Linux

- Dovecot: expunge Trash und Junk folder
- Dovecot Diffie-Hellman
- Apache 2.4 / Tomcat 9 / Ubuntu 20.04 / mod_jk - Problem nach Update von 18.04
- Neue Seite
- Debian 11 - statische IPv4 Adresse
- MySQL, MariaDB

Dovecot: expunge Trash und Junk folder

doveadm -D search -u <user> mailbox INBOX.Junk SENTBEFORE 60d

doveadm -D expunge -u <user> mailbox INBOX.Junk SENTBEFORE 60d

Alle Mailboxen:

doveadm -D search -A mailbox INBOX.Junk SENTBEFORE 60d

doveadm -D expunge -A mailbox INBOX.Junk SENTBEFORE 60d

Dovecot Diffie-Hellman

Introduction

The default key size for the DH group in Dovecot is 2048. In some cases, users may wish to increase this size for security purposes. This article discusses how this can be adjusted through the terminal.

Procedure

To increase the DH group key size you will need to alter the dh.pem file that Dovecot uses to determine this. By default this is configured to be /etc/dovecot/dh.pem in the dovecot.conf configuration. You can use the following command to verify the current setting for this value.

```
doveconf -S |  grep '^ssl_dh'
```

To check the current size use the following command on the dh.pem returned from the above command. Since the default is 2048 in size, it would return output like "DH Parameters: (2048 bit)". This command can also be used to verify the change was successful.

```
openssl dhparam -in /etc/dovecot/dh.pem -text -noout
```

To update the length you will need to alter with the following command. You would replace the 2048 value with the desired length. Please note that this command may take a long time to complete and should be allowed to finish.

```
openssl dhparam 2048 > /etc/dovecot/dh.pem
```

That's it, now your Dovecot service will use the new DH group key size!

For additional information regarding this change and the Dovecot configurations, see the official documentation from the upstream providers below.

https://wiki.dovecot.org/SSL/DovecotConfiguration

https://support.cpanel.net/hc/en-us/articles/1500002760162-How-to-change-DH-group-key-size-in-Dovecot

Apache 2.4 / Tomcat 9 / Ubuntu 20.04 / mod_jk - Problem nach Update von 18.04

Two things:

1. The `redirectPort="8443"` attribute for the AJP Connector in Tomcat's `server.xml` is for SSL connections. Since I'm still working on the basic connection, I haven't enabled SSL yet, so this should be `redirectPort="0"`
2. Between the two versions of Tomcat that I'm using, the attribute `secretRequired` for the AJP Connector was changed from a default of `false` to a default of `true`. Since I wasn't sending a password with the proxy connection, it failed. Discovered this when I finally remembered to check the Tomcat logs, too (`/var/log/tomcat9/catalina.{date}.log` on Ubuntu).

The following AJP config works and allows me to load the webapp reverse-proxied through Apache:

```
    <Connector protocol="AJP/1.3"
               port="8009"
               redirectPort="8080"
               enableLookups="false"
               URIEncoding="UTF-8"
               secretRequired="false" />
```

https://serverfault.com/questions/1058480/why-is-apache-not-proxying-to-tomcat

Neue Seite

Edit `openssl.conf` file:

```
sudo nano /etc/ssl/openssl.cnf
```

Add this line at the top:

```
openssl_conf = openssl_init
```

And add these lines at the end:

```
[openssl_init]
ssl_conf = ssl_sect

[ssl_sect]
system_default = system_default_sect

[system_default_sect]
CipherString = DEFAULT@SECLEVEL=1
```

It works for me. :)

For any system add at the top of `openssl.cnf`:

```
openssl_conf = default_conf
```

and at end of `openssl.cnf`:

- For Debian add:

  ```
  [system_default_sect]
  MinProtocol = TLSv1.0
  CipherString = DEFAULT@SECLEVEL=2
  ```

- For Ubuntu 20.04 add:

```
[system_default_sect]
MinProtocol = TLSv1    #important !
CipherString = DEFAULT@SECLEVEL=2 # in my case works good with very old software
```

Debian 11 - statische IPv4 Adresse

/etc/network/interfaces

```
# The primary network interface
allow-hotplug enp6s18
iface enp6s18 inet dhcp
```

ändern zu:

```
auto enp6s18
iface enp6s18 inet static
        address 192.168.50.33
        netmask 255.255.255.0
        gateway 192.168.50.1
        dns-domain home.arpa
        dns-nameservers 192.168.50.1
```

MySQL, MariaDB

Wordpress: Sessions löschen

```
DELETE
FROM `wp_options`
WHERE option_name LIKE '_wp_session%'
```

Datenbank kopieren

```
mysql> CREATE DATABASE testdb_copy;
mysql> SHOW DATABASES;


mysqldump -u root -p testdb > D:\Database_backup\testdb.sql


mysql -u root -p testdb_copy < D:\Database_backup\testdb.sql


mysql> SHOW TABLES;
```

Zugriff von außen

Original: https://webdock.io/en/docs/how-guides/database-guides/how-enable-remote-access-your-mariadbmysql-database

You can do it by editing the MariaDB default configuration file. Look for "bind-address" directive in these two locations (**make the change in whichever file you find that directive**). You can open the file using your favorite text editor:

```
$ nano /etc/mysql/my.cnf
```

OR

```
$ sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

Change the value of the bind-address from **127.0.0.1** to **0.0.0.0** so that MariaDB server accepts connections on all host IPv4 interfaces.

```
bind-address = 0.0.0.0
```

Save and close the file when you are finished. Then, restart the MariaDB service to apply the changes:

```
$ sudo systemctl restart mariadb
```

Grant Access to a User from a Remote System

In this section, we will create a new database named wpdb and a user named wpuser, and grant access to the remote system to connect to the database wpdb as user wpuser.

First, log in to the MariaDB shell with the following command:

```
$ mysql -u admin -p
```

Provide your admin (root) password as shown in the Webdock backend and when you get the prompt create a database and user with the following command:

```
MariaDB [(none)]> CREATE DATABASE wpdb;
MariaDB [(none)]> CREATE USER 'wpuser'@'localhost' IDENTIFIED BY 'password';
```

Next, you will need to grant permissions to the remote system with IP address 208.117.84.50 to connect to the database named wpdb as user wpuser. You can do it with the following command:

```
MariaDB [(none)]> GRANT ALL ON wpdb.* to 'wpuser'@'208.117.84.50' IDENTIFIED BY 'password'
WITH GRANT OPTION;
```

Next, flush the privileges and exit from the MariaDB shell with the following command:

```
MariaDB [(none)]> FLUSH PRIVILEGES;
MariaDB [(none)]> EXIT;
```

```
grant all on *.* to 'username'@'%' identified by 'password' with grant option;
```