OpenCPU Installation and Administration Manual for Ubuntu 12.04 (LTS)

Version 0.70-0

Jeroen Ooms

September 21, 2012

About

This document describes installation and administration of the `OpenCPU` server. `OpenCPU` runs on `Ubuntu Linux 12.04 (LTS)`, either on a `i386` (32 bit) or `x64` architecture. `Ubuntu Server` edition is recommended, but all versions of `Ubuntu 12.04`, including `Ubuntu Server`, `Ubuntu Desktop`, `Kubuntu`, `EDUbuntu`, etc are supported. `Ubuntu 12.04` is a Long Term Support edition (LTS), and the official nickname of this version of `Ubuntu` is *Precise Pangolin*, or for short: *Precise*. Version 12.04 refers to the month in which it was released: April 2012.

The installation described in this document should work out of the box on a clean version of `Ubuntu` 12.04. We highly recommend taking advantage of modern technology by using some sort of virtual server or cloud environment to host `OpenCPU`. Virtual servers are easy to install, delete or clone, and furthermore most cloud environments offer convenient preinstalled version of `Ubuntu Linux`. This way installation of `OpenCPU` is easy and can be done in a couple of minutes. If you have no idea where to start: Amazon EC2 is a nice and affordable pay-by-hour hosting service which includes great preinstalled versions of `Ubuntu Linux`.

Note that `OpenCPU` is Open Source software and does not come with any guarantees or warranties. The project is still in an early stage and most likely the software will contain bugs. Please post feedback, problems, comments, suggestions, etc, of any kind on github:

`https://github.com/jeroenooms/opencpu/issues`

Thank you for giving it a try. More information on the project can be found on the OpenCPU website: `http://opencpu.org`.

1 Installation

The following instructions will deploy a server with:

- Ubuntu 12.04
- OpenCPU 0.70
- R 2.15

The 12.04 version of Ubuntu ships with the following versions of third party dependencies:

- Linux kernel 3.2.0
- Apache 2.2.22 (includes mod-ssl)
- Apparmor 2.7.102
- Nginx 1.1.19

1.1 Packages

This section will show how to install OpenCPU on Ubuntu 12.04 (LTS). The OpenCPU software has been wrapped up in a number of packages. These include:

- opencpu-server – OpenCPU API server.
- opencpu-cache – load balancing / caching server (**optional**)
- opencpu-cran – installs all CRAN packages (**optional**)

Only installation of the opencpu-server package is required. The opencpu-cache and opencpu-cran packages contain optional additional functionality. When installing either of these packages, all dependencies will automatically be installed as well.

1.2 Installing Ubuntu Linux

The current version of OpenCPU requires an Ubuntu 12.04 system, either on a i386 (32 bit) and x64 architecture. Ubuntu Server edition is recommended, but all versions of Ubuntu 12.04, including Ubuntu Server, Ubuntu Desktop, KUbuntu, EDUbuntu, etc are supported. The preferred way of running OpenCPU is on a clean Ubuntu Server edition. A free copy of the Ubuntu Server installation disc can be obtained from the Ubuntu download page:

http://www.ubuntu.com/download/server/download

If you would like to run OpenCPU on an Amazon EC2 server, the best way is to use one of the official AMI's as provided by the Ubuntu team:

http://cloud-images.ubuntu.com/precise/current/

Another possibility is to install a Ubuntu on a virtual server inside another OS. For example, on Windows and Linux the free VMware Player or VMware vSphere Hypervisor (ESXi) can be used to run one or more virtual servers, and on Mac OSX, one can use Parallels. This way you can install Ubuntu and OpenCPU safely on top of an existing system, and easily remove it at any time.

1.3 Getting the system up-to-date

Throughout this manual it is assumed that you have shell access, either through a terminal or over SSH. Before beginning installation of `OpenCPU`, make sure you are running `Ubuntu 12.04 (Precise)` by entering:

```
cat /etc/*release
```

If it turns out the system is running an older version of `Ubuntu`, upgrade the OS to 12.04 first. If the system is indeed 12.04, continue by updating existing software packages on the system to the latest versions:

```
sudo apt-get update
sudo apt-get upgrade
```

Once the system is up to date, you can begin installing `OpenCPU`.

1.4 OpenCPU Installation

Start by adding the `opencpu-0.7` package respository to the system:

```
sudo apt-get install python-software-properties
sudo add-apt-repository ppa:opencpu/opencpu-0.7
```

The system will ask for confirmation on importing the public key. After the repository has been added to the system, update the package list:

```
sudo apt-get update
```

Once this has succeeded `OpenCPU` can be installed. Enter:

```
sudo apt-get install opencpu-server
```

This will be sufficient to get started with `OpenCPU`. `OpenCPU` has many dependencies, and first installation might take a while. If the installation finished without any problems, it will display the ip address of the host at the very end, which you can can open in your browser and use to test the server.

1.5 Uninstall `OpenCPU`

To remove `OpenCPU` from a system, enter:

```
sudo apt-get remove --purge opencpu-server
sudo apt-get autoremove
```

Note that if objects were saved in the store, you might have to remove this manually by removing contents of the `/mnt/export` directory.

2 Administration

The default install of `OpenCPU` contains 2 services that you can independently control. To control the `OpenCPU` web server, use:

```
sudo service opencpu-server {start | stop | restart}
```

This will completely enable/disable `OpenCPU` and restart the webserver. To enable/disable the sandbox, use:

```
sudo service opencpu-sandbox {start | stop | restart}
```

The sandbox is based on AppArmor and is there to prevent users from running malicious code through the R API. It is enabled by default, but for debugging or testing purposes you might want to temporary disable the sandbox.

2.1 Configuration Files

All configuration files for `OpenCPU` are stored in `/etc/opencpu`. The main server configuration file is called `server.conf`. To edit it, use a text editor, like nano:

```
sudo nano /etc/opencpu/server.conf
```

The configuration file is formatted in JSON. The syntax should be pretty intuitive. Make sure that when you edit it, you validate that it is still valid JSON or the server might not be able to read it. Below a list of the settings that can be modified in the server.conf file:

- `github.clientid` – Github client ID for this domain.
- `github.secretfile` – points to file with the github secret.
- `disable.eval` – disables evaluation of code in parameters.
- `enable.cors` – enable/disable CORS headers.
- `whitelist` – either `false` or an array of packages that are allowed.
- `key.length` – length of the hashkeys for store objects.
- `job.timeout` – soft limit on R jobs (in seconds)
- `time.limit` – harder limit on R jobs (in seconds)
- `max.storesize` – max size of an object in the store.
- `storedir` – directory of the object store.
- `cache.store` – cache-control value for getting objects (in seconds)
- `cache.call` – cache-control header value for function calls (in seconds)
- `cache.help` – cache-control header value for help files (in seconds)
- `return.warnings` – if warnings should be included in case of an error.

- `accesslog` – file with access log entries
- `errorlog` – file with error log entries
- `syslib` – library of `OpenCPU` system dependencies
- `libpaths` – paths where to look for packages
- `preload` – an array of packages to preload when the server starts.

After editing configurations, the server has to be restarted to activate them:

```
sudo service opencpu-server restart
```

2.2 Apache2

`OpenCPU` is based on the Apache2 webserver. It should not be necessary to manually edit these configurations, but in case you want to hack around, below a short intro to managing Apache2 on `Ubuntu`. To manage the server daemon:

```
sudo service apache2 {start | stop | restart}
```

This command calls the `/etc/init.d/apache2` script which should usually not be edited. The main configuration file for apache2 is located at

```
/etc/apache2/httpd.conf
```

However by convention this file should also rarely be edited. Custom configurations are located at:

```
/etc/apache2/mods-available/
/etc/apache2/sites-available/
```

These custom configurations can be activated and de-activated as follows:

```
sudo a2enmod mod_R
sudo a2dismod mod_R
sudo a2ensite opencpu-user
sudo a2dissite opencpu-user
```

These commands create or remove symoblic inside links to available configuration files inside the following directories:

```
/etc/apache2/mods-enabled/
/etc/apache2/sites-enabled/
```

All files in these directories are automatically included by the main `httpd.conf` file. The `OpenCPU` sites are defined in the following files:

```
/etc/apache2/sites-available/opencpu
/etc/apache2/sites-available/opencpu-user
/etc/apache2/sites-available/opencpu-apps
```

The Apache2 log files `access.log` and `error.log` are located at

```
/var/log/apache2/
```

2.3 HTTPS access and SSL certificates

By default Apache only listens to port 80 (HTTP). To enable HTTPS, run the following commands:

```
sudo a2enmod ssl
sudo a2ensite default-ssl
sudo service apache2 restart
```

This will enable SSL access to your website. Note that SSL is not compatible with the caching server; HTTPS requests will never be cached on the server.

2.4 Certificates

By default, Apache uses self signed a.k.a. snakeoil certificates. This is convenient for development servers, but in a production setting these should be replaced by SSL certificates signed by an official Certificate Authority.

The https configurations and locations of the certificates are defined in

```
/etc/apache2/sites-available/default-ssl
```

This file also contains detailed comments with configuration instructions.

When the caching server is enabled, the SSL certificates should be installed on NginX instead of Apache. Modify the file:

```
/etc/nginx/sites-available/opencpu
```

and look for the directives `ssl_certificate` and `ssl_certificate_key`.

3 The Caching / Load balancing Server

`OpenCPU` has an optional caching/load balancing server based on NGINX. To install it, enter:

```
sudo apt-get install opencpu-cache
```

When enabled, the server will automatically take over port 80 and 443. The caching server can be enabled/disabled using:

```
sudo service opencpu-cache {start | stop | restart}
```

The server configuration can be edited to include additional back-ends and hence work as a load balancer as well. The balancer is configured using NGINX configurations found in `/etc/nginx`.