Running the Wave Reference Server (WRS) on Debian/Ubuntu server

1 Prerequisites

- You need access to the Debian/Ubuntu server, and permisson to use sudo (default).
- If you have a new server, you can optionally install OpenNTPd for time synchronisation and enable unattended security updates:

```
wget -N http://admin-scripts.googlecode.com/svn/trunk/linux/ubuntu/new-server
cat new-server
sudo sh new-server
```

  - If you run a JeOS (Just enough OS) edition, you may consider to use the new-jeos script instead, which also installs OpenSSH, bash completion, man pages and nano.

2 Installation

- Get the WRS installation script:

```
wget -N http://admin-scripts.googlecode.com/svn/trunk/linux/ubuntu/install-wave
```

  - You may alternatively use the bit.ly shorcut:

```
wget -N http://bit.ly/install-wave
```

- Check the available options:

```
sh install-wave --help
```

- Install Sun Java 6, Openfire XMPP server and Wave protocol:

```
sudo sh install-wave
```

  - You can use any available options to override the default settings, or remove the *sudo* command to install wave for current user only. The installation script will still need sudo for a few commands though.
  - For unattended installation you need to accept the Sun Java License via the *--accept-sun-java-license* option.
  - Detailed installation protocols are created in the cache directory, see install-wave options for the specific location.
  - See also:
    * http://jamespurser.com.au/blog/Wave_Reference_Server_-_A_Startup_Guide
    * http://code.google.com/p/wave-protocol/wiki/Installation

    * http://www.onthetopofthewave.com/2009/09/wave-federation-prototype-server/
    * http://groups.google.com/group/wave-protocol

- Configure Openfire server according to the instructions provided by the script.

3 Running

- Run the wave server and the echoy agent:

```
wave-server >&2 2>> ~/wave-server.log &
wave-agent-echoey >&2 2>> ~/wave-agent-echoey.log &
```

  – Using *sudo* to run them under root is neither needed, nor recommended.

- Optionally use multitail for WRS and Openfire logspotting:

```
sudo multitail -M 0 ~/wave-server.log --mergeall \
    -ci red /var/log/openfire/error.log \
    -ci yellow /var/log/openfire/warn.log \
    -ci green /var/log/openfire/info.log \
    -ci blue /var/log/openfire/debug.log
```

  – Sudo is needed for accessing */var/log/openfire*.
  – You may need to install multitail first:

```
sudo aptitude install multitail
sudo sed -i 's/^\(check_mail:\).*/\10/' /etc/multitail.conf    ### disable mail check
```

  – You may optionally prepend *screen* before the command to run it inside its own screen, then spawn a working copy of bash (ctrl+a, c) and switch between them (ctrl+a, ctrl+a). There's also another advantage of using screen: if your ssh conection drops, the programs started under screen keep running and you can reconnect to them in new session with:

```
screen -DR
```

- Run your wave client:

```
wave-client-console $USER
```

- List available commands, start a new wave, open it, invite echoey, exchange greets and leave:

```
/
/new
/open 0
/add echoey@type-your-domain-here
Hi!
^D
```

  – See also:
    * http://code.google.com/p/wave-protocol/wiki/ConsoleClient

- Check the wave server and the echoy agent processes, and quit them via TERM signal:

```
ps -FC java|grep fedone[-]

kill `ps -FC java|grep fedone[-]|awk '{print $2}'`
```

  – Bum, your waves are gonna now, they were only kept in memory.
  – I may update install-wave script to allow running server and agents as deamons when suitable with later WRS releases.

4 Limitations

- Current WRS version (0.2, as of 2009/09/25) limitations:
  - No wave persistence between server restarts (memory only, no RDBMS backend)
  - No robots or embedding API
- Wave installation script todo:
  - Other configuration options, like ports (as needed)
  - Install WRS daemon running under dedicated user (when needed)
  - Detect if source was actually updated, and rebuild WRS + restart daemon in this case (if unattended/scheduled updates are needed) (does not make much sense until waves become persistent between WRS restarts)
  - Read all default settings from an existing installation, not only XMPP secret key
  - Pre-configure Openfire (would be only good for fully automatic installations)
  - Ejabberd support (feel free to implement it if you need it)

5 Updating

- Feel free to run the install-wave script anytime to update the WRS.
- Currently you need to specify all your custom options again, except for the XMPP secret key, which is automatically read from an existing installation.
- Currently you also need to manually check the wave-protocol.log to see if there were any WRS updates and restart the server in that case.

6 Federation

- You need to have an A and a SRV DNS records for your domain (server host FQDN) and also for a new wave subdomain of it (*wave.* + server host FQDN).
- You can use the dig command to check if the records are set to the server IP:

```
sudo aptitude install dnsutils   # install dig command from the dnsutils package

dig +short -t A ‘hostname --fqdn‘
dig +short -t A wave.‘hostname --fqdn‘

dig +short -t SRV _xmpp-server._tcp.‘hostname --fqdn‘
dig +short -t SRV _xmpp-server._tcp.wave.‘hostname --fqdn‘
```

- If your server is behind NAT, you also need to forward port 5269 to the server.
- See also:
  - http://code.google.com/p/wave-protocol/wiki/Federation
  - http://jamespurser.com.au/blog/Federating_Your_Wave_Server