Openfire XMPP Server

- Install and Setup Openfire XMPP Server

Install and Setup Openfire XMPP Server

https://www.youtube.com/embed/pwIpDj9WwlI

Encrypted messaging has become a hot mess of one off platforms, and painful setups over the past 10 to 15 years. Before that, everyone, and I mean EVERYONE, used a standardized protocol that was awesome, and still is. XMPP was used by Microsoft, Yahoo, Excite, Google, Apple, and hundreds of other platforms for their chat communications. Then, for some reason they all started branching off, creating their own communication mechanisms and protocols, close-sourcing most of them, and segregating their users. We've had a few successes in bridging those chat gaps like WhatsApp (now owned by Facebook or Meta as they like to be called...so there went privacy), Telegram, Facebook Messenger (same story as WhatsApp), and on, and on. So, if you want to run a secure, encrypted server, you are really left to do it yourself. Every company that insists they'll stand by your privacy probably initially means it, but will eventually cave to the pressures of government, revenue, or who knows what else.

Today, we add to our collection of great chat options that are self-hostable, open source, and secure. Openfire is an XMPP server that provides a lot of control in the way you run it from the server side. It's not hard to get setup and running, and really does a terrific job of messaging.

What you'll need

- A machine or server to run Openfire on
- comfort with the CLI for a very simple install step.
- A MySQL/MariaDB Database (don't worry, we'll install and set it up in this article and the accompanying video)
- Either a local FQDN / Hostname for your server, or a publicly set FQDN / hostname for your VPS / Server with an A Record that points to it properly (public is optional)
- About 20 minutes of your time

Installation of Openfire

We will first download Openfire as a .deb (if you're using Arch, fedora, redhat, or other non-debian based distribution) simply look for the appropriate file in the downloads for OpenFire.

To pull this, we'll use wget to download it.

```
wget https://www.igniterealtime.org/downloadServlet?filename=openfire/openfire_4.7.4_all.deb -o openfire.deb
```

Once downloaded, we want to install it using the apt tool with the command:

```
sudo apt install ./openfire.deb -y
```

This will install the Openfire server, and start the service running inisitally. But, we want to make sure it runs each time the server is started, so let's set that up real quick with the command:

```
sudo systemctl enable --now openfire
```

Now, just to be safe, let's restart the service:

```
sudo systemctl restart openfire
```

and then check that's active with:

```
sudo systemctl status openfire
```

You should see the status as "active" here.

Installing MariaDB / MySQL

Next, we need to install a database for openfire to use. I'm going to install mariadb, but feel free to use mysql if you prefer to do so.

Luckily, MariaDB is in the debian/ ubuntu repositories already, so we can just run:

```
sudo apt install mariadb-server -y
```

Once installed, we need to secure the installation with the command:

```
mysql-secure-installation
```

You'll be prompted to enter the root password, but we haven't create a root password yet, so just press 'Enter' without typing anything in.

Next, you'll be asked if you want to convert to the 'unix socket' security model. Answer 'Y" for yes, and press enter.

Now enter a long, strong, complex password, then press enter. This is the password for your 'root' user. You'll be prompted to confirm the password as well, so make sure to store it in an encrypted password manager like Vaultwarden.

Now, answer each of the remaining questions with 'Y', then press enter. Once done, your MariaDB system is ready to use.

We need to do a few things in the MariaDB system to get it ready for Openfire. First, let's log into it. Use the command

```
mysql -u root -p
```

You'll be prompted for the root password you just created. Enter it, then press 'Enter'. You should now be logged into your MariaDB install.

You'll see the prompt with something like:

`| mariadb>` or `| mysql>`

Now we need to create the database and 'openfire' user, and a password for that user, as well as give that user privileges over the openfire database. We'll then run a script that was installed with the openfire server earlier, and it will do the rest of the work on the database for us.

1. Create the openfire database.

```
CREATE DATABASE openfire;
```

2. Give privileges to an new user called 'openfire', and assign a password (long and strong).

```
GRANT ALL PRIVILEGES ON openfire.* TO openfire@localhost IDENTIFIED BY 'longandstrongpasswordhere';
```

You really need to put this into a password manager or save it somewhere. You'll need it during the Openfire setup wizard.

3. Now let's clear and reset privileges on the database properly.

```
FLUSH PRIVILEGES;
```

4. Let's tell the system we want to use our new 'openfire' database

```
USE openfire;
```

5. Let's run the scripts

```
source /usr/share/openfire/resources/database/openfire_mysql.sql;
```

You should get a set of 'query ok' messages.

If so, you're ready to o to your new Openfire server in the web browser.

Run the Setup Wizard

Now, let's run through the Openfire setup wizard.

First, set your preferred language, then Continue. Next, You need to make sure that Openfire is setting the proper FQDN and XMPP Domain. You can also adjust the ports.

Next, ,select your preferred encryption, and give a strong encryption key.

Now, choose the Standard Database type. Leave the driver alone, but in the connection URL space, you'll want to change the "HOSTNAME" placeholder to the IP of your MySQL server, or use 127.0.0.1 if both servers are running on the same machine. Now, change the "DATABASE"

placeholder to "openfire". Leave everything else alone. Enter the user as 'openfire', and the password you created for the 'openfire' user during the GRANT PRIVILEGES step above. Once entered, click 'Continue'.

Choose the type of profile settings you want, if using LDAP, you need to have an LDAP server, and now the settings to use it.

Now enter an email for your admin user, and a long, strong password for your admin user to log into the web admin system with. Click 'Continue / Finish', and you are done.

Now click the 'Login' button, and enter 'admin' as the user with the admin password you just created. You'll now be logged into the admin dashboard.

Check out the video for the overview, setting up users, and connecting clients to your server.

Support My Content on Patreon

Support my content and ongoing efforts on patreon. https://www.patreon.com/awesomeopensource