**INSTALL**

Download [GraalVM JDK](https://github.com/graalvm/graalvm-jdk) and install Native Image. The quickest way to get started is to run:

```bash
bash <(curl -sL https://get.graalvm.org/jdk)
```

**BUILD**

Use `bin/native-image` in the same way you would use `bin/java`. Here are some common examples:

- for a JAR:
  ```
  native-image -jar Foo.jar
  ```
- for a module:
  ```
  native-image -p Foo.jar = org.foo.Main
  ```
- for a class:
  ```
  native-image -cp Foo.jar org.foo.Main
  ```

**BUILD OPTIONS**

Enable the quick build mode for development:

```bash
-0
```

Specify a name for the resulting binary:

```bash
-o <myapp>
```

Build a shared library:

```bash
--shared
```

Build a statically linked binary with libc implementation:

```bash
--static --lib=clib | musl | bionic
```

Enable a specific set of URL protocols:

```bash
--enable-url-protocols=http | https | file | resource | JAR
```

Specify a list of packages and classes to be linked at build time:

```bash
--link-at-build-time=<comma-separated list of packages and classes>
```

Specify a garbage collector:

```bash
--gc=serial | G1 | epsilon
```

Provide java.lang.Terminator exit handlers:

```bash
--install-exit-handlers
```

**INTEGRATIONS**

Use the **Gradle** plugin for GraalVM Native Image:


Use the **Maven** plugin for GraalVM Native Image:


Use the **GitHub Action** for GraalVM:

[https://github.com/marketplace/actions/github-action-for-graalvm](https://github.com/marketplace/actions/github-action-for-graalvm)

**CONFIGURE**

There are three ways to configure Native Image for external Java libraries:

1. Configuration is provided by a framework such as [Micronaut](https://micronaut.io), [Helidon](https://helidon.dev), and [Quarkus](https://quarkus.io)
2. Configuration is automatically pulled in by the Maven/Gradle plugin from [GraalVM Reachability Metadata Repository](https://graalvm.github.io/reachability-metadatarepository/)
3. Configuration is provided manually. The Tracing Agent can help in this process.

**MONITOR**

Tune the garbage collector:

```bash
./myapp -Xmx<m> -Xms<m> ...
```

Gather garbage collection logs:

```bash
./myapp -XX:+PrintGC -XX:+VerboseGC
```

**PREREQUISITE**

Enable VM inspection when building a native executable with:

```bash
--enable-monitoring=<all, heapdump, jfr, jvmsstat> . Defaults to all.
```

**DEBUG**

Build a native executable with debug information and with compiler optimizations disabled:

```bash
-g -00
```

**JFR SUPPORT**

Dump the initial heap of a native executable:

```bash
./myapp -XX:+DumpHeapAndExit
```

**DOCUMENTATION**

* Available with GraalVM Enterprise Native Image

More info at [www.graalvm.org/native-image](http://www.graalvm.org/native-image)