

# Alpaquita Linux — the new OS for efficient containerized applications

bellsoft



Alpaquita  
Linux

# Introducing Alpaquita Linux

Alpaquita Linux is the OS developed by BellSoft with a goal to create a supported, secure, performant, and convenient deployment environment for Java applications and other use cases.

Alpaquita is based on Alpine Linux and not only maintains its features, but is loaded with a number of improvements.

The essential features of Alpaquita are

- Small size of the base image — Docker image only 3.69 MB
- Application startup time 45% faster than Java analogues (when using the Alpaquita Cloud Native Platform utilities, including Liberica JDK and Liberica Native Image Kit)
- Quality affordable enterprise support by BellSoft team
- Enhanced security with networking features, custom build options, kernel hardening, process isolation, and more
- Maximum performance with kernel optimizations, memory management, and libc implementation optimizations, including improved musl
- Unmatched flexibility with three libc variants (glibc, musl, and optimized musl) and three mallocs

Alpaquita Linux, as the essential part of Alpaquita Cloud Native Platform, is an ideal base OS to build the smallest and most performant containers that reduce your Cloud costs up to 20% immediately. Please refer to the Alpaquita Cloud Native Platform page and whitepaper to find out more, as we focus only on Alpaquita Linux in this document.

# Introducing Alpaquita Linux

## Kernel:

- Always on Linux LTS (5.10 currently)
- Secure Boot, signed modules, disabled modules with CVEs
- Better performance tunables, optimized size (xz)

## LIBC:

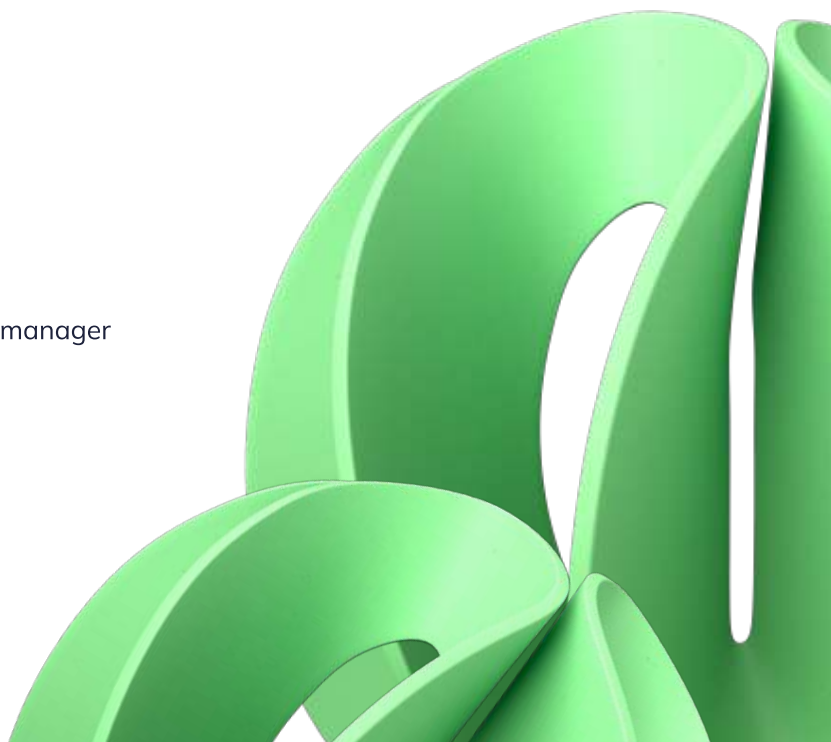
- musl and glibc support, 2 musl variants
- Optimized mallocs

## Userspace based on Alpine Linux aports:

- Busybox, on-demand coreutils
- core/universe repos & improved apk package manager
- Package downgrade support
- No graphics

## Compatibility with other software:

- Docker, QEMU support
- Optimized for Liberica Lite and NIK



# More on libc and glibc



We optimized stock musl to overcome the known limitations of this libc implementation. The performance of our musl perf is equal or superior to that of glibc (see the benchmarks on graphs below). musl-perf is 100% compatible with stock musl, so there won't be any issues with the migration.



We made a glibc-based Alpaquita available, which is still smaller and even more efficient than other glibc-based distributions in some cases (see throughput study results). This way, companies with specific workloads more suitable for glibc-based distros or unwilling to switch libc implementations can also benefit from lightweight and performant Alpaquita OS.

## Alpaquita vs Alpine

Although Alpaquita is based on Alpine, it is full of enhancements that provide a better enterprise environment for development and deployment. Find a comparative overview of Alpaquita and Alpine below:

FEATURE	ALPAQUITA	ALPINE
Commercial Support	Yes, as part of Alpaquita Cloud Native Platform	No
Free community support	Yes, for Stream version	Yes
Open source	Yes	Yes
Free to use	Yes	Yes
glibc support	Yes	No
musl support	musl def musl perf	musl def
License	EULA	GPL2 mostly, MIT for musl, various licenses for software components

You keep the small size of image and get the enterprise-level support for your containers with flexible plans and competitive prices.

Alpine managed to be small in part thanks to musl, a small but not as performant library. Alpaquita offers the faster glibc as an alternative and keeps the size of the base image lightweight.

Even the musl library was perfected by our engineers and in many cases shows better results than default musl.

Multiple licenses may lead to potential law issues. EULA means that we have verified all the packages regarding clean licenses and substituted some software to eliminate the risk of license violation. Therefore, Alpaquita is legally safe to use in enterprises.

FEATURE	ALPAQUITA	ALPINE
Rolling releases	Alpaquita Stream, stable releases	Alpine Edge, releases for testing
Stable releases	LTS releases aligned with Linux Kernel LTS	Stable releases each May and November
Platform compatibility	X86_64, ARM support in development	x86, x86_64, ARMhf, ARMv7, AArch64, ppc64le, s390x
Full Java support	Yes	No
GraalVM support	Yes	No
Ease of migration from other libc	Yes	No
Base image size (as of 12.2022)	3.69 MB	3.25 MB

Alpaquita rolling releases are provided ready to use and tested. Alpine Edge releases are built for testing purposes.

Alpaquita was optimized for Java applications and works perfectly with Liberica JDK and tools by Bellsoft.

GraalVM allows building performant Java native images with better startup times. Usually GraalVM binaries are built for glibc which is not supported in Alpine Linux

Alpine is musl-only, so no glibc migration without additional work is available.

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.

## Alpaquita Linux features explained

### SECURITY

Alpaquita Linux is set up for maximum protection enhanced with LTS releases and security patches released strictly on schedule. Below are some examples of security enhancements.

#### Security — Kernel

- Kernel lockdown enabled including early in boot prevents both direct and indirect access to a running kernel image
- Secure Boot is supported
- 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, for instance, `CONFIG_SECURITY_DMESG_RESTRICT=y` to avoid kernel address exposures for non-root users

## Security — Userspace

- No extra components means low attack surface
- Userspace compilation options are provided for security hardening:
  - **-fPIE, -pie** — position independent executable to protect from memory corruption attacks
  - **-fstack-protector-strong** — enables stack overflow security checks for routines with any type of buffer
  - **-Wformat-security** — warns about possible security problems
  - **-Wtrampolines** — warns when generating trampoline in object that requires executable stack

## 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.

### Kernel Performance (crucial changes)

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

- NUMA options, enabling memory placement with NUMA-aware scheduler as well as memory and taskset
- Task group support
- Improved latency with IOSCHED options for interactive, soft real-life applications
- Improved energy efficiency and reduced OS jitter with NO\_HZ option

This is not the full list of changes (which is much longer), but these we consider the most representative.

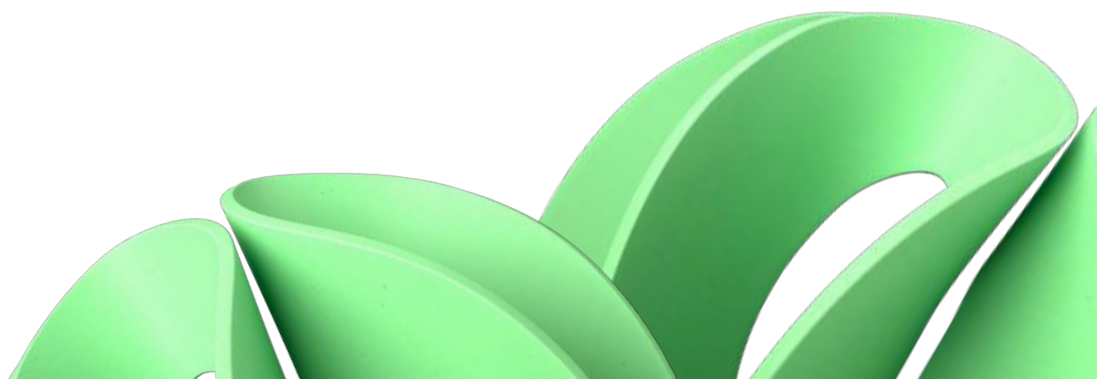
### Optimized musl LIBC

Alpaquita is loaded with optimizations of the default libc library:

- BellSoft musl is built with -O3 for internal, malloc, and string subsystems and -O2 for the rest (Alpine musl uses -Os for all subsystems)
- BellSoft musl supports indirect functions, internally implements CPU features discovering and includes various CPU-specific ASM optimizations (AVX512, EVEX, AVX2, SSE4, etc.)

Optimized musl implementation is 100% compatible with default musl eliminating any migration issues. The default musl package is also delivered. Moreover, glibc can be used in the glibc-based Alpaquita installation as another alternative.

To see the impact of BellSoft musl on performance, consult [graphs part 1](#).



## Malloc implementations

In addition to the malloc utilized in Alpine Linux, Alpaquita includes three other malloc variants you can use for tasks they do best:

- mimalloc is small (128K), efficient, used in large scale low-latency services
- rpmalloc is even smaller (64K) with lock free thread caching
- jemalloc is a time-tested implementation that emphasizes fragmentation avoidance and scalable concurrency support (616K)

To see the test results of using different malloc implementations, consult [graphs part 2](#).

## VERSATILITY OF USE

Alpaquita Linux is an optimal choice both for server and cloud use as it supports virtualization and cloud technologies, so even if you have a hybrid cloud application partially based on physical servers, partially in the cloud, you can use Alpaquita Linux for both scenarios. The small size of the operating system enables the creation of microcontainers or smaller VMs. Base Alpaquita image can be used as is or customized with packages from Alpaquita package repositories. We also provide images with tools and utilities for Java, Python, and C/C++ development.

## CONVENIENCE

We made Alpaquita OS easy to set up and use. We deliver it via many different platforms, including:

- ISO 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, including base image, Liberica Runtime Container, Liberica NIK image, GCC image, and Python image

## SUPPORT

### Licensing

Alpaquita licenses are the same you are used to and contain no “surprises”. EULA will not include any harsh terms that will influence your workflow.

- OSS:
  - Mostly GPL
  - Liberica & NIK under GPLv2+CE
- Full due diligence of licenses for all packages
- Permissive and clear EULA

### Support options

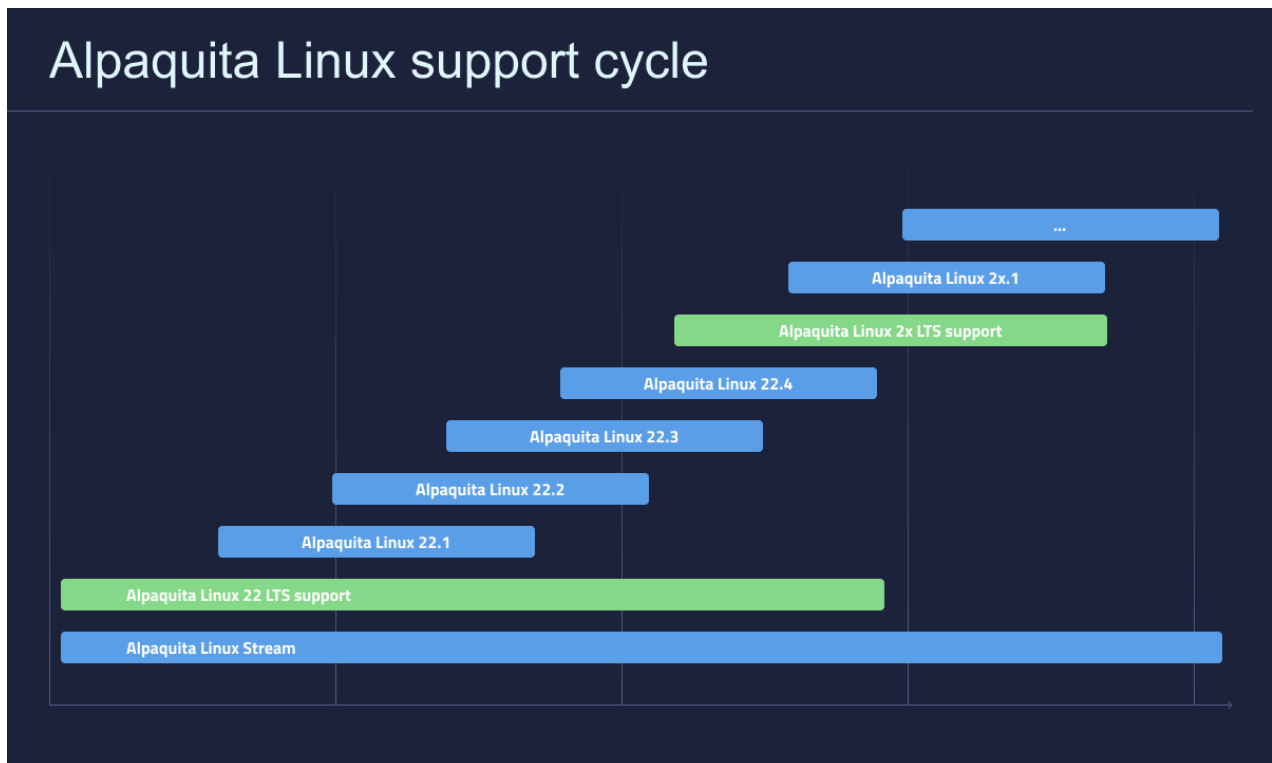
Alpaquita Stream version is free to use and supported by the community. The LTS version of OS is included in the Alpaquita Cloud Native Platform and supported by BellSoft as one of the products in the platform. Please consult the [corresponding page](#) or download the White Paper on Alpaquita Cloud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

## Support cycle

- Alpaquita LTS releases will be aligned with Linux Kernel LTS: 5.10 Kernel LTS for Alpaquita LTS 22
- Alpaquita LTS will be supported for four years, which is two years longer than Alpine
- There will be a 2 years overlap of the previous LTS release with the next one
- Alpaquita Stream will have rolling releases to provide grounds for testing new features and updates for LTS releases

To better understand the support cycle, see the [graph #3](#).

## Graph #3. Alpaquita Linux support cycle



## FUTURE PLANS

Alpaquita is the system that will be developed further, with expanded LTS support and new features added with every new release. We plan to include the following features:

- x86\_64 v2
- Optimized network subsystem
- Optimized zip implementation
- musl & glibc LTO
- LibERICA NIK performance optimizations
- ARMv8 support
- HWCAPS (v3 and v4)
- Kernel LTO
- musl, glibc, zip, jvm PGO
- Graal EE NIK performance parity
- AWS, GCP, Azure support
- optimization of other languages in NIK: Node.js, Python
- supported low pause collector in NIK





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

alpaquita-22-glibc alpaquita-22-musl-def alpaquita-22-musl-perf alpine-3.16  
centos-9 debian-11 redhat-8



Lower is better

**Note:** musl def is the stock musl implementation, musl perf is optimized BellSoft musl implementation

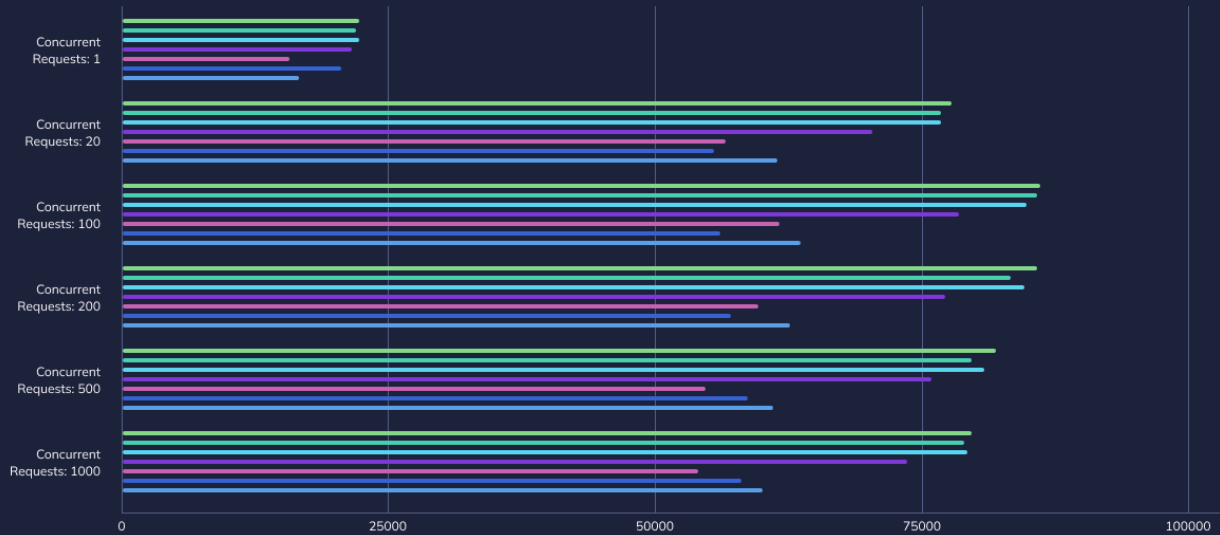
**Tests:** basic functional string tests with 1 million iterations.

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)

alpaquita-22-glibc alpaquita-22-musl-def alpaquita-22-musl-perf alpine-3.16  
centos-9 debian-11 redhat-8



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

## Graphs part 2. Malloc performance results



Results in relative speed 1/s, **higher is better**

malloc benchmarks utilized in the study:

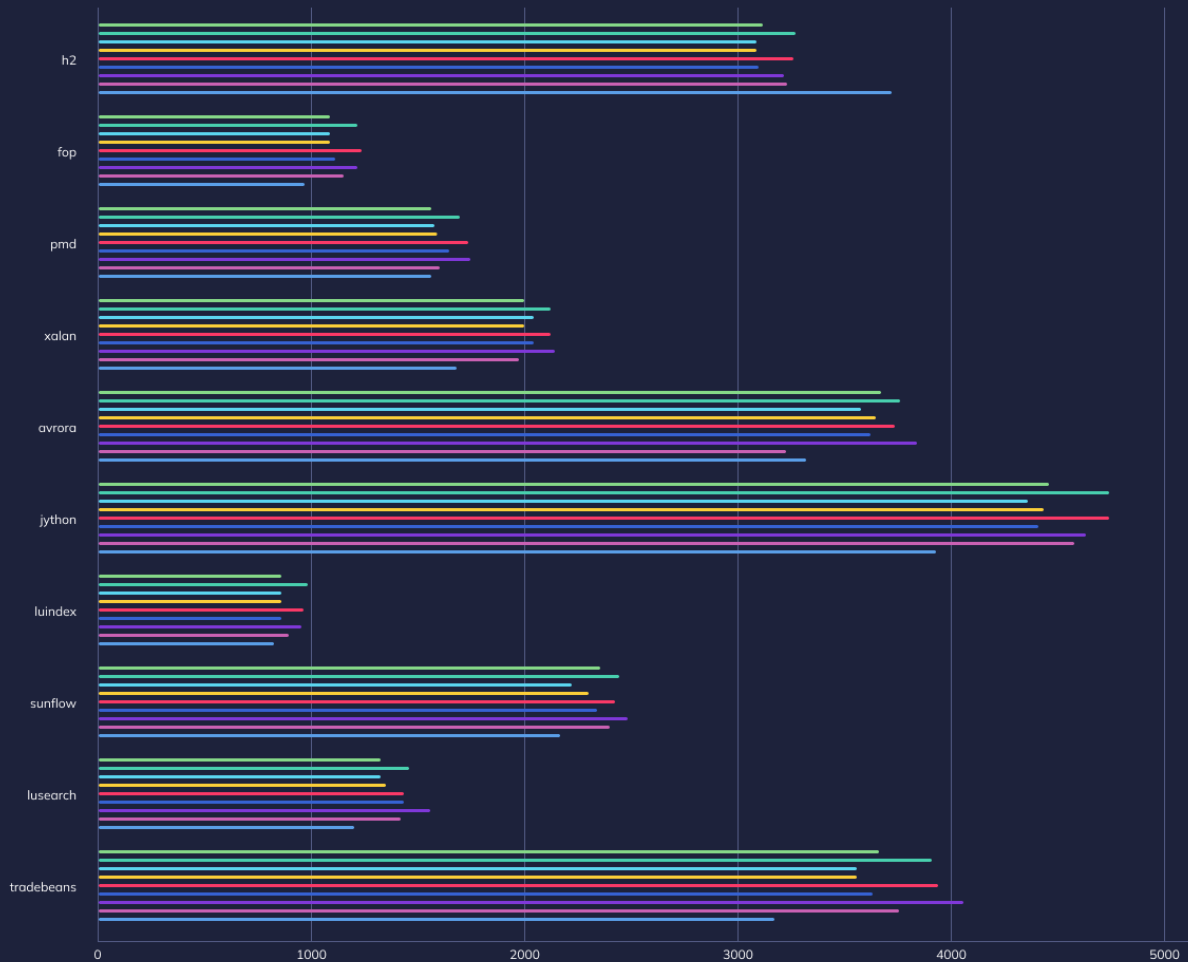
- espresso: a programmable logic array analyzer in the context of cache aware memory allocation
- barnes: a hierarchical n-body particle solver [4], simulating the gravitational forces between 163840 particles
- alloc-test: simulates intensive allocation workloads with a Pareto size distribution
- cache-thrash: 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)

alpaquita-22-glibc alpaquita-22-musl-def alpaquita-22-musl-jpf alpaquita-22-musl-mpf  
alpaquita-22-musl-perf alpaquita-22-musl-rpf alpine-3.16 centos-9 redhat-8



[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



## 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](mailto:bob@bell-sw.com)

+1 (702) 2135959

