

Alpaquita —

ENTERPRISE LINUX DISTRIBUTION FOR
EFFICIENT CONTAINERIZED APPLICATIONS

SEPTEMBER 2024

Abstract green 3D rings or loops, resembling a stylized 'Q' or a series of overlapping circles, positioned in the bottom right corner of the image.

Why BellSoft Created Alpaquita Linux?

In the vast landscape of Linux distributions, none fully meet the stringent demands of modern, containerized Java applications, which require:

- **A lightweight image:** a small static footprint for sustainable and scalable containerized Java environments.
- **Resource efficiency:** optimized performance and RAM consumption for predictable scalability and lower cloud bills.
- **Security and reliability:** up-to-date security features, thorough CVE tracking, and the availability of long-term support (LTS) releases and end-to-end support with service level agreements (SLAs).

One may argue that Alpine Linux provides most of the above. In 2019, BellSoft added support for musl libc to OpenJDK project. However, after thorough testing, we discovered several shortcomings with Alpine Linux: it lags in performance, offers limited cloud support, and does not provide adequate support for Java, particularly with GraalVM. Additionally, the absence of full security feature sets, differences in libc (musl vs. glibc) complicating migrations, and the lack of official LTS releases leave gaps in stability and long-term viability.

THEREFORE, BELLSOFT CREATED ALPAQUITA LINUX, A DISTRIBUTION DESIGNED TO FILL THIS GAP IN THE MARKET, PROVIDING AN ENTERPRISE OS FOR EFFICIENT AND SECURE CONTAINERIZED JAVA APPLICATIONS.

BELLSOFT IS THE ONLY VENDOR ON THE MARKET THAT OFFERS SUPPORT FOR JAVA AND OS. ALPAQUITA IS A VIABLE ALTERNATIVE TO ALPINE LINUX FOR ENTERPRISES WITH HIGH SECURITY STANDARDS.

JEP 386: ALPINE LINUX PORT. BELLSOFT PORTED THE JDK TO ALPINE LINUX AND TO OTHER LINUX DISTRIBUTIONS THAT USE MUSL AS THEIR PRIMARY C LIBRARY, ON BOTH THE X64 AND AARCH64 ARCHITECTURES.


Introducing Alpaquita Linux

Alpaquita is a lightweight and secure, enterprise-level Linux distribution primarily optimized for containerized Java applications. It retains the best features of Alpine Linux while incorporating additional perks for enhanced development, security, and performance.

Essential Features of Alpaquita:



Base Image Size:


3.26 MB



Runs on
X86_64

Libraries Supported

 **GLIBC** |  **MUSL-DEF
MUSL-PERF**





Works with
Python




Runs on
AArch64

The only Linux-based distribution optimized for Java™






Support for
GraalVM Native




Works with
C







Works with
C++

Support for
the OpenJDK CRaC API

Production Ready Containers, both **JDK** and **JRE**

 Alpaquita Linux +  Liberica JDK Lite


Available on

 |  |  | 

Docker Hub | GitHub | AWS | Azure

-30 %
Disk Space


30 %
RAM Usage



4-year
LTS releases

EULA License 

Free For **Commercial**
and **Production** use



24/7 support

Alpaquita Linux, as a core component of Alpaquita Cloud Native Platform, is an ideal base OS for creating the smallest and high-performing containers that **immediately reduce your Cloud costs up to 20%**. Please refer to [the Alpaquita Cloud Native Platform page](#) and white paper to find out more.

KEY TECHNICAL ADVANTAGES

Kernel:

- Always based on Linux LTS (6.10 currently)
- SecureBoot, signed modules
- Better performance tunables, optimized size

LIBC:

- musl and glibc support
- Optimized musl, optimized mallocs

Architectures:

- x86_64, AArch64

Userspace Based on Alpine Linux Aports:

- busybox, binutils, ssh
- main/aports & apk package manager
- no graphics

MORE ON LIBC AND GLIBC

Userspace Performance:

- Liberica Performance Edition
- Liberica NIK with extra GCs
- Improved compilation techniques for packages leverage modern CPU capabilities

Cloud native:

- Pre-built and up-to-date containers on Docker Hub, GitHub Container Registry (ghcr), Microsoft Container Registry (mcr), and Amazon Elastic Container Registry (ECR)
- VMs readily available in AWS and Azure
- Improved QEMU support

Developer experience:

- native WSL support, VS code app
- User-friendly installer, qcow images

We optimized stock musl to address its known limitations. The performance of our musl perf matches or exceeds that of glibc (see the benchmarks on graphs below). Importantly, musl-perf is 100% compatible with stock musl, ensuring seamless migration without any compatibility issues.





We introduced a **glibc-based variant of Alpaquita**, which in many cases is smaller and more efficient than other glibc-based distributions in some cases (see throughput study results). This enables companies with workloads that work better with glibc-based distros or those hesitant to switch libc implementations to benefit from lightweight and performant capabilities of Alpaquita OS.



ALPAQUITA VS ALPINE LINUX

While Alpaquita shares its foundation with Alpine, it incorporates numerous enhancements tailored to create a superior enterprise environment for development and deployment. Below is a comparative overview highlighting the advantages of Alpaquita over Alpine:

FEATURE	 ALPAQUITA	 ALPINE
Open source	Yes	Yes
Free to use	Yes	Yes
Base image size (as of Feb 2024)	3.26 MB	3.25 MB
Free community support	Yes, for Stream version	Yes
Commercial Support	Yes, as part of Alpaquita Cloud Native Platform	No
glibc support	Yes	No
musl support	musl def / musl perf	musl def
Ease of migration from other libc	Yes	No
SecureBoot	Yes	No
LTS support period	4 years	n/a
Stable releases	LTS releases aligned with Linux Kernel LTS	Stable releases each May and November
Rolling releases	Alpaquita Stream, stable releases	Alpine Edge, releases for testing
Platform compatibility	X86_64, AArch64	x86, x86_64, ARMhf, ARMv7, AArch64, ppc64le, s390x
Full Java support	Yes	No
GraalVM support	Yes	No
Production-ready Java containers	Yes	No
Containers with JRE	Yes	No
Containers with CRaC**	Yes	No
License	End-User License Agreement (EULA)	GPL2 mostly, MIT for musl, various licenses for software components

By choosing Alpaquita, you retain the small image size of Alpine while gaining access to enterprise-level support for your containers.

Alpine's small size comes from using musl, a compact but slower library. Alpaquita offers a faster, glibc-based image while still being lightweight.

Our engineers have extensively optimized the musl library, often surpassing the performance of the default implementation.

Alpine uses musl exclusively, so migrating to glibc requires extra effort.

BellSoft provides continuous updates for Alpaquita that are suitable for production use. In contrast, Alpine Edge releases are primarily for testing.

Alpaquita is optimized for Java applications and integrates seamlessly with Liberica JDK and BellSoft's tools.

GraalVM binaries are typically built for glibc, which is not supported by Alpine Linux.

BellSoft provides pre-built containers with Alpaquita and Liberica JDK Lite. These images are continuously updated, ensuring that each rebuild includes the latest updates for both the OpenJDK and the Linux distribution.

By implementing an EULA and thoroughly verifying all packages, we've mitigated risks of license violations by substituting certain software.

As such, Alpine Linux is a platform where only the community can solve your issues in an unspecified time, whereas there is a commercial support for Alpaquita Linux provided as the part of Alpaquita Cloud Native Platform. Still, even free Alpaquita Stream releases are full of features that you can use within your own containers for free.

**CRaC (Coordinated Restore at Checkpoint) is a Java API that helps to slash startup and warmup times of Java apps from seconds to milliseconds.

Alpaquita Linux Distribution: Features Explained

SECURITY

Alpaquita Linux was created to meet the requirements of enterprise security audits. Along with a clear support roadmap for LTS releases and security updates, Alpaquita offers a full security suite with modern security features and CVE tracking.

Security — Kernel

- **Kernel lockdown enabled early in boot:** Prevents both direct and indirect access to a running kernel image.
- **Secure Boot support:** Ensures that only trusted software is loaded during the boot process.
- **Kernel module signing with SHA-512:** Increases kernel security by disallowing the loading of unsigned modules or modules signed with an invalid key.
- **Security hardening options:** Includes settings like `CONFIG_SECURITY_DMESG_RESTRICT=y` to avoid kernel address exposures for non-root users.

Security — Userspace

- **Minimal extra components:** Reduces the attack surface.
- **Userspace compilation options are provided for security hardening:**
 - `fPIE, -pie` (**Position Independent Executable**): Protects from memory corruption attacks.
 - `-fstack-protector-strong`: Enables stack overflow security checks for routines with any type of buffer.
 - `-Wformat-security` : Warns about potential security problems in format strings.
 - `-Wtrampolines` : Warns when generating trampolines in objects that require executable stacks.

Security - Addressing CVEs

- **Member of OpenJDK vulnerability group:** BellSoft is part of the OpenJDK vulnerability group.
- **CVEs in Linux components:** Pulling fixes from upstream (aports, kernel.org, musl, glibc) with a 24-hour build SLA for delivery to repositories and containers
- **CVEs in OpenJDK/GraalVM:** Following quarterly security releases
- **BellSoft offers Security Advisory that includes:**
 - Full open listings of addressed CVEs and affected components
 - OpenJDK/Liberica JDK security advisory
 - Tooling and scanning to support efforts:
 - Internal CVE DB checking against MITRE/NIST
 - CVE checker
 - Static code and binary analysis tools.

PERFORMANCE

Alpaquita Linux was built with high performance in mind. We removed some older components of Alpine Linux that made it compatible with outdated hardware and optimized the configs of the core to reduce the static footprint and increase the performance of Alpaquita. We will continue this optimization in future releases.

See the White Paper on Alpaquita Cloud Native Platform to compare the speed of Alpaquita versus competitors in containers.

Kernel Performance (crucial changes)

Kernel built with config options to support better performance, including:

- **NUMA options:** Enables memory placement with a NUMA-aware scheduler, as well as memory and taskset management.
- **Task group support:** Provides efficient task grouping for better resource allocation.
- **Improved latency:** Uses IOSCHED options to optimize performance for interactive and soft real-time applications.
- **Enhanced energy efficiency and reduced OS jitter:** Incorporates the NO_HZ option to minimize interruptions and improve overall system responsiveness.

These are just a few of the most notable changes; the full list of kernel optimizations is extensive.

Optimized musl LIBC

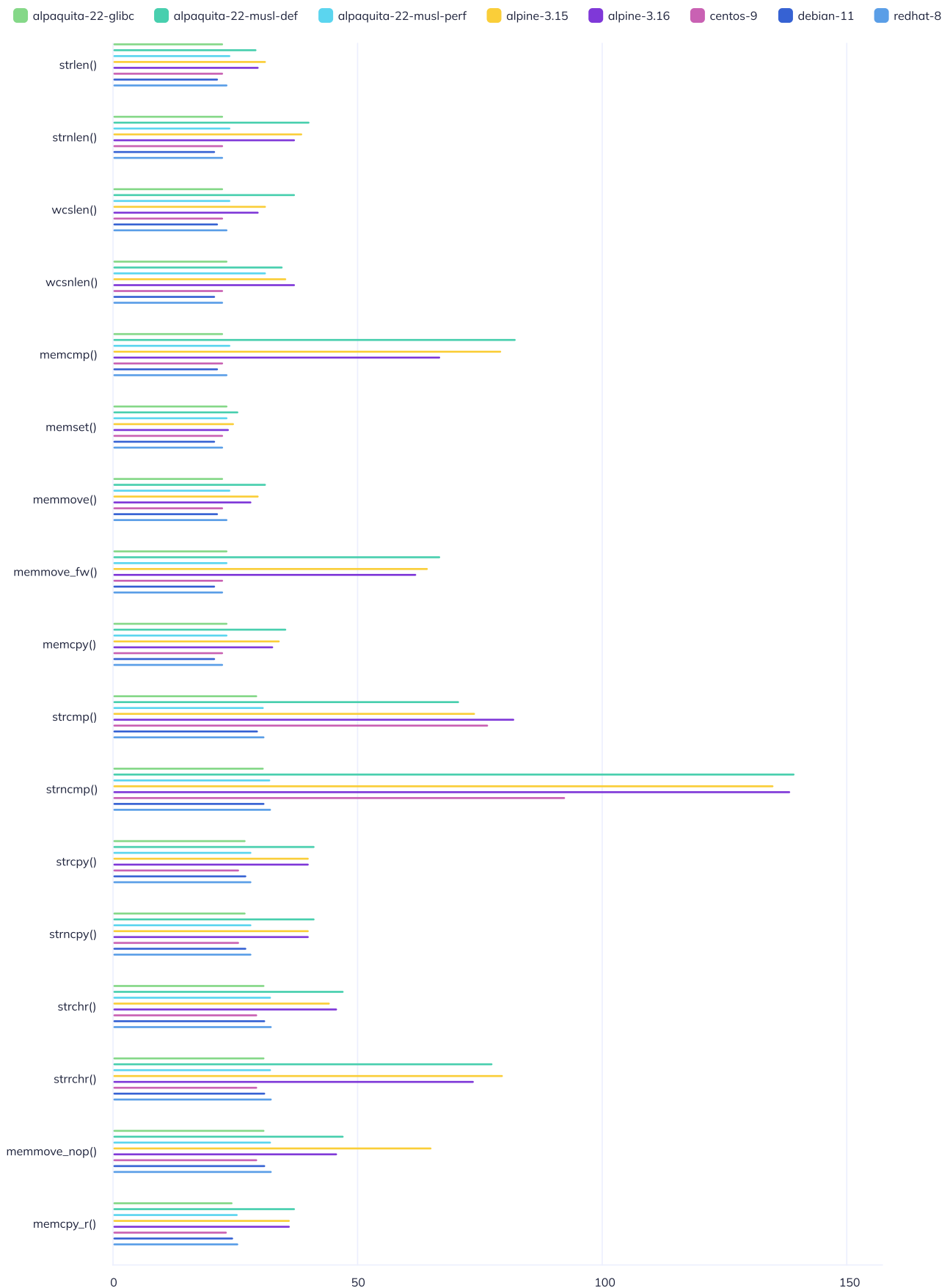
To boost performance, Alpaquita has implemented significant optimizations to the default musl libc (musl def) library.

	MUSL PERF	MUSL DEF
Default Compiler Optimizations	built with -O3 for internal, malloc, and string subsystems, and -O2 for the rest.	musl def is optimized for size, using -Os for all subsystems.
Indirect Functions and ASM optimizations	supports indirect functions, internally implements CPU feature detection, and includes various CPU-specific ASM optimizations (AVX512, EVEX, AVX2, SSE4, etc.).	does not support indirect functions and includes only generic ASM optimizations for memcpy, memset, and memmove.

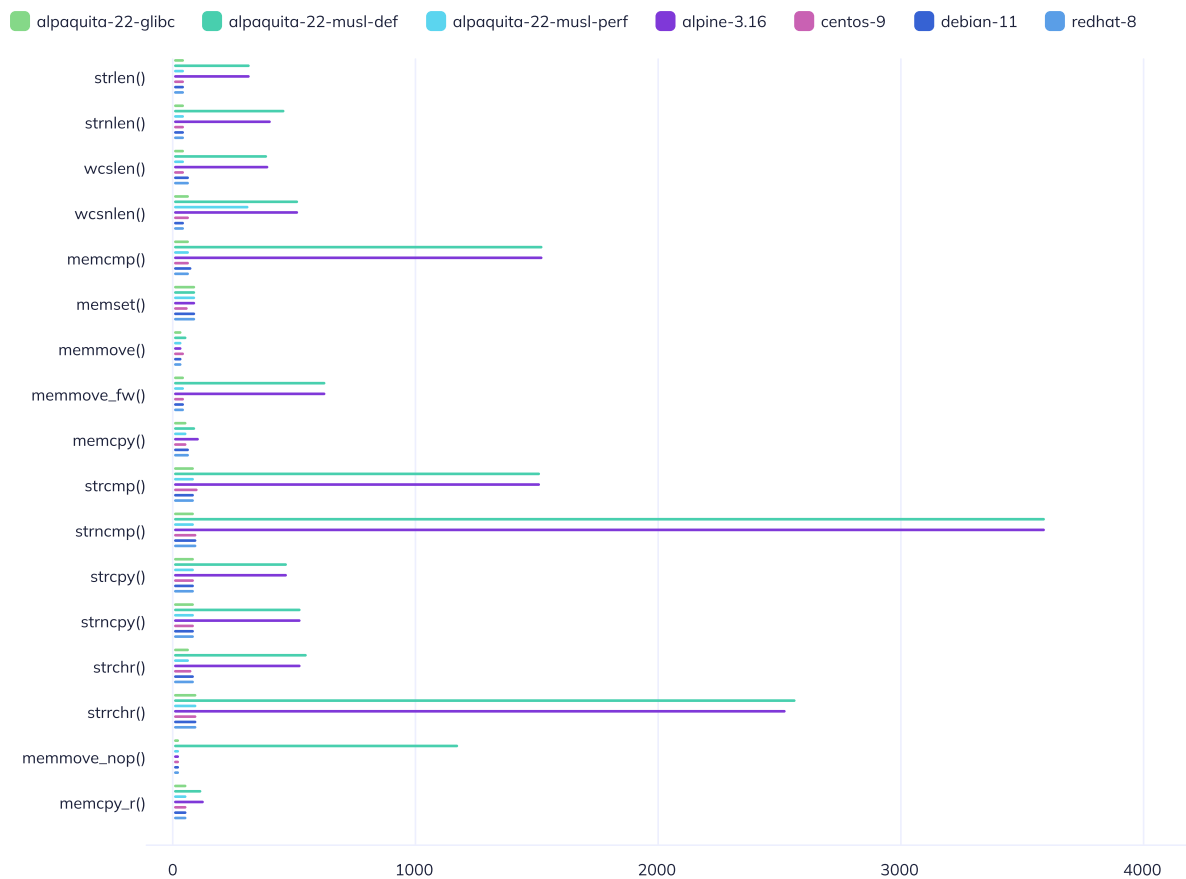
Alpaquita's optimized musl implementation is 100% compatible with the default musl, ensuring seamless migration from Alpine Linux. However, Alpaquita also offers default musl and glibc packages to provide alternatives for users with specific requirements.

For a comparison of the performance of Alpaquita's optimized musl (musl perf) with the default musl and glibc, [please refer to the graphs:](#)

String operations 132 chars, ns/op (lower is better)

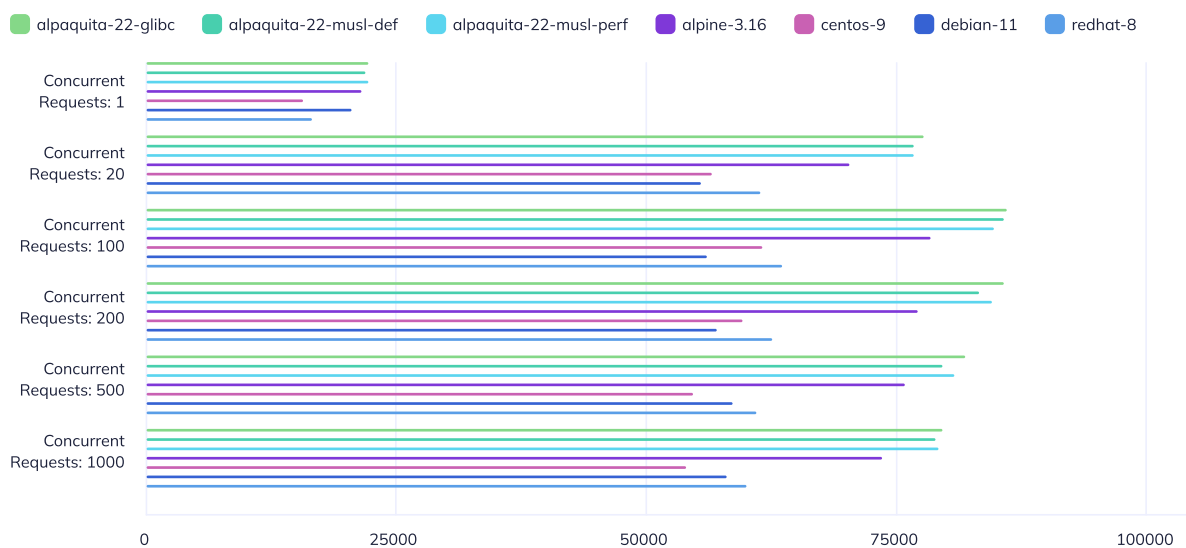


String operations 4100 chars, ns/op (lower is better)



Lower is better Note: musl def is the stock musl implementation, musl perf is optimized BellSoft musl implementation
Tests: basic functional string tests with 1 million iterations.
 Results are shown in relative avg speed, 1K/ns.
Machine: bare-metal, Intel Core i5-6600 CPU 3.30GHz.

Nginx, reqs/s (higher is better)



The [Nginx benchmark](#) provided by Phoronix was used to test the throughput.
 Results in requests per second, **higher is better**

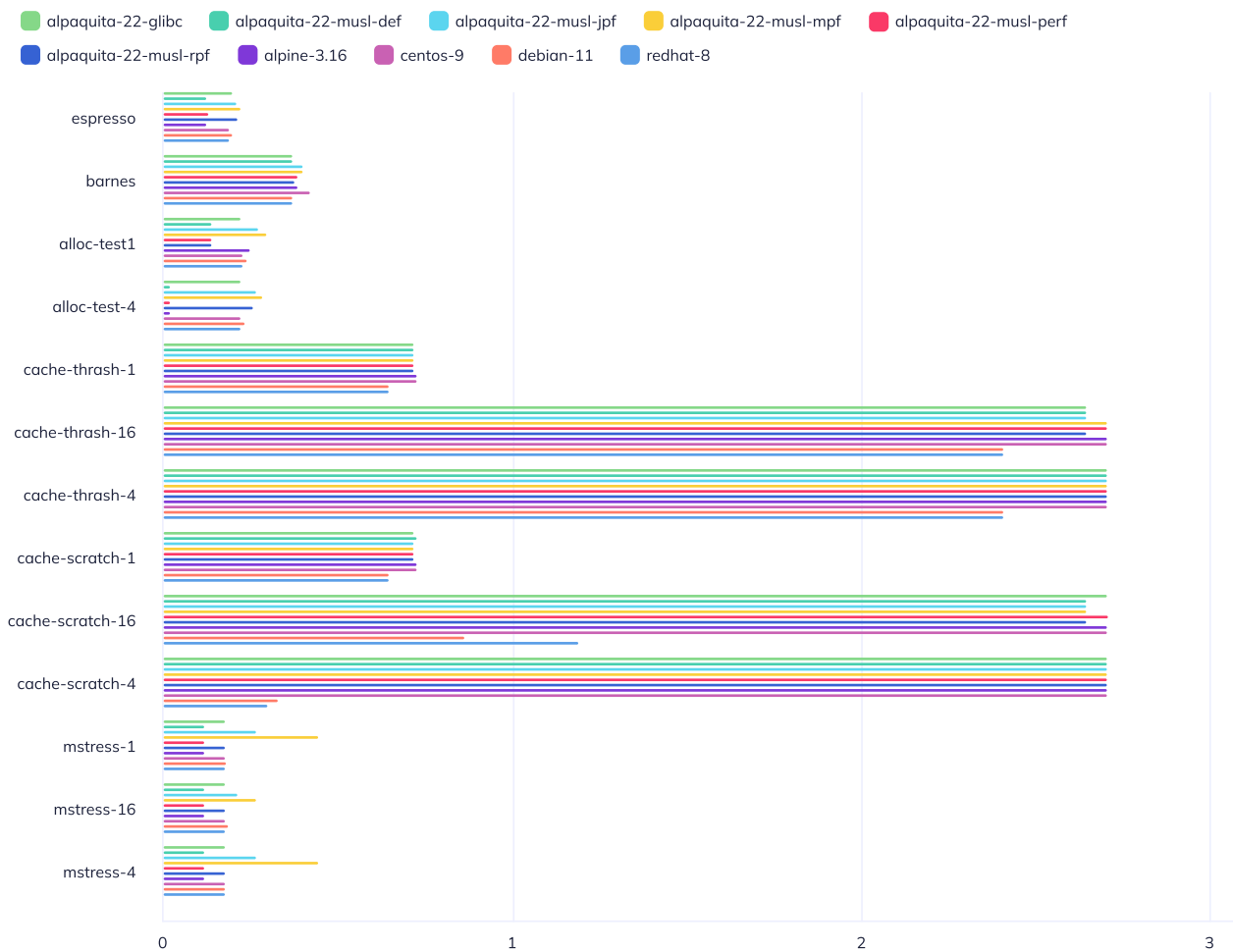
Malloc implementations

Alpaquita incorporates three additional malloc implementations tailored for different use cases:

- **mimalloc**: Small (128K), efficient, ideal for large scale low-latency services;
- **rpmalloc**: Even smaller (64K) with lock free thread caching;
- **jemalloc**: Time-tested implementation that emphasizes fragmentation avoidance and scalable concurrency support (616K).

For test results comparing the performance of different malloc implementations, [please refer to the graphs](#):

malloc performance, operation, 1/s (higher is better)



Results in relative speed 1/s, [higher is better](#)

malloc benchmarks utilized in the study:

- espresso: a programmable logic array analyzer in the context of cache aware memory allocation
- barnes: a hierarchical n-body particle solver [4], simulating the gravitational forces between 163840 particles
- alloc-test: simulates intensive allocation workloads with a Pareto size distribution
- cache-thrash: part of Hoard benchmarking suite, designed to exercise heap cache locality
- cache-scratch: introduced with the Hoard allocator to test for passive-false sharing of cache lines
- mstress: simulates real-world server-like allocation patterns, using N threads with with allocations in powers of 2 where objects can migrate between threads and some have long life times

Alpaquita configurations with mallocs:

- alpaquita-22-musl-perf-je — Alpaquita with musl-perf + jemalloc
- alpaquita-22-musl-perf-mi — Alpaquita with musl-perf + mimalloc
- alpaquita-22-musl-perf-rp — Alpaquita with musl-perf + rpmalloc

DaCapo, ms (lower is better)



[DaCapo benchmark suite](#) is a set of real-world Java applications with different memory loads used to evaluate system/CPU performance.

Results in ms, **lower is better**

DaCapo benchmarks utilized in the study:

- h2: executes a JDBCbench-like in-memory benchmark, executing a number of transactions against a model of a banking application
- fop: takes an XSL-FO file, parses it and formats it, generating a PDF file
- pmd: analyzes a set of Java classes for a range of source code problems
- xalan: transforms XML documents into HTML
- avrora: simulates a number of programs run on a grid of AVR microcontrollers
- jython: interprets a the pybench Python benchmark
- luindex: uses Lucene to indexes a set of documents
- sunflow: renders a set of images using ray tracing
- lusearch: uses Lucene to do a text search of keywords over a corpus of data comprising the works of Shakespeare and the King James Bible
- tradebeans: runs the daytrader benchmark via a Java Beans to a GERONIMO backend with an in memory h2 as the underlying database



VERSATILITY OF USE

Alpaquita Linux is an ideal solution for both server and cloud environments, supporting virtualization and cloud technologies. Whether you're managing a hybrid cloud application partially based on physical servers, partially in the cloud, Alpaquita is well-suited for both. Its small size enables the creation of microcontainers or smaller virtual machines (VMs).

Alpaquita base image can be deployed as is or customized with packages from Alpaquita's dedicated package repositories. Additionally, we offer images with tools and utilities for Java, Python, and C/C++ development.

CONVENIENCE

We made Alpaquita easy to set up and use. It is delivered through various platforms such as:

- SO distribution, mini rootfs tarball, WSL image;
- APK (Alpine Package Keeper) repository providing upgrades. APK utility is improved to support the new features;
- Docker Hub repositories with pre-built images:
 - Liberica Runtime Container,
 - Liberica NIK,
 - GCC,
 - Python.
- Images for AWS, GCP, Azure.
- Private docker registry for instant bug fixes, LTS behind 2 years.

SUPPORT

Licensing

Following our OSS philosophy, Alpaquita is open-source software that is free for commercial and production use. Our [End-User License Agreement \(EULA\)](#) is like any other OSS, and you can review it on our website.

- Open Source Software (OSS):
 - Mostly GPL
 - Liberica & NIK under GPLv2+CE
- Due diligence, Comprehensive review of licenses for all packages
- Clear and permissive EULA

Support options

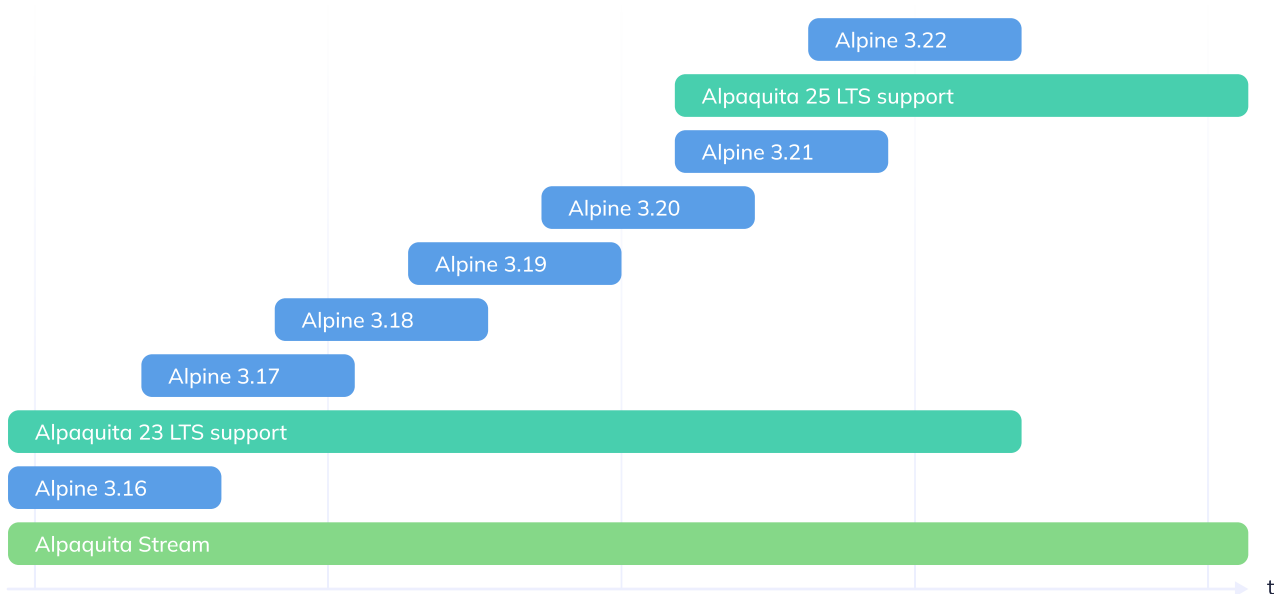
Alpaquita Stream version is free for commercial and production use and supported by the community.

The LTS version of Alpaquita is a part Alpaquita Cloud Native Platform and supported by BellSoft as one of the products in the platform. Please consult the [corresponding page](#) or download the White Paper on Alpaquita Cloud Native Platform to learn more.

Support cycle

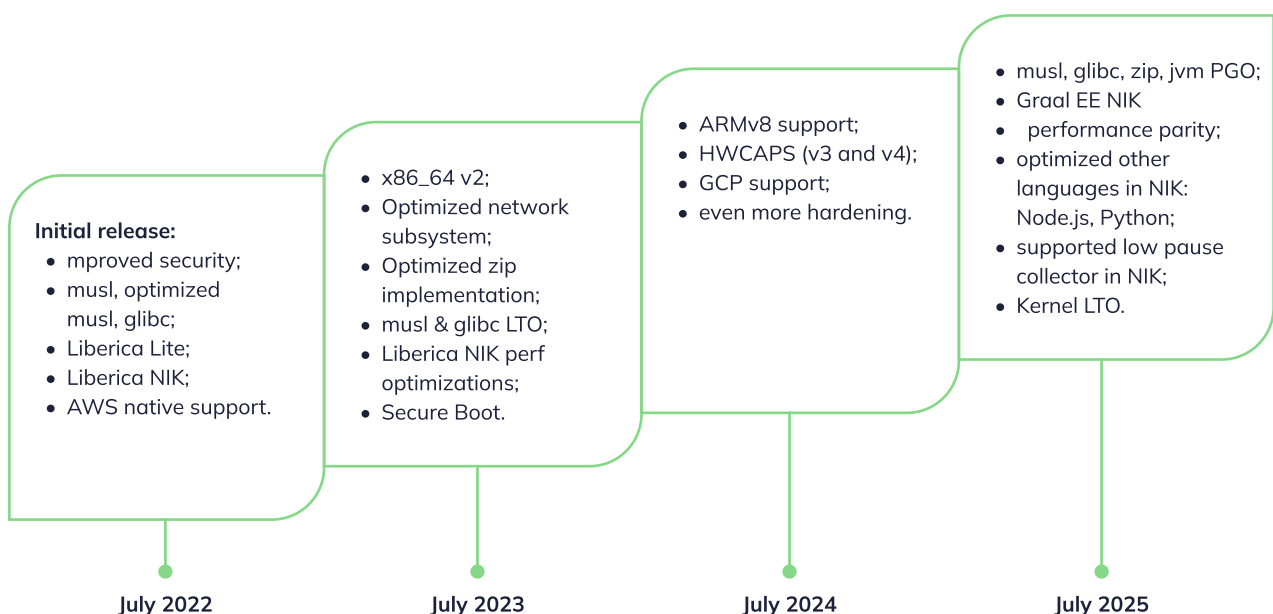
- Alpaquita LTS releases synchronize with Linux Kernel LTS versions; for instance, Alpaquita LTS 22 aligned with Linux Kernel LTS 5.10.
- Each Alpaquita LTS release received support for four years.
- A new LTS version comes out approximately every two years, creating a two-year overlap, during which two versions will be supported simultaneously.
- Alpaquita Stream will continuously receive rolling releases to facilitate the testing of new features and updates intended for LTS releases.

For a visual representation of the support cycle, [please refer to graph:](#)



Future plans

BellSoft is committed to advancing Alpaquita through continuous development, introducing new features with each subsequent release. Our roadmap includes the integration of the following enhancements:



CONTACT US TODAY

Have more questions about Alpaquita Linux, our other products, or enterprise support plans? Our sales representative, Bob Boshehri, will provide you with the assistance you need.

Feel free to reach out to Bob using the contact details below or [schedule a meeting](#) with him.



Bob Booshehri

Java Expert Group

bob@bell-sw.com

+1 (702) 2135959