiel Konfiguration

```
filebeat:
  prospectors:
    -
      paths:
        - /opt/shibboleth-idp/logs/idp-process.log
      input_type: log
      fields:
        document_type: idp-process
        sub_service: Shibboleth
        sub_group: IdM
      multiline.pattern: '^[[:space:]]+|^Caused by:'
      multiline.negate: false
      multiline.match: after
```

Einbinden der Shibboleth Logs in Graylog 3/3



Fragen?

Vielen Dank
für Ihre Aufmerksamkeit