

Introducing Debusine



Enrico Zini

Carles Pina i Estany

Introducing Debusine



Carles Pina i Estany, Colin Watson, Emilio Pozuelo Monfort, Enrico Zini, Helmut Grohne, Jochen Sprickerhof, Raphaël Hertzog, Stefano Rivera, Sylvain Beucler, with input from various Debian teams

What is Debusine

Scheduling and distribution of Debian-related tasks to worker machines

Package build, QA checks, to collecting distribution-wide data...

Specific for Debian, generic enough to aim at handling most of Debian's need

Debusine for Debian development

Easy way to run tools such as sbuild, autopkgtest, lintian, blhc, piuparts...

This can turn out to be a convenient infrastructure to implement PPA/bikesheds for Debian

Next generation Debian infrastructure

Possible replacement for wanna-build and builddd, both for Freexian's internal use and Debian

Pipelines with build, QA, approval and signature steps that are triggered by package uploads

Run distribution-wide QA experiments

A way to allow DDs to run archive-wide QA experiments, like full rebuilds, using cloud resources

Debian-friendly people and companies will be able to sponsor external computing time, while trusted builds will remain under Debian's control

Actual implementation

- Python, Django, supporting bullseye and bookworm
- Server/worker architecture
- Public project on Salsa, all development happens in public issues and MRs
- GPL3, no CLA
- Unit tests, integration tests

Project governance

- Started as something useful internally in Freexian
- It looks like it may be useful also in Debian in various ways
- Got funding to develop something concrete
- If it ends up being used, then it means it worked

Project governance

- Currently guaranteed development: STF funding for 3 people full time for a year, focusing on fixed set of milestones
- If Debusine turns out to be useful to/used by Debian, then Debian should own it
- Everything is set up for ownership and governance to move away from Freexian

Milestones and deadlines!

The Sovereign Tech Fund supports the development, improvement, and maintenance of open digital infrastructure. Its goal is to sustainably strengthen the open source ecosystem, focusing on security, resilience, technological diversity, and the people behind the code.

STF is funding Freexian's work in Debusine.

We have a number of milestones funded by STF, with deadlines. We release when it's ready, and in this case we already know when it's going to be ready!

M1: add QA tasks

Deadline: 2023-12-31

Allows any developer to immediately run extensive QA analysis on their ongoing packaging work, without requiring a full setup of the QA toolchain on their own system

- DONE: salsa SSO integration
- DONE: design the draft interfaces for the Autopkgtest and Lintian tasks
- DONE: implement the Autopkgtest task
- DONE: implement the Lintian task

M1: add QA tasks

Deadline: 2023-12-31

- **DONE:** reach out to autopkgtest, ci.debian.net and lintian maintainers to validate our design
- **MUST:** deploy and announce a debusine instance accessible to Debian Developers
- **MUST:** implement user web views for the results of the Autopkgtest and Lintian tasks

M1: add QA tasks

Deadline: 2023-12-31

- DONE: ask DSA for a debian.org machine
- DONE: users can upload artifacts through the web UI
- DONE: users can add new tasks through the web UI
- DONE: implement the piuparts task
- MAY: command line tool to upload packages and trigger QA tests on them
- MAY: implement a “blhc” task

M2: automation, orchestration, collection

Deadline: 2024-04-30

Automatically triggered QA tasks

- MUST: Implement basic "workflows"
- MUST: Implement "collections" of artifacts
- MUST: mirror an external repository in debusine, as a collection
- MUST: implement a way to keep collections in sync

M2: automation, orchestration, collection

Deadline: 2024-04-30

- SHOULD: automatically recompute old QA data
- SHOULD: prioritize user tasks over automated QA tasks
- SHOULD: start workflows from the web UI
- SHOULD: show workflow progress in the web UI
- SHOULD: test with a real workload (lintian and autopkgtest)
- MAY: View for the results of lintian analysis (lintian.debian.org?)
- MAY: Logical tests in workflows

M3: feature parity with builddd

Deadline: 2024-06-30

Possible replacement for the builddd network

- MUST: dogfooding: replace Freexian's internal rebuilddd
- MUST: present debusine to the wanna-build team
- MUST: schedule all the binary builds required for a new source upload
- MUST: wait until the required build dependencies are available
- MUST: upload source and binary packages
- MUST: support for bin-NMUs and give-backs

M3: feature parity with builddd

Deadline: 2024-06-30

- SHOULD: workflow step: barrier
- SHOULD: web UI to navigate historical records of builds and build attempts, with stable URLs that can be used in online discussions and bug reports
- MAY: Web UI to navigate pending package builds
- MAY: realtime view of build logs

M4: workflow for security team

Deadline: 2024-09-30

Debusine can be a useful tool for the Debian security team

- MUST: involve the Debian Security Team in the design
- MUST: private workspaces for embargoed security updates
- MUST: publish from a private workspace to a public one
- MUST: reverse autopkgtest
- MUST: build debdiffs and show them in the web UI
- MUST: manual validation by a human
- MUST: package signing for secure boot

M4: workflow for security team

Deadline: 2024-09-30

- SHOULD: leave annotations in the web interface
- SHOULD: sign .dsc and .changes on the server
- SHOULD: signature of .changes with a worker key
- MAY: mark which workers can build security updates
- MAY: track which worker built/signed a package

M5: scaling with the cloud

Deadline: 2024-12-31

Debian developers can schedule their own archive-wide tests/experiments without disrupting regular day-to-day work

- Use external cloud infrastructure spin up workers to run large tests/experiments
- Collect worker resource statistics to autoscale cloud resources based on expected workloads
- Support multiple cloud providers from the start to avoid lock-in

Introducing Debusine

Live demo!

Introducing Debusine

- <https://freexian-team.pages.debian.net/debusine/>
- <https://salsa.debian.org/freexian-team/debusine/>
- you can watch project on Salsa or specific discussions, and join discussions
- as soon as there'll be a DD accessible instance, you can try it out during all stages of development

Introducing Debusine

Q&A