

Introduction -

Jabberd2 is **XMPP protocol based Instants Messaging (IM) server**. Jabberd2 is highly scalable,high performance jabber server. The beauty of the Jabberd2 architecture lies in the fact that its component architecture distributes services across **six components**, each of which communicates over TCP/IP.

1) Router - is the backbone of Jabber server. It accepts connections from Jabberd components and passes XML packets between components

2) Server to Server (S2S) - component handles communications with external servers. S2S passes packets between other components and external servers, and it performs dial-back to authenticate remote Jabber servers.

3) Resolver - acts in support of the S2S component. It resolves hostnames for S2S as part of dialback authentication.

4) Session Manager (SM) - component implements instant messaging features like message passing,presence,roster and subscription etc. + DB connection

5) Client to Server (C2S) - component handles communication with Jabber clients like connection,passing packets to SM, authenticate and register users.

6) Jabber core - logging and third party plugin communication.

** To compile/install **Jabberd-2.2.9** we need following packages on Debian/Ubuntu (similar on Redhat/Fedora or other OS)

- **libpam0g libpam0g-dev** (PAM support)
- **openssl libssl-dev** (TLS/SSL support)
- **libudns0 libudns-dev** (DNS Resolver Library)
- **libidn11 libidn11-dev libnet-libidn-perl** (libidn provides necessary string manipulation functionality for Jabberd2)
- **mysql-common libdbd-mysql-perl mysql-server-5.1 mysql-client-5.1 libmysqlclient16-dev** (MySQL DB authentication)

** **Jabberd2** supports **five authentication** (user) mechanism -

- * **PAM**
- * **MySQL Database**
- * **Berkeley DB**
- * **PostgreSQL Database**
- * **SQLite DB**
- * **OpenLDAP**

** Following ports are used by jabberd2 -

- * **port 5222** - non-SSL client connection
- * **port 5223** ? SSL client connection
- * **port 5269** ? server to server connection

* port 5347 ? jabberd2 router

Step 1] Create system User and Group for Jabberd 2 Server

(<http://codex.xiaoka.com/wiki/jabberd2:start>) ?

```
root@laptop:~# addgroup ?system jabber
```

```
Adding group `jabber' (GID 61) ...
```

```
Done.
```

```
root@laptop:~#
```

```
root@laptop:~# adduser ?system ?home /usr/local/jabberd-2.2.9/ ?shell /bin/false ?gid
61 jabber
```

* Verify system User and Group... (steps for Ubuntu)

```
root@laptop:~# id jabber
```

```
uid=125(jabber) gid=61(jabber) groups=61(jabber)
```

```
root@laptop:~#
```

Step 2] Download the latest version of Jabberd2 -

* Extract the source then compile/install it as shown below with PAM/MYSQL DB for authentication with SSL

```
root@laptop:/var/src/# wget -c
```

```
http://codex.xiaoka.com/pub/jabberd2/releases/jabberd-2.2.9.tar.bz2
```

```
root@laptop:/var/src# tar xvfj jabberd-2.2.9.tar.bz2
```

```
root@laptop:/var/src# cd jabberd-2.2.9
```

```
root@laptop:/var/src/jabberd-2.2.9# ./configure ?prefix=/usr/local/jabberd-2.2.9/
?enable-debug ?enable-mysql ?enable-ssl ?enable-pam ?enable-ssl
```

```
...
```

```
.....
```

```
checking for Libidn version >= 0.3.0... yes
```

```
checking for dns_init in -ludns... yes
```

```
checking gssasl.h usability... yes
```

```
checking gssasl.h presence... yes
```

```
checking for gssasl.h... yes
```

```
checking for gssasl_check_version in -lgsasl... yes
```

```
checking for GnuSASL version >= 0.2.27... no
```

```
configure: error: no SASL backend available out of: gssasl
```

```
root@laptop:/var/src/jabberd-2.2.9#
```

Step 3] Facing problem like "configure: error: no SASL backend available out of: gssasl"

~

Don't worry download latest version of gssasl library from URL ~

http://alpha.gnu.org/gnu/gsas1/

* Download latest version of **GNU SASL (gsasl)** ...

```
root@laptop:/var/src/# wget -c http://alpha.gnu.org/gnu/gsas1/gsas1-0.2.29.tar.gz
```

* Extract the source then compile/install it ...

```
root@laptop:/var/src# tar xvfz gsasl-0.2.29.tar.gz
root@laptop:/var/src# cd gsasl-0.2.29/
root@laptop:/var/src/gsas1-0.2.29# ./configure ?prefix=/usr/local/gsas1/
root@laptop:/var/src/gsas1-0.2.29# make
root@laptop:/var/src/gsas1-0.2.29# make install
root@laptop:/var/src/gsas1-0.2.29#
```

* **Verify the "gsasl" version**

```
root@laptop:/var/src/gsas1-0.2.29# /usr/local/gsas1/bin/gsas1 ?version
gsasl (GNU SASL) 0.2.29
Copyright (C) 2008 Simon Josefsson.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```

Written by Simon Josefsson.

```
root@laptop:/var/src/gsas1-0.2.29#
```

Step 4] Go back to Jabberd2 source and start compiling/installing as shown in below -

```
root@laptop:/var/src/jabberd-2.2.9# ./configure ?prefix=/usr/local/jabberd-2.2.9/
?enable-debug ?enable-mysql ?enable-ssl ?enable-pam ?enable-ssl
?with-extra-include-path=/usr/local/gsas1/include/
?with-extra-library-path=/usr/local/gsas1/lib/
root@laptop:/var/src/jabberd-2.2.9# make
root@laptop:/var/src/jabberd-2.2.9# make install
```

**** Create log and runtime directories ~**

```
root@laptop:/usr/local/jabberd-2.2.9# mkdir /usr/local/jabberd-2.2.9/var
root@laptop:/usr/local/jabberd-2.2.9# mkdir /usr/local/jabberd-2.2.9/var/run
```

```
root@laptop:/usr/local/jabberd-2.2.9# ls -l
total 20
drwxr-xr-x 2 jabber jabber 4096 2009-10-11 18:21 bin
drwxr-xr-x 3 jabber jabber 4096 2009-10-11 18:21 etc
drwxr-xr-x 3 jabber jabber 4096 2009-10-11 18:21 lib
drwxr-xr-x 3 jabber jabber 4096 2009-10-11 18:21 share
drwxr-xr-x 3 jabber jabber 4096 2009-10-11 18:42 var
root@laptop:/usr/local/jabberd-2.2.9#
```

Step 5] Configure jabberd-2.2.9 ~

* Setup (jabberid@laptop.ubuntu.me) Domain Name (hostname of server), IP address, port and log setting in client (c2s.xml) & server (sm.xml) configuration file -

NOTE ~ Domain Name not necessary to be hostname of server. But it should be resolvable (DNS) to one of the IP of server.

```
root@laptop:/usr/local/jabberd-2.2.9# hostname
laptop.ubuntu.me
root@laptop:/usr/local/jabberd-2.2.9#
```

```
root@laptop:/usr/local/jabberd-2.2.9# vi /usr/local/jabberd-2.2.9/etc/sm.xml
```

```
<pidfile>/usr/local/jabberd-2.2.9/var/run/sm.pid</pidfile>
```

```
<id>laptop.ubuntu.me</id>
```

```
<ip>0.0.0.0</ip>                <!? default: 127.0.0.1 ?>
```

```
<port>5347</port>                <!? default: 5347 ?>
```

```
<log type='file'>
```

```
<file>/usr/local/jabberd-2.2.9/var/log/sm.log</file>
```

```
root@laptop:/usr/local/jabberd-2.2.9# vi /usr/local/jabberd-2.2.9/etc/c2s.xml
```

```
<pidfile>/usr/local/jabberd-2.2.9/var/run/c2s.pid</pidfile>
```

** To auto enable registration (in c2s.xml file ?register-enable='true' is required)

```
<id register-enable='true'>laptop.ubuntu.me</id>
```

```
<ip>0.0.0.0</ip>
```

```
<port>5222</port>
```

```
<log type='file'>
```

```
<file>/usr/local/jabberd-2.2.9/var/log/c2s.log</file>
```

```
root@laptop:/usr/local/jabberd-2.2.9# vi /usr/local/jabberd-2.2.9/etc/s2s.xml
```

```
<pidfile>/usr/local/jabberd-2.2.9/var/run/s2s.pid</pidfile>
```

```
<log type='file'>
```

```
<file>/usr/local/jabberd-2.2.9/var/log/s2s.log</file>
```

Step 6] Configure Jabberd-2.2.9 for Storage and Authentication 9using MySQL DB) -

* Make sure that database "Jabberd2?" doesn't exist (if exist either drop db or change DB name in db-setup.mysql file). If not export MySQL DB dump from Jabberd2 source.....

```
root@laptop:/usr/local/jabberd-2.2.9# mysql -u root -p < /var/src/jabberd-2.2.9/tools/db-setup.mysql
```

Enter password:

```
root@laptop:/usr/local/jabberd-2.2.9# mysql -u root -p
```

Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 176

Server version: 5.1.31-lubuntu2 (Ubuntu)

Type ?help;' or ?\h' for help. Type ?\c' to clear the buffer.

```
mysql> show databases;
```

```
+?????????????--+
| Database          |
+?????????????--+
| information_schema |
| jabberd2          |
| mysql              |
+?????????????--+
3 rows in set (0.00 sec)
```

```
mysql> use jabberd2;
```

Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed

```
mysql> show tables;
```

```
+?????????+
| Tables_in_jabberd2 |
+?????????+
| active              |
| authreg              |
| disco-items         |
| logout              |
| motd-message        |
| motd-times          |
| privacy-default     |
| privacy-items       |
| private              |
| queue                |
| roster-groups       |
| roster-items        |
| status               |
| vacation-settings   |
| vcard                |
+?????????+
15 rows in set (0.00 sec)
```

```
mysql>
```

* Creating mysql user for jabberd2 ie ?jabberd2? with access to DB "jabberd2? -

```
mysql> GRANT select,insert,delete,update ON jabberd2.* to ?jabber'@'localhost'
IDENTIFIED by ?mypassword';
```

Query OK, 0 rows affected (0.00 sec)

```
mysql>quit
```

Bye

```
root@laptop:/usr/local/jabberd-2.2.9#
```

* Now verify access to DB ~

```
root@laptop:/usr/local/jabberd-2.2.9# mysql -u jabberd2 -p
```

Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 178

Server version: 5.1.31-lubuntu2 (Ubuntu)

Type ?help;' or ?\h' for help. Type ?\c' to clear the buffer.

```
mysql> show databases;
```

```
+????????+
```

```
| Database          |
```

```
+????????+
```

```
| information_schema |
```

```
| jabberd2          |
```

```
+????????+
```

2 rows in set (0.00 sec)

```
mysql> quit
```

Bye

```
root@laptop:/usr/local/jabberd-2.2.9#
```

Step 7] Change c2s.xml and sm.xml config file for MySQL DB support ?

```
root@laptop:/usr/local/jabberd-2.2.9# vi /usr/local/jabberd-2.2.9/etc/c2s.xml
```

```
<!? Authentication/registration database configuration ?>
```

```
<authreg>
```

```
<!? Backend module to use ?>
```

```
<module>mysql</module>
```

```
<!? MySQL module configuration ?>
```

```
<mysql>
```

```
<!? Database server host and port ?>
```

```
<host>localhost</host>
```

```
<port>3306</port>
```

```
<!? Database name ?>
<dbname>jabberd2</dbname>
<!? Database username and password ?>
<user>jabberd2</user>
<pass>mypassword</pass>
```

```
root@laptop:/usr/local/jabberd-2.2.9# vi /usr/local/jabberd-2.2.9/etc/sm.xml
```

```
<!? Storage database configuration ?>
<storage>
<!? Dynamic storage modules path ?>
<path>/usr/local/jabberd-2.2.9/lib/jabberd</path>

<!? By default, we use the SQLite driver for all storage ?>
<driver>mysql</driver>
```

```
<!? MySQL driver configuration ?>
<mysql>
<!? Database server host and port ?>
<host>localhost</host>
<port>3306</port>
```

```
<!? Database name ?>
<dbname>jabberd2</dbname>
<!? Database username and password ?>
<user>jabberd2</user>
<pass>mypassword</pass>
```

** To auto enable registration (in sm.xml file)

```
<auto-create/>
```

```
?????????????-
```

NOTE ~ It is not enough to add users to the 'authreg' table because this only introduces users to the c2s component, but not to the sm component. Correct entries are required in the 'active' table as well. It is best to use a Jabber client to register users.

Step 8] Let's start Jabberd-2 server (Test configuration) ?

```
root@laptop:~# su -l jabber -s /bin/bash -c "/usr/local/jabberd-2.2.9/bin/jabberd -b"
root@laptop:~#
```

* check whether ports are open or not

```
root@laptop:~# netstat -nlp
```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
-------	--------	--------	---------------	-----------------	-------

```

PID/Program name
tcp          0          0 0.0.0.0:5347          0.0.0.0:*          LISTEN
31662/router
tcp          0          0 0.0.0.0:5222          0.0.0.0:*          LISTEN
13883/c2s
tcp          0          0 127.0.0.1:3306         0.0.0.0:*          LISTEN
2892/mysqld
tcp          0          0 0.0.0.0:5269          0.0.0.0:*          LISTEN
13886/s2s
...
.....

```

root@laptop:~#

**** Now Register the user "jabberd@laptop.ubuntu.me" and password "secret" using Jabber IM client**

root@laptop:/usr/local/jabberd-2.2.9# tail -f var/log/c2s.log

```

...
Mon Oct 12 00:43:15 2009 [notice] [8] registration succeeded, requesting user creation:
jid=jabberd@laptop.ubuntu.me
Mon Oct 12 00:43:15 2009 [notice] [8] SASL authentication succeeded:
mechanism=DIGEST-MD5; authzid=jabberd@laptop.ubuntu.me
Mon Oct 12 00:43:15 2009 [notice] [8] bound: jid=jabberd@laptop.ubuntu.me/Telepathy
Mon Oct 12 00:44:20 2009 [notice] [9] [192.168.0.1, port=48307] connect

```

* Checking DB entry -

```

mysql> SELECT * FROM active;
+??????????+????????+?????+
| collection-owner          | object-sequence | time          |
+????????????+????????+?????+
| jabberd@laptop.ubuntu.me |                  1 | 1255288395   |
+????????????+????????+?????+
1 row in set (0.00 sec)

```

```

mysql> SELECT * FROM authreg;
+???-+????????+???-+
| username | realm          | password |
+???-+????????+???-+
| jabberd  | laptop.ubuntu.me | secret   |
+???-+????????+???-+
1 row in set (0.00 sec)

```

mysql>

=> Testing completed successfully....

Step 9] Configuring Jabberd2 for SSL/TLS Connections ?

Let's configure jabberd2 for SSL/TLS connection. Jabberd2 is designed to provide for SSL/TLS connections not only between Jabber clients and the server, but also between the Jabberd server components (sm, s2s and c2s) and the Jabberd router. A single SSL certificate may be used for these two functions (Jabber client to Jabberd and Jabberd component to router), or two separate keys may be used.

*** Generate Self signed SSL Certificate...**

```
root@laptop:/usr/local/jabberd-2.2.9# openssl req -new -x509 -newkey rsa:1024 -days 365
-keyout privkey.pem -out server.pem
Generating a 1024 bit RSA private key
.+++++
.....+++++
writing new private key to ?privkey.pem'
Enter PEM pass phrase:
....
Common Name (eg, YOUR name) []:laptop.ubuntu.me
root@laptop:/usr/local/jabberd-2.2.9#
```

*** Remove Passphrase from private key**

```
root@laptop:/usr/local/jabberd-2.2.9# openssl rsa -in privkey.pem -out privkey.pem
```

**** Combine the Private and Public Key and delete private key**

```
root@laptop:/usr/local/jabberd-2.2.9# cat privkey.pem >> server.pem
```

```
root@laptop:/usr/local/jabberd-2.2.9# rm privkey.pem
```

*** Change permission...**

```
root@laptop:/usr/local/jabberd-2.2.9# chown jabber:jabber
/usr/local/jabberd-2.2.9/server.pem
root@laptop:/usr/local/jabberd-2.2.9# ls -l /usr/local/jabberd-2.2.9
total 24
drwxr-xr-x 2 jabber jabber 4096 2009-10-11 22:17 bin
drwxr-xr-x 3 jabber jabber 4096 2009-10-12 01:03 etc
drwxr-xr-x 3 jabber jabber 4096 2009-10-11 20:16 lib
-rw-r?r? 1 jabber jabber 2217 2009-10-12 01:17 server.pem
drwxr-xr-x 3 jabber jabber 4096 2009-10-11 20:16 share
drwxr-xr-x 4 jabber jabber 4096 2009-10-12 00:20 var
root@laptop:/usr/local/jabberd-2.2.9#
```

```
root@laptop:/usr/local/jabberd-2.2.9# vi /usr/local/jabberd-2.2.9/etc/c2s.xml
<ssl-port>5223</ssl-port>
<pemfile>/usr/local/jabberd-2.2.9/server.pem</pemfile>
```

```
root@laptop:/usr/local/jabberd-2.2.9# vi /usr/local/jabberd-2.2.9/etc/s2s.xml
```

```
<pemfile>/usr/local/jabberd-2.2.9/server.pem</pemfile>
```

```
root@laptop:/usr/local/jabberd-2.2.9# vi /usr/local/jabberd-2.2.9/etc/sm.xml
```

```
<pemfile>/usr/local/jabberd-2.2.9/server.pem</pemfile>
```

```
<pemfile>/usr/local/jabberd-2.2.9/server.pem</pemfile>
```

**** Now restart the server and check log ...**

```
root@laptop:/usr/local/jabberd-2.2.9# tail -f var/log/c2s.log
```

```
Mon Oct 12 01:28:57 2009 [notice] connection to router established
Mon Oct 12 01:28:57 2009 [notice] [0.0.0.0, port=5222] listening for connections
Mon Oct 12 01:28:57 2009 [notice] [0.0.0.0, port=5223] listening for SSL connections
```

*** While registering user ~**

- 1) Required SSL/TLS
- 2) Force old SSL (5223 port)

Please enable above two setting and uncheck "Allow plaintext auth unencrypted streams"

NOTE ~ While login first time (auto registration mode) make sure to check "Create this new account on the server" checkbox in pidgin (bottom)

**** It works !!**

Step 10] Init.d startup script for Jabberd2 and Mu-Conference ?

```
root@laptop:~# /etc/init.d/jabberd2 start
Starting the Jabberd2 IM Server...
router 11095 | sm 11099 | s2s 11102 | c2s 11106 |mu-conf 11149
...
Done.
```

```
root@laptop:~# /etc/init.d/jabberd2 status
Jabberd2 IM Server status -
router ? 11095 | sm ? 11099 | s2s ? 11102 | c2s ? 11106 | mu-conf 11149
root@laptop:~#
```

*** Now check network setting...**

```
root@laptop:~# netstat -nlp
```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
PID/Program name					
tcp	0	0	0.0.0.0:5347	0.0.0.0:*	LISTEN
11095/router					

```

tcp          0      0 0.0.0.0:5222          0.0.0.0:*        LISTEN
11106/c2s
tcp          0      0 0.0.0.0:5223          0.0.0.0:*        LISTEN
11106/c2s
tcp          0      0 127.0.0.1:3306        0.0.0.0:*        LISTEN
2899/mysqld
tcp          0      0 0.0.0.0:5269          0.0.0.0:*        LISTEN
11102/s2s

```

```
root@laptop:~# /etc/init.d/jabberd2 stop
```

```
Stopping the Jabberd2 IM Server...
```

```
Done.
```

```
root@laptop:~#
```

```
** Want to see the script ~
```

```
root@laptop:~# cat /etc/init.d/jabberd2
```

```
#!/bin/bash
```

```
## Jabberd2 IM Server
```

```
## Jabber User/Group ? jabber/jabber
```

```
## command to start ~ su -l jabber -s /bin/bash -c "${BASE_PATH}/bin/jabberd -b"
```

```
##
```

```
#c2s
```

```
BASE_PATH="/usr/local/jabberd-2.2.9?"
```

```
c2s_pid="${BASE_PATH}/var/run/c2s.pid"
```

```
#s2s
```

```
s2s_pid="${BASE_PATH}/var/run/s2s.pid"
```

```
#sm
```

```
sm_pid="${BASE_PATH}/var/run/sm.pid"
```

```
#router
```

```
router_pid="${BASE_PATH}/var/run/router.pid"
```

```
#Mu-Conference
```

```
mu_conf_pid="${BASE_PATH}/var/run/mu-conference.pid"
```

```
case "$1?" in
```

```
start)
```

```
## checking whether Jabberd2 is running or not
```

```
if [ -f ${c2s_pid} ];then
```

```
c2spid=$(cat ${c2s_pid})
```

```
echo "Jabberd2 IM Server ~ ?c2s' is running (pid ${c2spid})"
```

```
elif [ -f ${s2s_pid} ];then
```

```
s2spid=$(cat ${s2s_pid})
```

```
echo "Jabberd2 IM Server ~ ?s2s' is running (pid ${s2spid})"
```

```
elif [ -f ${sm_pid} ];then
```

```
smpid=$(cat ${sm_pid})
```

```
echo "Jabberd2 IM Server ~ ?sm' is running (pid ${smpid})"
```

```
elif [ -f ${router_pid} ];then
```

```

routerpid=$(cat ${router_pid})
echo "Jabberd2 IM Server ~ ?router' is running (pid ${routerpid})"
else
echo "Starting the Jabberd2 IM Server..."
su -l jabber -s /bin/bash -c "${BASE_PATH}/bin/jabberd -b"
su -l jabber -s /bin/bash -c "${BASE_PATH}/bin/mu-conference -B -c
${BASE_PATH}/etc/mu-conference.xml" > /dev/null 2>&1
echo "router $(cat ${router_pid}) | sm $(cat ${sm_pid}) | s2s $(cat ${s2s_pid}) | c2s
$(cat ${c2s_pid}) |mu-conf $(cat ${mu_conf_pid})"
echo "..."
echo "Done."
fi
;;

stop)
echo "Stoping the Jabberd2 IM Server..."
if [ -f ${sm_pid} ];then
kill -9 $(cat ${sm_pid})
fi
if [ -f ${router_pid} ];then
kill -9 $(cat ${router_pid})
fi
if [ -f ${c2s_pid} ];then
kill -9 $(cat ${c2s_pid}) > /dev/null 2>&1
fi
if [ -f ${s2s_pid} ];then
kill -9 $(cat ${s2s_pid}) $(cat ${mu_conf_pid}) > /dev/null 2>&1
fi
##
killall -9 -u jabber
rm -f ${router_pid} ${sm_pid} ${s2s_pid} ${c2s_pid} ${mu_conf_pid} > /dev/null 2>&1
echo "Done."
;;

status)
echo "Jabberd2 IM Server status -"
if [ -f ${c2s_pid} ];then
c2spid=$(cat ${c2s_pid})
fi
if [ -f ${s2s_pid} ];then
s2spid=$(cat ${s2s_pid})
fi
if [ -f ${sm_pid} ];then
smpid=$(cat ${sm_pid})
fi
if [ -f ${router_pid} ];then
routerpid=$(cat ${router_pid})
fi
if [ -f ${router_pid} ];then

```

```
mupid=$(cat ${mu_conf_pid})
fi
echo "router ? ${routerpid} | sm ? ${smpid} | s2s ? ${s2spid} | c2s ? ${c2spid} |
mu-conf ${mupid}"
;;
*)
echo "Usage: $0 {start|stop|status}"
exit 1
esac
exit 0
#DONE
root@laptop:~#
```

Thank you,
Arun Bagul

Similar Posts:

- [????? ?????? ????? \(jabberd2\) ?? ?????? ? ??? ??????? ?????????? ?????? ??????????](#)
- [Multi User Conferencing \(MU-?????????????\) ??? Jabber \(jabberd2\)](#)
- [?? ??? ????? \(? ? ?????\) Jabber??? \(Jabberd2\)](#)
- [??????????MU?-???Jabber??????jabberd2\)](#)
- [?????? \(?-???Jabber\) \(Jabberd2\)](#)
- [????????? ??????? ?????????? \(?? ?????????\) ?? ????? \(Jabberd2\)](#)
- [??? ?????????? ????????????????? \(?????-?????????\) ????????? ?? ??? Jabberd2 \(\)](#)

Share and Enjoy:

-
-
-
-
-

-
-
-
-
-
-
-
-

This post is also available in language: [German](#), [Italian](#), [Arabic](#)