

Maintaining the Debian Charliecloud Package

Peter Wienemann

May 8, 2025
HPSF Conference 2025

Outline

- 1 Debian
- 2 Charliecloud in Debian
- 3 Debian Quality Assurance Tools
- 4 Fruitful Interplay

Debian Project

- Linux distribution developed by volunteers around the world
- Strong focus on free software
- Known for stability and smooth upgrades
- New (major) versions released approximately every two years
 - Support by Debian security team for ≈ 3 years
 - Support by longterm support (LTS) team until ≈ 5 years after release
 - Extended longterm support (ELTS) up until ≈ 10 years after release can be bought
- Base for many derivative distributions
- $> 30\,000$ source packages, $> 120\,000$ binary packages

Debian Archive Structure

- Suites: unstable (sid), testing, stable, oldstable, oldoldstable
 - unstable: Any Debian Developer/Maintainer can upload new package versions at any time, new packages need FTP-master review
 - testing: Staging area for next stable release, packages automatically migrate within a few days from unstable to testing unless autopkgtests fail or release-critical (RC) bugs filed (exception: freeze phase), unfixed RC bugs lead to auto-removal
 - (old)stable: Controlled by release team, contents of testing become next stable after freeze phase ($\approx$ every two years)
- Areas: main, contrib, non-free, non-free-firmware
 - Only main is part of Debian
 - Nevertheless use of other areas is supported

Debian (Binary) Packages

- Debian binary packages distributed in deb format
 - ar archive file
 - Contains three files:
 - `debian-binary`: deb format version
 - `control.tar.xz`: tar archive containing metadata, scripts run during installation/removal
 - `data.tar.xz`: tar archive containing data files to install
- deb files can be installed/removed with `dpkg` package manager (medium-level tool) or `apt`/`apt-get`/`aptitude` etc. (high-level tools).

Debian Package Maintenance

- Tasks of package maintainers (among others)
 - Provide recipe to build deb package(s) from source
 - Address bugs, forward them to upstream projects where applicable
 - Update package to integrate new upstream releases
 - Sometimes fixing bugs in Debian packages requires backporting upstream fixes to account for misalignment between upstream releases and Debian release cycle
 - Keep package compliant with various requirements:
 - Debian Free Software Guidelines
 - Legal requirements (licences/copyright)
 - Debian Policy
- There is no canonical toolchain/workflow to maintain packages in Debian but there are recommendations

Building Debian Packages

- Everything is built from scratch (e. g. regenerate configure by running autoreconf)
- Build process must not download software from the internet
- Avoid using vendored software wherever possible
- Built in fresh, clean, minimal environment (separate kernel namespaces or chroot)
 - Ensures build dependencies are complete
 - Ensures history of build host does not impact build
 - Ensures no downloads are done during build (dedicated network namespace)

Brief History of Charliecloud in Debian

- Initially packaged by Lucas Nussbaum at the end of 2017/beginning of 2018
- Team-maintained under the umbrella of Debian HPC Team
- I joined packaging effort in January 2018, have been continuously contributing since then
- Has been part of three Debian releases so far: Buster (July 2019), Bullseye (August 2021) and Bookworm (June 2023)

Charliecloud Package Structure in Debian

- Charliecloud distributed over six binary packages in Debian
 - charliecloud: Meta package installing build and runtime tools
 - charliecloud-common: Common files for all packages
 - charliecloud-builders: Containing build tools and corresponding man pages
 - charliecloud-runtime: Containing runtime tools and corresponding man pages
 - charliecloud-doc: Containing documentation
 - charliecloud-tests: Containing test suite

Debian Quality Assurance Tools

- Assembling a distribution means significant integration work:
Make sure all packages work together (at build time and runtime)
- Prerequisite: Powerful QA tools
- They differ in terms of
 - what they cover
 - when they are run
 - on which architectures they are run
- Will present some of them on following slides

Lintian

- Debian package linter, checks for (among others)
 - packaging errors/policy violations
 - deprecations
 - other common errors (e. g. typos, etc.)
- Examples for Charliecloud issues raised by Lintian
 - Executables without man pages:
<https://github.com/hpc/charliecloud/pull/110>
 - Erroneously executable files:
<https://github.com/hpc/charliecloud/issues/648>
 - Wrong man page section:
<https://github.com/hpc/charliecloud/pull/880>
 - Typos: <https://github.com/hpc/charliecloud/pull/647>

piuparts

- Tool testing installation, upgrade and purging of packages
- Tests performed in temporary, minimal environment (e. g. kernel namespaces or chroots)
- Compares state before installation and after removal
- Reports differences

autopkgtest

- Runs test suite to test package functionality
- To be set up by package maintainers (but can e. g. recycle upstream tests)
- Run on maintainer's local machine, as part of CI pipeline after push to repository (prior to upload) and after upload
- Post-upload tests run on all release architectures
- Also used to check for regressions in case of new versions of dependencies
- Regressions block migration to testing
- Autopkgtest for Debian Charliecloud package: Basically runs “90 seconds to Charliecloud” tutorial

autopkgtest (2)

- Charliecloud issue found by autopkgtest: Syntax warnings with Python 3.12,
<https://github.com/hpc/charliecloud/issues/1834>
- Some fallout issues brought about by introduction of autopkgtest:
 - Add `-q/--quiet` option:
<https://github.com/hpc/charliecloud/issues/1660>,
<https://github.com/hpc/charliecloud/issues/1613>,
<https://github.com/hpc/charliecloud/issues/1732>,
 - Silent `ch-image` build failure on archs without `seccomp` support:
<https://github.com/hpc/charliecloud/issues/1771>
 - `seccomp` support for `s390x`:
<https://github.com/hpc/charliecloud/issues/1772>

blhc

- Build log hardening check
- Scans build logs for missing hardening flags
- Charliecloud issue found by blhc
 - Missing \$(CPPFLAGS) \$(LDFLAGS) in Makefile for some examples:
<https://github.com/hpc/charliecloud/pull/462>

reprotest

- Checks whether package build is reproducible, i. e. build artefacts are bit-by-bit reproducible by anybody
- Allows to check whether given binary package was actually built from given source package
- Charliecloud issue found by reprotest
 - Locale dependent ordering of generated bats file contents:
<https://github.com/hpc/charliecloud/issues/1849>

Salsa CI Pipeline

The screenshot shows the Salsa CI Pipeline interface for the 'charliecloud' project. The top navigation bar includes the 'debian' logo, an 'Explore' link, and 'Sign In' and 'Register' buttons. A search bar is located on the left. The left sidebar contains a 'Project' section with links to 'charliecloud', 'Manage', 'Plan', 'Code', 'Build', 'Pipelines', 'Jobs', 'Pipeline schedules', 'Artifacts', 'Deploy', 'Operate', 'Analyze', and 'Help'. The main content area displays the pipeline details for 'Update changelog for 0.38-3 release'. It shows a 'Passed' status, the creator 'Peter Wienemann', and the commit '894ff75'. The pipeline is for the 'latest' branch and has 12 jobs, 14 minutes 17 seconds of execution time, and is queued for 30 seconds. The 'Pipeline' tab is selected, showing a flow of three stages: 'provisioning' (with 'extract-source'), 'build' (with 'build', 'build i386', and 'build source'), and 'test' (with 'autopkgtest', 'bitc', 'lintian', 'missing-breaks', 'piuparts', 'reprotest', 'test-build-all', and 'test-build-any').

debian Explore

Debian HPC team / charliecloud / Pipelines / #846026

Sign In Register

Search or go to...

Project

- charliecloud
- Manage
- Plan
- Code
- Build
- Pipelines
- Jobs
- Pipeline schedules
- Artifacts
- Deploy
- Operate
- Analyze
- Help

Update changelog for 0.38-3 release

Passed Peter Wienemann created pipeline for commit `894ff75` 1 week ago, finished 1 week ago

For `main`

latest 60 12 jobs 14 minutes 17 seconds, queued for 30 seconds

Pipeline Jobs 12 Tests 8

Group jobs by Stage Job dependencies

provisioning

- extract-source

build

- build
- build i386
- build source

test

- autopkgtest
- bitc
- lintian
- missing-breaks
- piuparts
- reprotest
- test-build-all
- test-build-any

Source: <https://salsa.debian.org/hpc-team/charliecloud/-/pipelines/846026>

Regular Archive Rebuilds

- Debian regularly rebuilds its archive (or parts thereof) to look for issues which were/will be introduced by changes
- Detects many issues especially after profound changes (e. g. switching to new compiler version)
- Bugs often filed well in advance, severity level raised over time until they become release-critical (leading to auto-removal if they remain unfixed)
- Allows early heads-up to upstream projects – ideally leading to new upstream releases with fixes

Fruitful Interplay

- Charliecloud developers regularly seek feedback from distribution maintainers
- Bidirectional information flow, a few examples:
 - Remove troublesome debian link in upstream repository:
<https://github.com/hpc/charliecloud/issues/118>
 - Turn off autotools maintainer mode?
<https://github.com/hpc/charliecloud/issues/595>
 - Switch documentation tools?
<https://github.com/hpc/charliecloud/issues/836>
 - Remove upstream Debian packaging?
<https://github.com/hpc/charliecloud/issues/935>
 - Storage directory support period:
<https://github.com/hpc/charliecloud/pull/1507>
 - Path adjustments, made Debian specific patch obsolete:
<https://github.com/hpc/charliecloud/pull/1724>

Summary

- In healthy environment, upstream projects and distributions cross-fertilize each other
- Different focuses, different toolsets, more people broaden coverage, resulting in better end products
- Last but not least being part of a healthy community is more fun
- Charliecloud is an exemplary upstream project