Universal Webserver

Technical Manual of the `Universal Webserver`

Organisation: Copyright (C) 2019-2020 Olivier Boudeville

Contact: about (dash) universal-webserver (at) esperide (dot) com

Creation date: Saturday, May 2, 2020

Lastly updated: Sunday, May 24, 2020

Status: Work in progress

Version: 0.0.2

Dedication: Users and maintainers of the `Universal Webserver`.

Abstract: The Universal Webserver, part of the Universal Server umbrella project, provides a multi-domain, multi-virtualhost webserver integrating various web-related services.

We present here a short overview of these services, to introduce them to newcomers.

The next level of information is to read the corresponding source files, which are intensely commented and generally straightforward.

Table of Contents

Overview

We present here a short overview of the services offered by the *Universal Webserver*, to introduce them to newcomers.

The goal is to provide an integrated framework in order:

- to better operate websites based on virtual hosting, so that a networked computer can serve as many websites corresponding to as many domains as wanted; this involved reading and interpreting vhost and other configuration information, handling properly 404 errors, producing access logs that are adequate for web analytics, rotating all logs, etc.
- to link to the Universal Server optionally (i.e. if available, knowing both should be able to run in the absence of the other), in order to offer a web front-end for it

The next level of information about US-Web is to read the corresponding source files, which are intensely commented and generally straightforward.

Layer Stack

From the highest level to the lowest, as summarised here, a software stack involving the Universal Webserver usually comprises:

- the *Universal Webserver* services themselves (i.e. this US-Web layer)
- Cowboy (for a small, fast and modern web framework)
- [optional] Awstats (for the analysis of access log files)
- US-Common (for US base facilities)
- Ceylan-Traces (for advanced runtime traces)
- Ceylan-WOOPER (for OOP)
- Ceylan-Myriad (as an Erlang toolbox)
- Erlang (for the compiler and runtime)
- GNU/Linux

The shorthand for `Universal Webserver` is `uw`.

Recommendations

In terms of security, we would advise:

- to stick to the **latest stable version** of all software involved (including US-Web and all its stack, Erlang, and the operating system itself)
- to ensure that a **firewall** blocks everything from the Internet by default, including the EPMD port(s) (i.e. both the default Erlang one and any non-standard one specified through the `epmd_port` key defined in US-Common's us.config); one may get inspiration from our iptables.rules-Gateway.sh script for that
- still in `us.config`, to set:
  - a strong-enough Erlang **cookie**: set the `vm_cookie` key to a well-chosen value, possibly a random one deriving from an output of `uuidgen`
  - possibly a **limited TCP port range** (see the `tcp_port_range` key)
  - the **execution context** to `production` (see the `execution_context` key)
- to rely on dedicated, **low-privileged users and groups** (US-Web relies on `authbind`; refer to our start-us-web.sh script for that)
- to apply a streamlined, reproducible **deployment process**, preferably based on our deploy-us-web-release.sh script
- to use also the counterpart stop-us-web.sh script and to have them triggered through `systemd`; we provide a corresponding us-web.service **unit file** for that, typically to be placed in `/etc/systemd/system` and whose `ExecStart/ExecStop` paths shall preferably be symlinks pointing to the latest deployed US-Web release (ex: `/opt/universal-server/us_web-latest`)
- to **monitor regularly** both:
  - the US-Web server itself (see our monitor-us-web.sh script for that, relying on the trace supervisor provided by the Ceylan-Traces layer)
  - the remote accesses made to the websites (typically by enabling the US-Web "meta" feature, generating and updating automatically a dedicated website displaying in one page all hosted websites and linking to their web analysis report; refer to the `log_analysis` key of the US-Web configuration file (ex: see us-web-for-tests.config as an example thereof)

Licence

The `Universal Webserver` is licensed by its author (Olivier Boudeville) under the GNU Affero General Public License as published by the Free Software Foundation, either version 3 of this license, or (at your option) any later version.

This allows the use of the Universal Webserver code in a wide a variety of software projects, while still maintaining copyleft on this code, ensuring improvements are shared.

We hope indeed that enhancements will be back-contributed (ex: thanks to merge requests), so that everyone will be able to benefit from them.

Current Stable Version & Download

As mentioned, the single, direct prerequisites of the Universal Webserver are:

- Cowboy
- Awstats (as an optional, runtime-only dependency)
- US-Common

The latter relies on Ceylan-Traces, which implies in turn Ceylan-WOOPER, then Ceylan-Myriad and Erlang.

We prefer using GNU/Linux, sticking to the latest stable release of Erlang, and building it from sources, thanks to GNU `make`.

We recommend indeed obtaining Erlang thanks to a manual installation (refer to the corresponding Myriad prerequisite section for more precise guidelines), Awstats thanks to your distribution of choice (ex for Arch Linux: `pacman -S awstats`) and, for all Erlang-related software, to rely on rebar3.

One wanting to be able to operate on the source code of these dependencies may define appropriate symbolic links in a `_checkouts` directory created at the root one's `us-web` clone, these links pointing to relevant GIT repositories.

Using Cutting-Edge GIT

This is the installation method that we use and recommend; the Universal Webserver `master` branch is meant to stick to the latest stable version: we try to ensure that this main line always stays functional (sorry for the pun). Evolutions are to take place in feature branches and to be merged only when ready.

Once Erlang and possibly Awstats are available, it should be just a matter of executing our get-us-web-from-sources.sh script for downloading and building all dependencies at once, and run a test server (use its `--help` option for more information.

For example:

```
$ cd /tmp
$ wget https://raw.githubusercontent.com/Olivier-Boudeville/us-web/master/p
$ sh ./get-us-web-from-sources.sh --checkout
Switching to checkout mode.

 Installing US-Web in /tmp...

 Cloning into 'us_web'...
 [...]
 ===> Compiling us_web
 Starting the us_web release (EPMD port: 4506):
 [...]
 US-Web launched, please point a browser to http://localhost:8080 to check

$ firefox http://localhost:8080 &
```

One shall then see a text-only page such as:

```
This is static website D. This is the one you should see if pointing
to the default virtual host corresponding to the local host. This
shows that the US-Web server is up and running.
```

Understanding the role of the main US configuration file and of the corresponding US-Web configuration file for this test should be fairly straightforward.

Based on that, devising one's version of them should allow to have one's US-Web server running at the cost of very little efforts.

Using OTP-Related Build/Runtime Conventions

As discussed in these sections of Myriad, WOOPER, Traces and US-Common, we added the (optional) possibility of generating a Universal Webserver *OTP application* out of the build tree, ready to result directly in an *(OTP) release*. For that we rely on rebar3, relx and (possibly) hex.

Then we benefit from a standalone, complete Universal Webserver able to host as many virtual hosts on any number of domains as needed.

As for Myriad, WOOPER, Traces and US-Common, most versions of the Universal Webserver are also published as Hex packages.

For more details, one may have a look at rebar.config.template, the general rebar configuration file used when generating the Universal Webserver OTP application and release (implying the automatic management of all its dependencies).

Support

Bugs, questions, remarks, patches, requests for enhancements, etc. are to be reported to the project interface (typically issues) or directly at the email address mentioned at the beginning of this document.

Please React!

If you have information more detailed or more recent than those presented in this document, if you noticed errors, neglects or points insufficiently discussed, drop us a line! (for that, follow the Support guidelines).

Ending Word

Have fun with the Universal Webserver!

Universal Webserver