Installation Broker Monitoring

Installation des Broker Monitoring für den AKTIN Broker

Alle Schritte dieser Anleitung werden als Root unter RedHat vorgenommen. Bei anderen Distributionen können die Befehle abweichen. Hier verwendete Benutzernamen und Passwörter, abseits von "Root" sind als Platzhalter zu betrachten und durch eigens erzeugte zu ersetzen. Auch sollte immer auf Versionsnummern in den Befelhen geachtet werden, diese können sich im Laufe der Zeit ändern (Z.B.: Neue Version von OpenJDK 8).

Schritt-für-Schritt-Anleitung

Alle Schritte dieser Anleitung werden als Root unter RedHat vorgenommen. Bei anderen Distributionen können die Befehle abweichen. Hier verwendete Benutzernamen und Passwörter, abseits von "Root" sind als Platzhalter zu betrachten und durch eigens erzeugte zu ersetzen. Auch sollte immer auf Versionsnummern in den Befelhen geachtet werden, diese können sich im Laufe der Zeit ändern (Z.B.: Neue Version von OpenJDK 8).

1. Installation von MySQL (Anleitung wird als root ausgeführt auf Red Hat 7):
    a. Das MySQL Repository muss installiert werden, sofern nciht bereits geschehen.

MySQL Repository installieren

```
yum localinstall https://dev.mysql.com/get/mysql80-community-release-el7-3.noarch.rpm
```

    b. Installation von MySQL:

Installation MySQL

```
yum install mysql-community-server
```

    c. MySQL Service starten und einschalten:

Service Config

```
systemctl start mysqld.service ## use restart after update

systemctl enable mysqld.service
```

    d. Root Passwort erzeugen (Im Output erscheint anschließend das erzeugte Passwort in der Konsole):

Zufalls-Root-Passwort erzeugen

```
grep 'A temporary password is generated for root@localhost' /var/log/mysqld.log |tail -1
```

    e. MySQL Secure Installation:
        i. **Change root password**
        ii. **Remove anonymous users**
        iii. **Disallow root login remotely**
        iv. **Remove test database and access to it**
        v. **Reload privilege tables**

Start Secure Installation

```
/usr/bin/mysql_secure_installation
```

Konfigurations-Dialog

```
Securing the MySQL server deployment.

Enter password for user root:

The existing password for the user account root has expired. Please set a new password.
```

```
New password:

Re-enter new password:

VALIDATE PASSWORD PLUGIN can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD plugin?

Press y|Y for Yes, any other key for No: y

There are three levels of password validation policy:

LOW    Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary
file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 0
Using existing password for root.

Estimated strength of the password: 100
Change the password for root ? ((Press y|Y for Yes, any other key for No) : y

New password:

Re-enter new password:

Estimated strength of the password: 50
Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for
No) : y
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.


Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.


Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
 - Dropping test database...
Success.

 - Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
```

f. Verbindung zur Datenbank:

Datenbankzugriff

```
mysql -u root -p

## OR ##
mysql -h localhost -u root -p
```

g. Anlegen einer Datenbank (Beispiel):
   i. DB_NAME = webdb
   ii. USER_NAME = webdb_user
   iii. REMOTE_IP = 10.0.15.25
   iv. PASSWORD = password123
   v. PERMISSIONS = ALL

Beispiel DB-Erzeugung

```
## CREATE DATABASE ##
mysql> CREATE DATABASE webdb;

## CREATE USER ##
mysql> CREATE USER 'webdb_user'@'10.0.15.25' IDENTIFIED BY 'password123';

## GRANT PERMISSIONS ##
mysql> GRANT ALL ON webdb.* TO 'webdb_user'@'10.0.15.25';

##  FLUSH PRIVILEGES, Tell the server to reload the grant tables  ##
mysql> FLUSH PRIVILEGES;
```

h. Firewall-Konfiguration:

Firewall-Ausnahme hinzufügen

```
firewall-cmd --permanent --zone=public --add-port=3306/tcp
```

Firewall neu starten

```
systemctl restart firewalld.service
```

i. Folgende Tabellen müssen in der angelegten Datenbank erzeugt werden:

i. **Tabelle: aktin_nodes**

```
CREATE TABLE `aktin_nodes` (
  `id_aktin_nodes` int(11) NOT NULL,
  `aktin_clientDN` varchar(255) DEFAULT NULL,
  `aktin_monitor_node` tinyint(1) NOT NULL,
  `aktin_last_start` varchar(255) DEFAULT NULL,
  `aktin_last_contact` varchar(255) DEFAULT NULL,
  `aktin_last_write` varchar(255) DEFAULT NULL,
  `aktin_last_reject` varchar(255) DEFAULT NULL,
  `aktin_imported` int(11) DEFAULT NULL,
  `aktin_updated` int(11) DEFAULT NULL,
  `aktin_invalid` int(11) DEFAULT NULL,
  `aktin_failed` int(11) DEFAULT NULL,
  `aktin_last_weekly` varchar(255) DEFAULT NULL,
  `aktin_last_monthly` varchar(255) DEFAULT NULL,
  `aktin_next_delay_reno` varchar(255) DEFAULT NULL,
  `aktin_next_error_reno` varchar(255) DEFAULT NULL,
  `aktin_next_no_import_reno` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id_aktin_nodes`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

ii. **Tabelle: aktin_email_contacts**

```
CREATE TABLE `aktin_email_contacts` (
  `id_aktin_email_contacts` int(11) NOT NULL AUTO_INCREMENT,
  `aktin_email_name` varchar(255) DEFAULT NULL,
  `aktin_email_adress` varchar(255) NOT NULL,
  `aktin_email_node_id` int(11) NOT NULL,
  `aktin_email_notify_on_error` tinyint(1) DEFAULT NULL,
  `aktin_email_notify_on_delay` tinyint(1) DEFAULT NULL,
  `aktin_email_weekly_report` tinyint(1) DEFAULT NULL,
  `aktin_email_monthly_report` tinyint(1) DEFAULT NULL,
  `aktin_email_notify_on_no_imports` tinyint(1) DEFAULT NULL,
  PRIMARY KEY (`id_aktin_email_contacts`),
  KEY `aktin_node_id_key_idx` (`aktin_email_node_id`),
  CONSTRAINT `node_id_key` FOREIGN KEY (`aktin_email_node_id`) REFERENCES `aktin_nodes`
(`id_aktin_nodes`) ON DELETE RESTRICT ON UPDATE RESTRICT
) ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT CHARSET=utf8
```

iii. **Tabelle: aktin_email_contacts**

```
CREATE TABLE `aktin_email_contacts` (
  `id_aktin_email_contacts` int(11) NOT NULL AUTO_INCREMENT,
  `aktin_email_name` varchar(255) DEFAULT NULL,
  `aktin_email_adress` varchar(255) NOT NULL,
  `aktin_email_node_id` int(11) NOT NULL,
  `aktin_email_notify_on_error` tinyint(1) DEFAULT NULL,
  `aktin_email_notify_on_delay` tinyint(1) DEFAULT NULL,
  `aktin_email_weekly_report` tinyint(1) DEFAULT NULL,
  `aktin_email_monthly_report` tinyint(1) DEFAULT NULL,
  `aktin_email_notify_on_no_imports` tinyint(1) DEFAULT NULL,
  PRIMARY KEY (`id_aktin_email_contacts`),
  KEY `aktin_node_id_key_idx` (`aktin_email_node_id`),
  CONSTRAINT `node_id_key` FOREIGN KEY (`aktin_email_node_id`) REFERENCES `aktin_nodes`
(`id_aktin_nodes`) ON DELETE RESTRICT ON UPDATE RESTRICT
) ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT CHARSET=utf8
```

2. Konfiguration der BrokerMonitoringApp (Konfiguration muss vor dem bauen mit Maven vorgenommen werden):
   a. Viele Parameter der App können über eine Properties-Datei konfiguriert werden ein aktuelles Beispiel der Datei:

Datei: application.properties

```
#Refreshrate of the monitoring thread on the server in milliseconds
broker.monitoring.refresh.rate = 1800000
#
#Settings for the access on the broker
#
#URL of broker host adress
broker.host=https://broker.aktin.org
#path on host to auth service
broker.auth.path=/auth/login
#path on host to node list
broker.nodes.path=/broker/node/
#path on host to individual stats of the nodes
broker.nodes.stats.id=/stats/
#path to information to different software versions of support software on the server of a node
broker.nodes.versions.id=/versions/
#login name to the broker
broker.user.name=something
#password to login name to he broker
broker.user.password=password
#
#The following settings are for the email client and depend on the used email server/service
#
#enables tls encryption in the email client
broker.mail.smtp.tls=true
#enables ssl encryption in the mail client
broker.mail.smtp.ssl=false
#enables the usage of a proxy connection
broker.mail.smtp.proxy=false
#The url to the email smtp host
broker.mail.smtp.host=smtp.uni-oldenburg.de
#The port of the email server/service
broker.mail.smtp.port=587
#host url of the proxy connection
broker.mail.proxy.host=host.de
#port of the used proxy connection
broker.mail.proxy.port=587
#username for the authentication in the used proxy connection
broker.mail.proxy.username=blank
#password for the authentication in the used proxy connection
broker.mail.proxy.password=blank
#username for the authentication on the email server/service
broker.mail.username=username
#password for the authentication on the email server/service
broker.mail.password=password
#sender email adress
broker.mail.from=it-support@aktin.org
#email adress for reply
broker.mail.reply.to.adress=it-support@aktin.org
#name of replicant
broker.mail.reply.to.name=AKTIN Support
#
#Names and Email adresses for notification within the broker monitoring
#
broker.mail.admin.name=Test Admin
broker.mail.admin.adress=test.admin@uol.de
broker.mail.projectlead.name=Test Projectlead
broker.mail.projectlead.adress=test.projectlead.uol.de
#
#Tags to parse the xml responses from the broker
#
broker.response.xml.tag.node.id=ns2:id
broker.response.xml.tag.node.clientDN=ns2:clientDN
broker.response.xml.tag.node.last-contact=ns2:last-contact
broker.response.xml.tag.node.start=start
broker.response.xml.tag.node.last-write=last-write
broker.response.xml.tag.node.last-reject=last-reject
broker.response.xml.tag.node.imported=imported
broker.response.xml.tag.node.updated=updated
broker.response.xml.tag.node.invalid=invalid
broker.response.xml.tag.node.failed=failed
#
```

```
#Settings for the monitoring process
#
#Maximum delay of node contact in hours
broker.time.acceptanceInterval=24
#Maximum delay of imports by node in hours value should not below "broker.time.acceptanceInterval"
broker.time.no.imports.acceptanceInterval=48
#Minimum amount on data insert events (import, update, invalid and failed) on the node to
calculate error rate
broker.statistics.evaluation.border=100
#The accepted error rate, values above will trigger notification
broker.statistics.evaluation.error.limit=0.01
#The time range for the error rate calculation
broker.statistics.evaluation.error.time.days=7
#Min number of invalid imports for the error rate calculation
broker.statistics.evaluation.error.min.invalid=3
#Defines the intervals to resend a notification
broker.notification.resend.in.days=7
broker.notification.error.resend.in.days=14
#
#The percentage to split any border interval to show a signal like a traffic light. Red will be
above border, green under the percentage and yellow between these values
#
broker.border.split.for.signal=0.5
#
#The hours to split any border interval to show a signal like a traffic light. Red will be above
border, green under the configured amount of hours and yellow between these values
#
broker.border.split.for.signal.last.contact.hours=2
#The hours to split any border interval to show a signal like a traffic light. Red will be above
border, green under the configured amount of hours and yellow between these values
#
broker.border.split.for.signal.last.import.hours=12
```

b. Das Benutzermanagement wird über eine Spring-Bibilothek abgewickelt, hier für müssen nur Umgebungsparameter und Benutzer einer Datei definiert werden. Ein Beispiel der Datei:

Datei: application-security.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans:beans
    xmlns="http://www.springframework.org/schema/security"
    xmlns:beans="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/security
    http://www.springframework.org/schema/security/spring-security.xsd">

    <http auto-config="true" use-expressions="true">
        <intercept-url pattern="/nodelist/**"
            access="hasAnyRole('ROLE_ADMIN','ROLE_USER')" />
        <intercept-url pattern="/"
            access="hasAnyRole('ROLE_ADMIN','ROLE_USER')" />
        <intercept-url pattern="/explanation"
            access="hasAnyRole('ROLE_ADMIN','ROLE_USER')" />
        <intercept-url pattern="/admin"
            access="hasRole('ROLE_ADMIN')" />
        <form-login login-page="/login"
            default-target-url="/nodelist" authentication-failure-url="/error"
            username-parameter="username" password-parameter="password" />
        <csrf />
        <logout logout-success-url="/logout" />
    </http>

    <authentication-manager>
        <authentication-provider>
            <!-- Before the password, there must be an encryption algorithm tag like
                {MD5}. Supported default tags are listed below, see DelegatingPasswordEncoder.
                Additional informations see : https://docs.spring.io/spring-security/site/docs/5.
0.5.RELEASE/reference/htmlsingle/#pe-dpe
                (10.3) and https://docs.spring.io/spring-security/site/docs/current/api/org
/springframework/security/crypto/factory/PasswordEncoderFactories.html -->
            <user-service>

                <!-- Current in SHA-256 encryption  username=user password=test123-->
                <user name="user"
                    password="{SHA-256}
ecd71870d1963316a97e3ac3408c9835ad8cf0f3c1bc703527c30265534f75ae"
                    authorities="ROLE_ADMIN" />
            </user-service>
            <password-encoder ref="encoder" />
        </authentication-provider>
    </authentication-manager>

    <!-- Creates a DelegatingPasswordEncoder with default mappings. Additional
        mappings may be added and the encoding will be updated to conform with best
        practices. However, due to the nature of DelegatingPasswordEncoder the updates
        should not impact users. The mappings current are:
        bcrypt - BCryptPasswordEncoder
        (Also used for encoding)
        ldap - LdapShaPasswordEncoder MD4 - Md4PasswordEncoder
        MD5 - new MessageDigestPasswordEncoder("MD5")
        noop - NoOpPasswordEncoder
        pbkdf2 - Pbkdf2PasswordEncoder
        scrypt - SCryptPasswordEncoder
        SHA-1 - new MessageDigestPasswordEncoder("SHA-1")
        SHA-256 - new MessageDigestPasswordEncoder("SHA-256")
        sha256 - StandardPasswordEncoder Returns: the PasswordEncoder to use -->

    <beans:bean id="encoder"
        class="org.springframework.security.crypto.factory.PasswordEncoderFactories"
        factory-method="createDelegatingPasswordEncoder">
    </beans:bean>

</beans:beans>
```

c. Die Datenbankverbindung muss in der Datei "servlet-context.xml" definiert werden. Der Anhang "?serverTimezone=UTC" an den Datenbanknamen ist nötig, falls die Zeitzone beim Datenbank server nicht auf UTC gesetzt ist (Was sie bei eienr standardinstallation anscheinend nciht ist). Ein Beispiel der Konfiguration:

Datenbankanbindung

```
<beans:bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource"
        destroy-method="close">
        <beans:property name="driverClassName" value="com.mysql.cj.jdbc.Driver" />
        <beans:property name="url"
            value="jdbc:mysql://localhost:3306/monitoringDB?serverTimezone=UTC" />
        <beans:property name="username" value="somthing" />
        <beans:property name="password" value="password1234" />
 </beans:bean>
```

Verwandte Artikel

- Ubuntu Speicher Erweiterung
- Ubuntu Distributions Upgrade
- Broker Anfrage
- HTTPS-Zugriff aktivieren
- Manuelle Datenabfrage