

bellsoft

# DEBUG YOUR NATIVE SPRING

Dmitry Chuyko

bell-sw.com | 2024




bellsoft

# WHO WE ARE

Dmitry Chuyko

https://openjdk.org/census#dchuyko

# About BellSoft

bell//soft

BellSoft was founded in 2017 by Java and Linux experts with 15+ years of experience working in Sun/Oracle.
Headquarters in San Jose, California.

Members of:

- JCP Executive Committee
- OpenJDK Vulnerability Group
- GraalVM Advisory Board
- Linux Foundation
- Cloud Native Computing Foundation













GraalVM™

OpenJDK™

# Our products

bellsoft

## Java



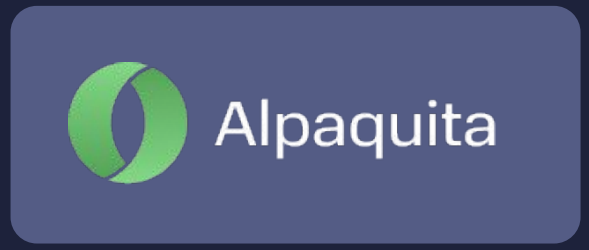

Liberica JDK is a 100% open-source vanilla Java 8, 11, 17 and 21 implementation.




Liberica Native Image Kit JDK is a GraalVM based tool for creating performant native images.

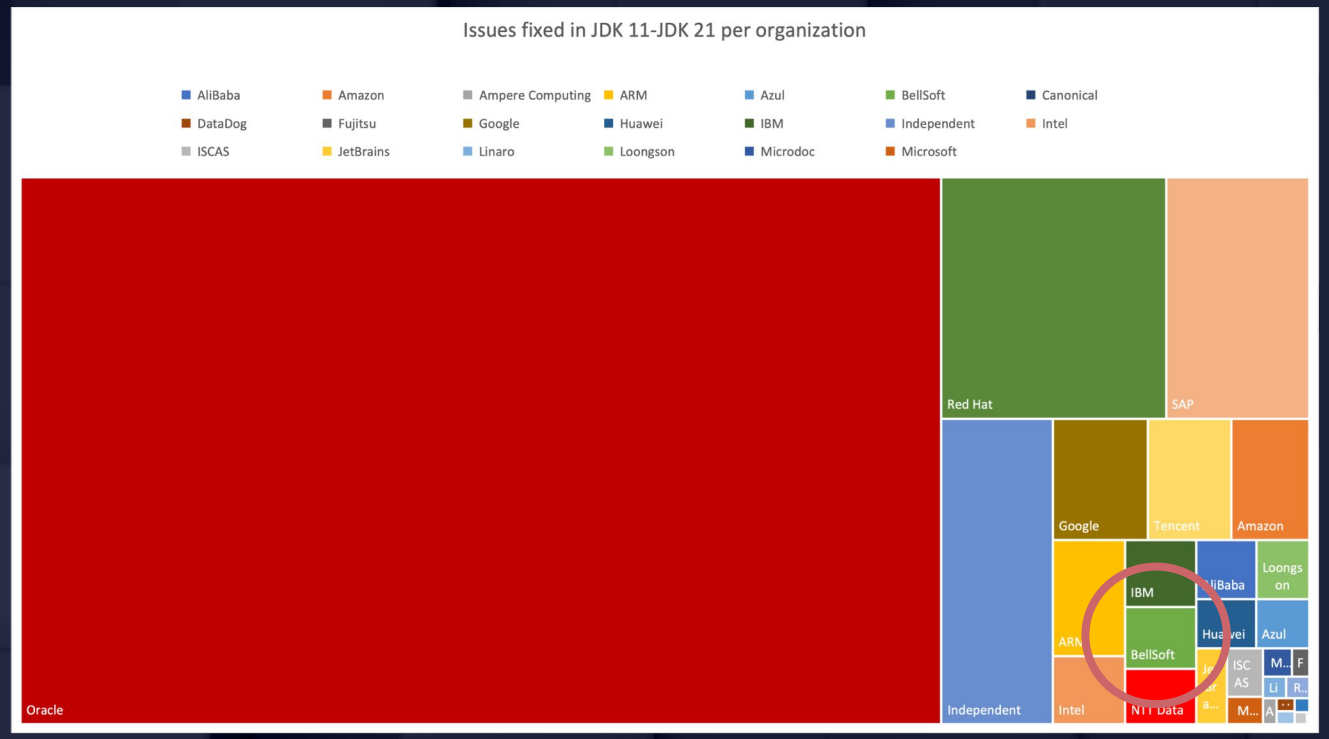
Release schedule for all the products conforms to the LTS roadmap.
All products are available for a large number of platforms.

## Linux




Alpaquita Linux is 100% Alpine compatible, secure, and optimized for Java.

bellsoft




# BellSoft's contributions to OpenJDK

- Leading OpenJDK contributor
- JEP 315 (aarch64 optimization)
- JEP 386 (Alpine Linux port)
- Maintain the upstream Arm port
- Brought musl libc support to GraalVM



# GraalVM Native Image

## Standalone executable

- No OpenJDK runtime
- Closed world assumption
- Special treatment of dynamic features

## Benefits

- Instant warmup
- Lower footprint
- High performance
- Great for the Cloud



# Spring Boot PetClinic

Spring Boot 3.2.1 – supports natives

Liberica NIK 23 (JDK 21)

Fat jar **57 MB**

Native image **166 MB** (release)



# REGULAR JAVA DEBUGGING

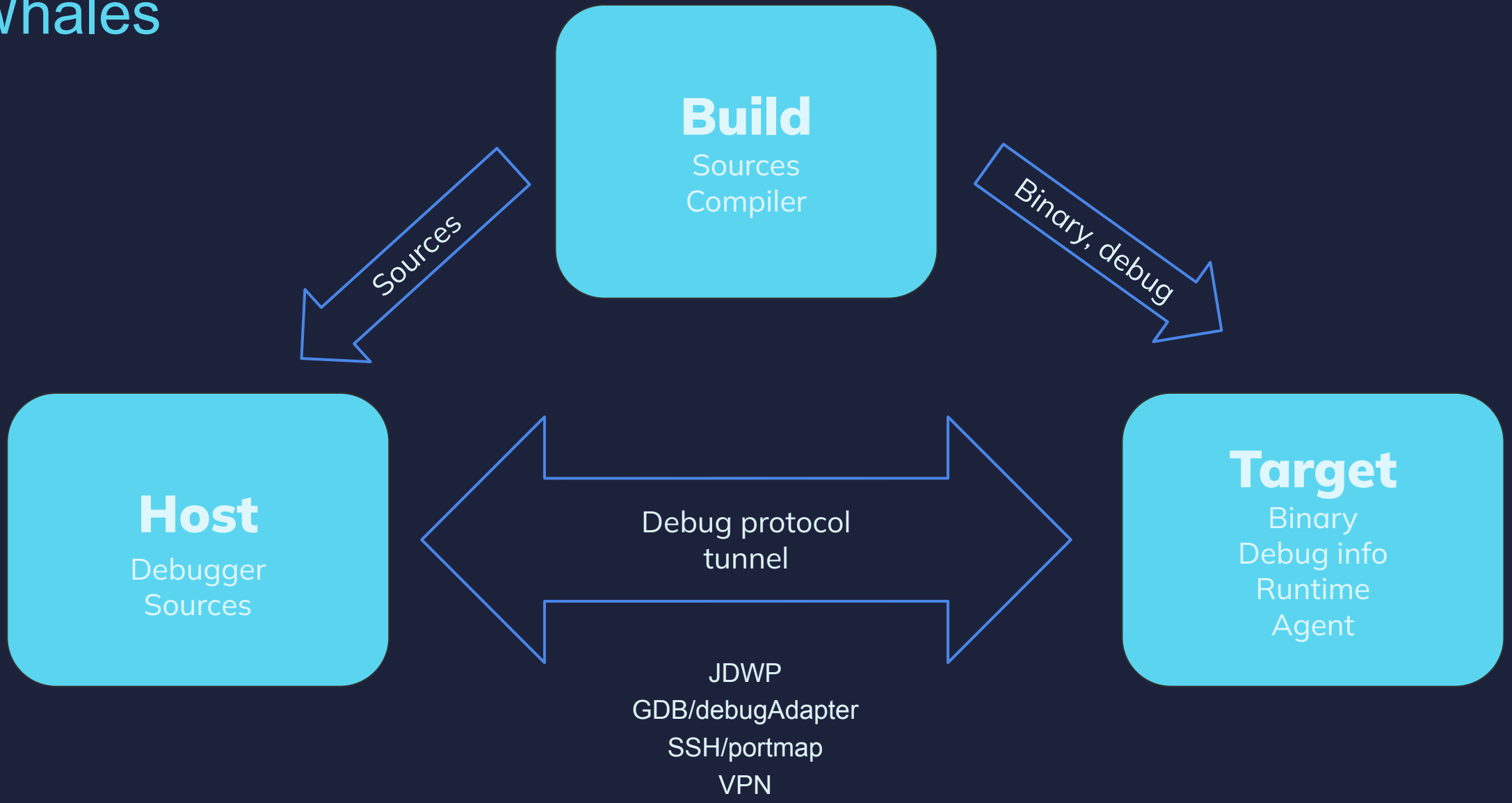
JDB DEMO

# JDB works everywhere. But how?

```
$ java \
-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=8000 \
-Xmx1g -Dserver.port=8081 -jar spring-petclinic-3.2.0-SNAPSHOT.jar

$ jdb -attach 8000
```

# 3 Whales


Build
Sources
Compiler
Sources
Binary, debug
Host
Debugger
Sources
Debug protocol
tunnel
Target
Binary
Debug info
Runtime
Agent
JDWP
GDB/debugAdapter
SSH/portmap
VPN




*actually a slice in .class

# Native Image Debugging

## Currently supported targets:

- Linux: GDB
  - Linux/x86_64 os the best tested one
  - Linux/AArch64 has known issues
- Windows: support is still under development
- MacOS: LLDB
  - Open PR, available in Liberica NIK
  - macOS/x86_64 LLDB & GDB
  - macOS/AArch64 LLDB
  - Based on Linux implementation

# Native Image Debugging

Currently supported features:

- Break points (file:line,  method name)
- Step into, step over
- Stack backtraces (not including frames detailing inlined code)
- Printing of primitives
- Structured (by field) printing of Java objects
- Casting/printing objects at different levels of generality
- Access through object networks via path expressions
- Reference by name to methods, static fields
- Reference by name to args and locals
- Reference by name to class constants

**https://www.graalvm.org/latest/reference-manual/native-image/debugging-and-diagnostics/DebugInfo/**

# Native Image Debugging

`-g (-H:GenerateDebugInfo)`

`-O0 (or -O[N] -H:+SourceLevelDebug )`

`-H:DebugInfoSourceSearchPath`

`-H:DebugInfoSourceCacheRoot (relative to -H:Path)`

`-H:-DeleteLocalSymbols`




MAC
MACH-O & LLDB DEMO

# LLDB works locally

```
...
% sudo xattr -r -d com.apple.quarantine \
bellsoft-liberica-vm-core-openjdk21.0.3+10-23.1.3+1-macos-amd64.tar.gz
...

% JAVA_HOME=~/bellsoft-liberica-vm-core-openjdk21-23.1.3/Contents/Home
./mvnw -Pnative clean native:compile

212 MB release, 509 MB debug

% ./spring-petclinic

% codesign lldb … OR

% sudo lldb
```


LINUX
GDB CONTAINER DEMO

# GDB works remotely

```
$ docker run -d -i --name spring-debug -m 1g -p 8083:8080 -p 2345:2345 \
  --cap-add=SYS_PTRACE \
  --security-opt seccomp=unconfined --security-opt apparmor=unconfined \
  bellsoft/alpaquita-linux-base:stream-musl

...

# apk add libstdc++ freetype gdb libstdc++-dev freetype-dev
...
# tar -xvf \
 bellsoft-liberica-vm-core-openjdk21.0.3+10-23.1.3+1-linux-x64-musl.tar.gz
...
```

# GDB works remotely. Tired of command line?

```
# ./mvnw dependency:list

# ./mvnw dependency:build-classpath \
  -Dmdep.pathSeparator="," -Dmdep.outputFile=deps

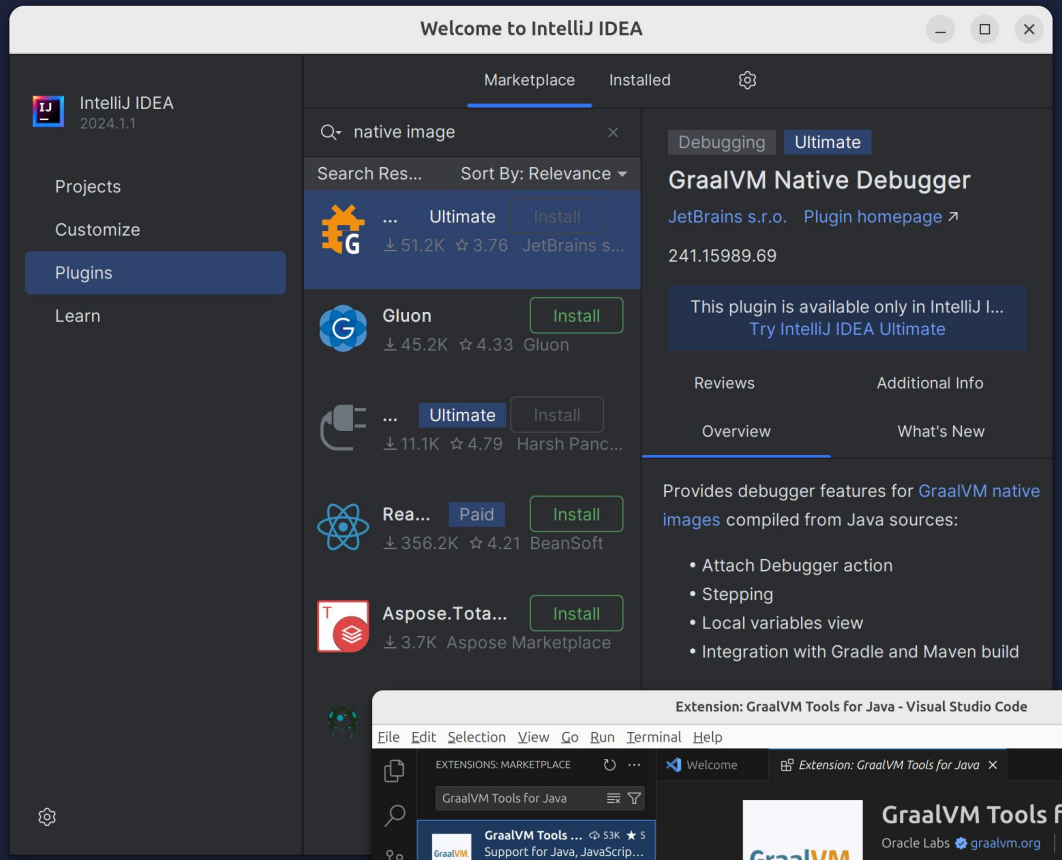
# sed -i 's/.jar,/-sources.jar,/g' deps

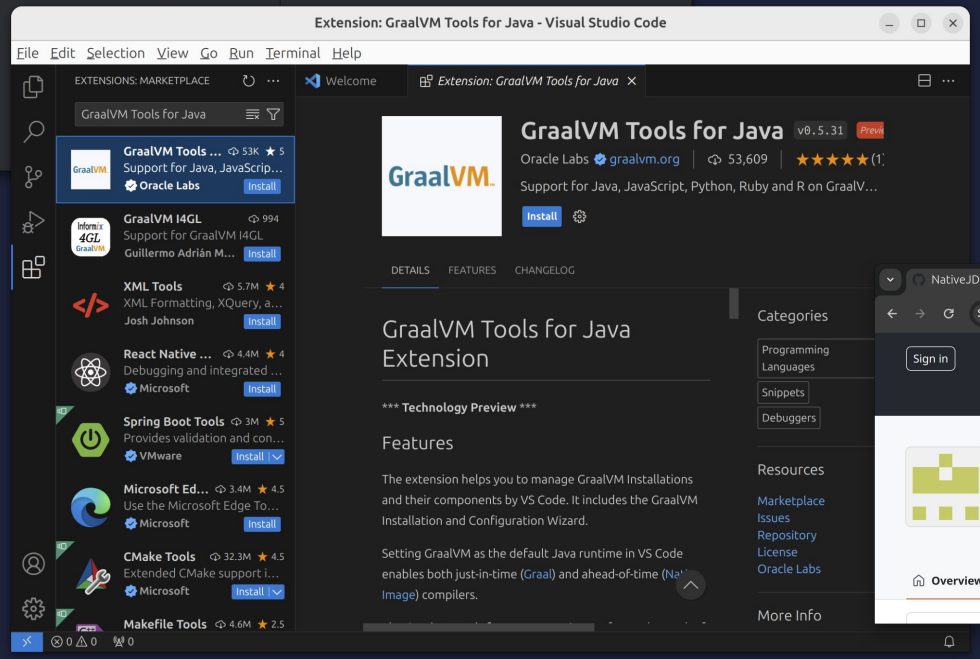
-H:DebugInfoSourceSearchPath

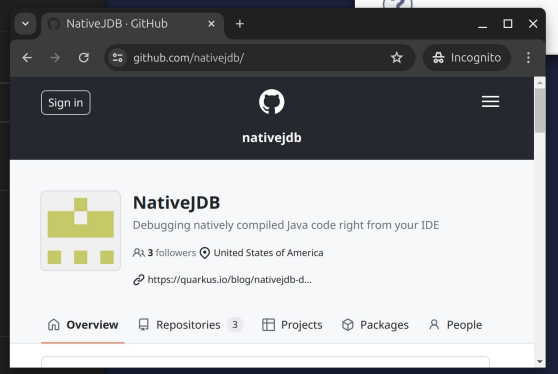
# gdbserver --attach :2345 ${pid}

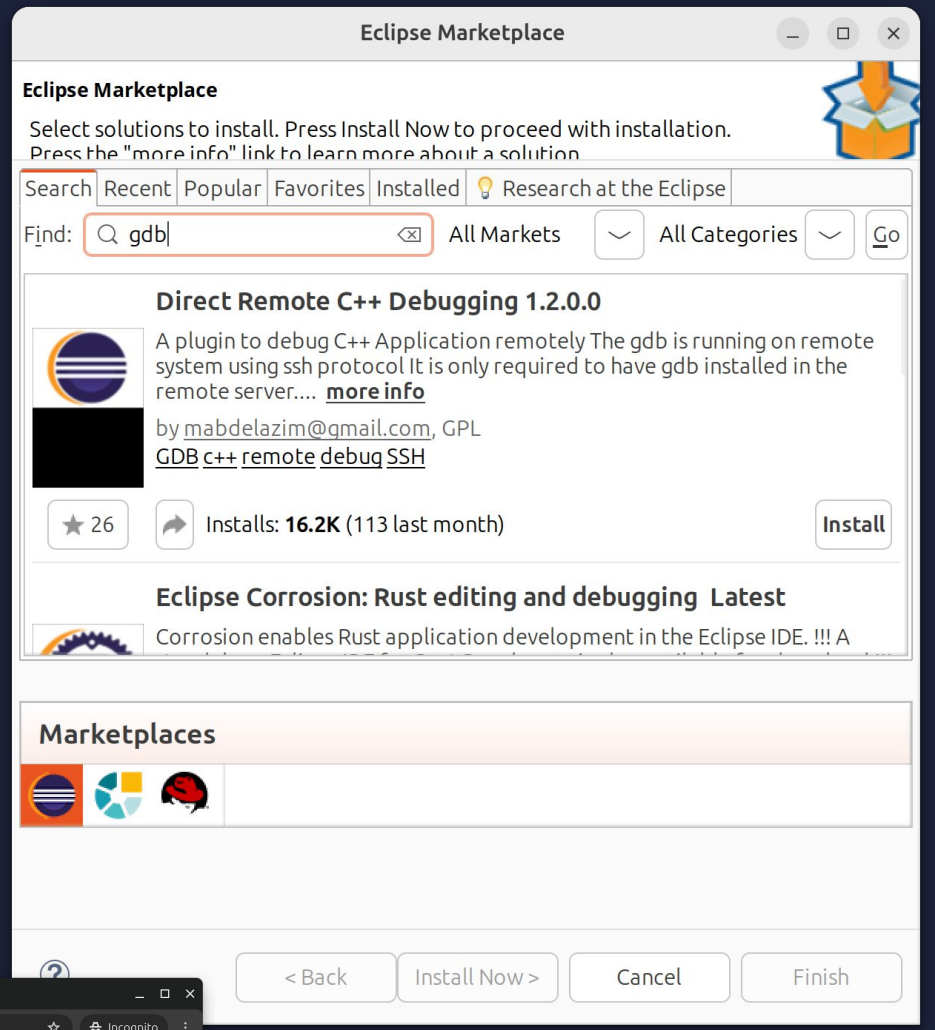
kubectl port-forward

(gdb) target remote 127.0.0.1:2345
```


Welcome to IntelliJ IDEA
IntelliJ IDEA
2024.1.1
Projects
Customize
Plugins
Learn
Marketplace
Installed
native image
Search Res...
Sort By: Relevance
... Ultimate Install
51.2K 3.76 JetBrains s...
Gluon Install
45.2K 4.33 Gluon
... Ultimate Install
11.1K 4.79 Harsh Panc...
Rea... Paid Install
356.2K 4.21 BeanSoft
Aspose.Tota... Install
3.7K Aspose Marketplace
Debugging Ultimate
GraalVM Native Debugger
JetBrains s.r.o. Plugin homepage
241.15989.69
This plugin is available only in IntelliJ I...
Try IntelliJ IDEA Ultimate
Reviews
Additional Info
Overview
What's New
Provides debugger features for GraalVM native images compiled from Java sources:
• Attach Debugger action
• Stepping
• Local variables view
• Integration with Gradle and Maven build


Extension: GraalVM Tools for Java - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXTENSIONS: MARKETPLACE
GraalVM Tools for Java
GraalVM Tools ... 53K 5
Support for Java, JavaScrip...
Oracle Labs Install
GraalVM I4GL 994
Support for GraalVM I4GL
Guillermo Adrián M... Install
XML Tools 5.7M 4
XML Formatting, XQuery, a...
Josh Johnson Install
React Native ... 4.4M 4
Debugging and integrated ...
Microsoft Install
Spring Boot Tools 3M 5
Provides validation and con...
VMware Install
Microsoft Ed... 3.4M 4.5
Use the Microsoft Edge To...
Microsoft Install
CMake Tools 32.3M 4.5
Extended CMake support i...
Microsoft Install
Makefile Tools 4.6M 2.5
Welcome
Extension: GraalVM Tools for Java
GraalVM Tools for Java v0.5.31
Oracle Labs graalvm.org 53,609
Support for Java, JavaScript, Python, Ruby and R on GraalV...
Install
DETAILS FEATURES CHANGELOG
GraalVM Tools for Java Extension
*** Technology Preview ***
Features
The extension helps you to manage GraalVM Installations and their components by VS Code. It includes the GraalVM Installation and Configuration Wizard.
Setting GraalVM as the default Java runtime in VS Code enables both just-in-time (Graal) and ahead-of-time (Native Image) compilers.
Categories
Programming Languages
Snippets
Debuggers
Resources
Marketplace
Issues
Repository
License
Oracle Labs
More Info


NativeJDB · GitHub
github.com/nativejdb/
Incognito
Sign in
nativejdb
NativeJDB
Debugging natively compiled Java code right from your IDE
3 followers United States of America
https://quarkus.io/blog/nativejdb-d...
Overview
Repositories 3
Projects
Packages
People


Eclipse Marketplace
Eclipse Marketplace
Select solutions to install. Press Install Now to proceed with installation.
Press the "more info" link to learn more about a solution
Search
Recent
Popular
Favorites
Installed
Research at the Eclipse
Find: gdb
All Markets
All Categories
Go
Direct Remote C++ Debugging 1.2.0.0
A plugin to debug C++ Application remotely The gdb is running on remote system using ssh protocol It is only required to have gdb installed in the remote server.... more info
by mabdelazim@gmail.com, GPL
GDB c++ remote debug SSH
26
Installs: 16.2K (113 last month)
Install
Eclipse Corrosion: Rust editing and debugging Latest
Corrosion enables Rust application development in the Eclipse IDE. !!! A
Marketplaces
< Back
Install Now >
Cancel
Finish


IDE
REMOTE GDB ECLIPSE DEMO

# IDE works remotely

## Eclipse

- Install plugin (and CDT)
- Add C and C++ project natures to the Java project
- Debug
  - C/C++ Remote
  - GDB Manual Remote debugging
  - Project: our dual-nature project
  - Program: target binary
  - Debugger/Connection port: 2345
  - Source
    - New Workspace folder: target/sources
    - Path mapping: build -> host

# Conclusions

bellsoft

## Try at home

- Native image debug is crucial
- Experimental implementation available
- Essential platforms are covered
- Essential features are covered
- Java mimics C++, tools are reused
- IDE support exists

## Call us

- Developers are busy
- Runtime and compiler bugs exist
- Provide info and access

bell//soft

# Thank you for your attention!




bell-sw.com/blog

@dchuyko


bellsoft
Q & A