Introduction to Gradle

Who am I?

Donát Csikós
Software Engineer
Gradle Inc.
Java development & tooling

Table of Contents

- Use-case
- Gradle core features
- IDE support
- Gradle Cloud Services

CERN

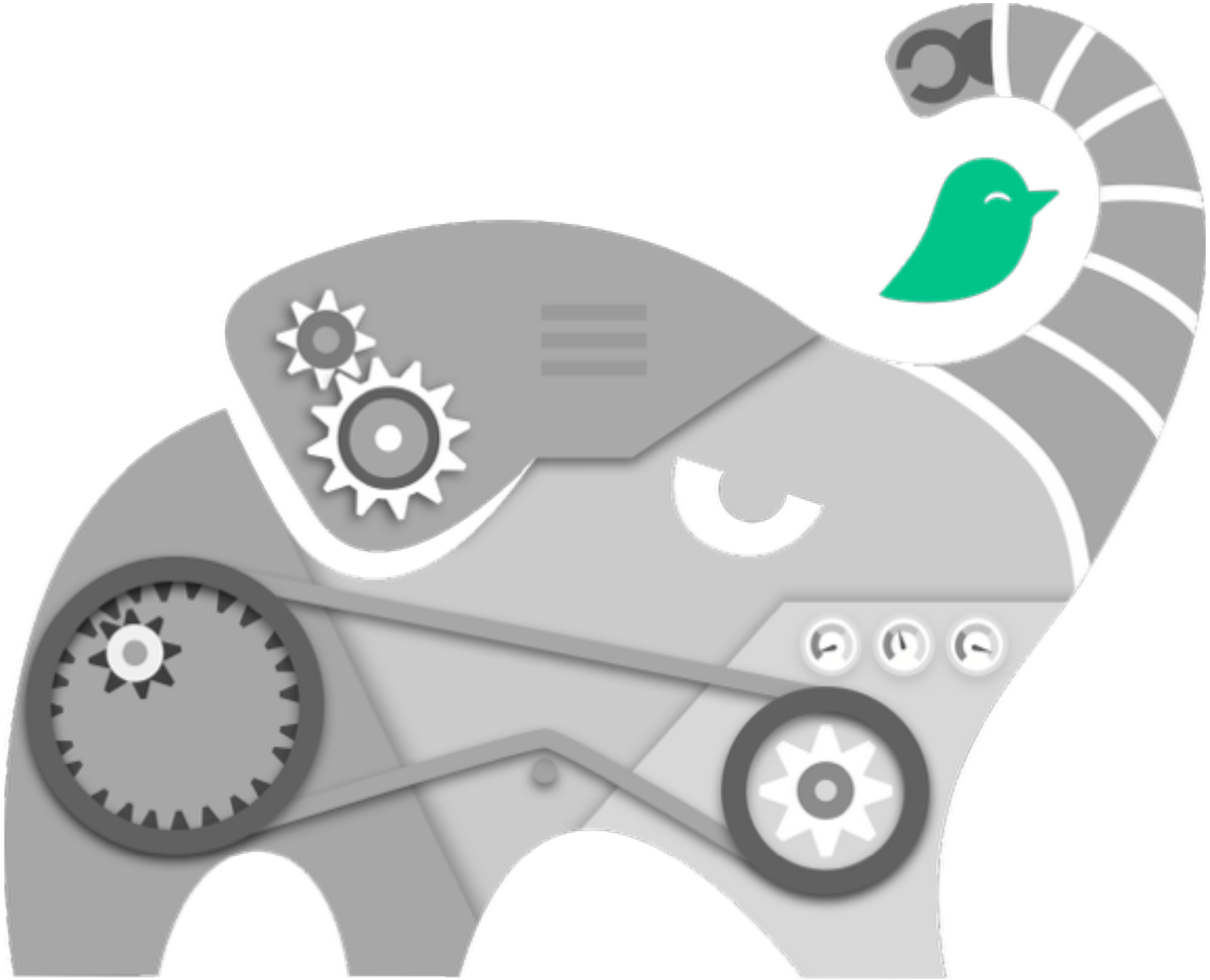
Use-case

In-house build system

- CommonBuild
  - Based on Ant
  - Central release repository
  - C++ and Java support
  - Dynamic versions
- Issues: maintenance, scalability and reproducibility

Migration

- Goal: smooth upgrades
- Failed attempt with Maven
- Successful attempt with Gradle



Gradle core features

The Gradle Build Tool

“The Gradle Build Tool is a modern open source polyglot build automation system”

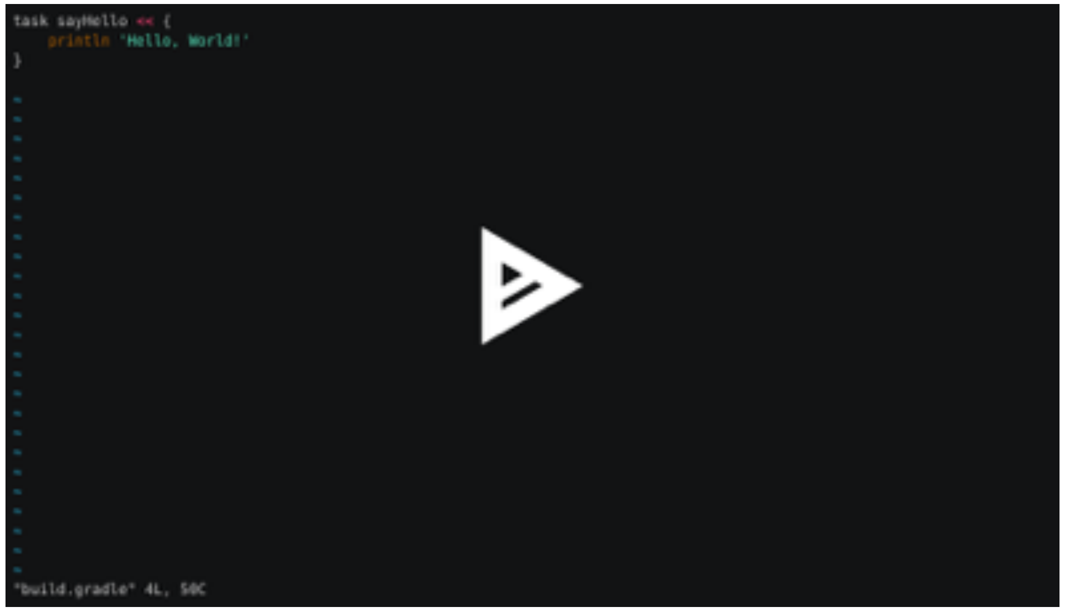
- Like Maven, but…
- No restrictions, only conventions
- Enabler for continuous delivery

Build definition

- `build.gradle` script
  - Groovy language + Gradle DSL
- Resources
  - User guide
  - DSL reference

Task definition

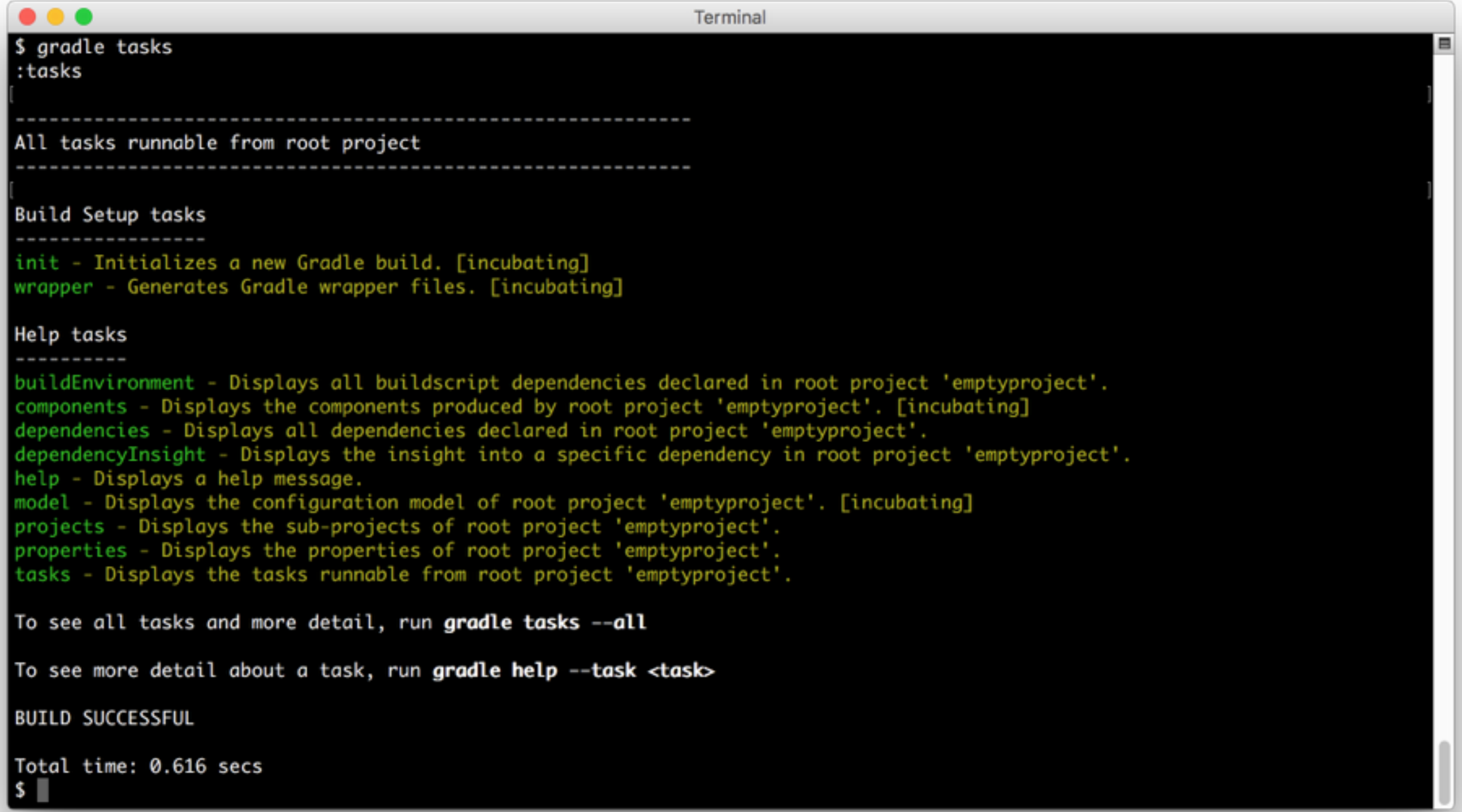
```
task sayHello() << {
    println "Hello, World!"
}
```



```
task sayHello << {
    println 'Hello, World!'
}
"build.gradle" 4L, 58C
```

- Task = single unit of work
- Backed by an org.gradle.api.Project instance
- << is shorthand for the Task.doLast(Closure) method

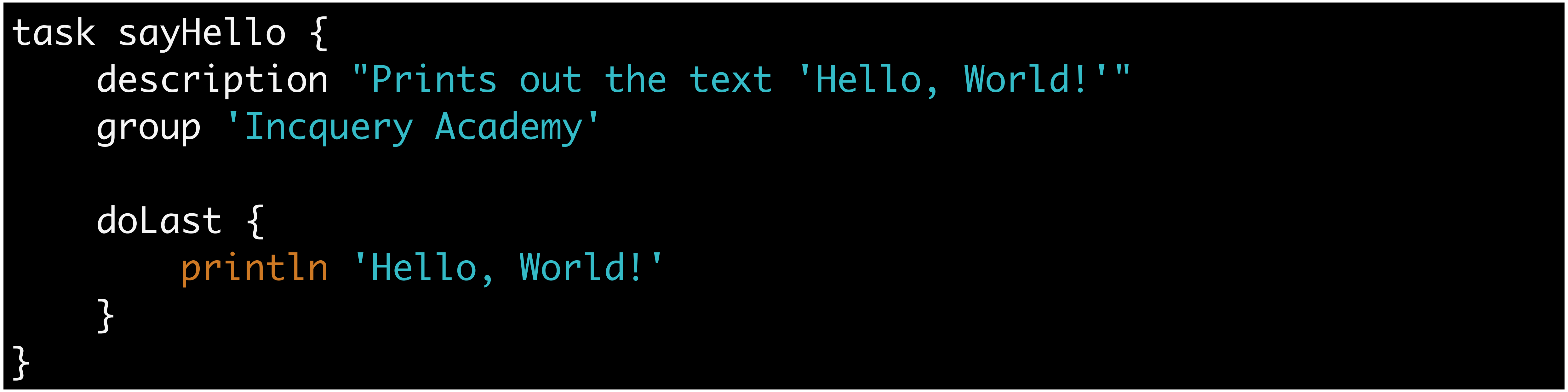
Built-in tasks



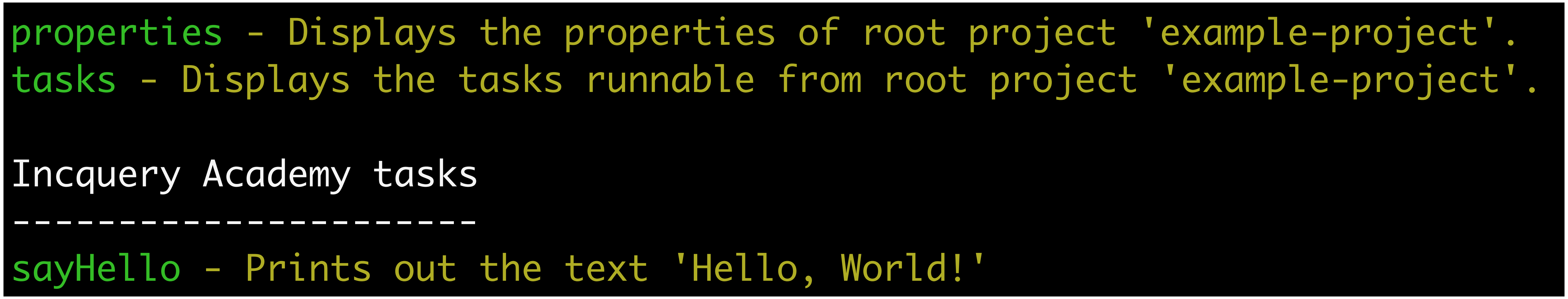
```
$ gradle tasks
:tasks

------------------------------------------------------------
All tasks runnable from root project
------------------------------------------------------------

Build Setup tasks
-----------------
init - Initializes a new Gradle build. [incubating]
wrapper - Generates Gradle wrapper files. [incubating]

Help tasks
----------
buildEnvironment - Displays all buildscript dependencies declared in root project 'emptyproject'.
components - Displays the components produced by root project 'emptyproject'. [incubating]
dependencies - Displays all dependencies declared in root project 'emptyproject'.
dependencyInsight - Displays the insight into a specific dependency in root project 'emptyproject'.
help - Displays a help message.
model - Displays the configuration model of root project 'emptyproject'. [incubating]
projects - Displays the sub-projects of root project 'emptyproject'.
properties - Displays the properties of root project 'emptyproject'.
tasks - Displays the tasks runnable from root project 'emptyproject'.

To see all tasks and more detail, run gradle tasks --all

To see more detail about a task, run gradle help --task <task>

BUILD SUCCESSFUL

Total time: 0.616 secs
$
```

Task properties

- Group and description: visual aid to organise tasks



```
task sayHello {
    description "Prints out the text 'Hello, World!'"
    group 'Incquery Academy'

    doLast {
        println 'Hello, World!'
    }
}
```

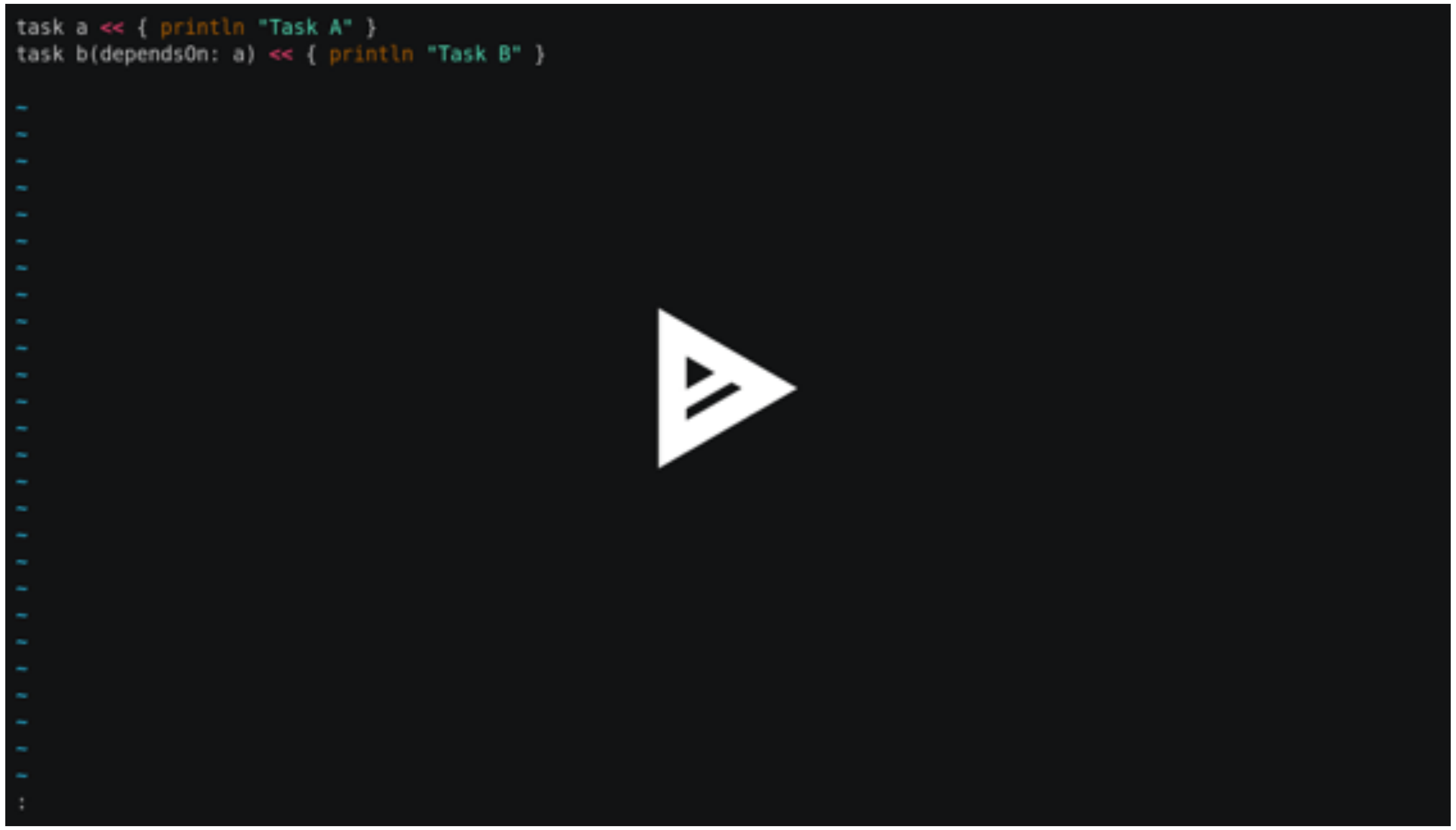


```
properties - Displays the properties of root project 'example-project'.
tasks - Displays the tasks runnable from root project 'example-project'.

Incquery Academy tasks
----------------------
sayHello - Prints out the text 'Hello, World!'
```

Task properties

- Dependencies

```
task a << { println "Task A" }
task b(dependsOn: a) << { println "Task B" }
```



```
task a << { println "Task A" }
task b(dependsOn: a) << { println "Task B" }

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:
```

Task types

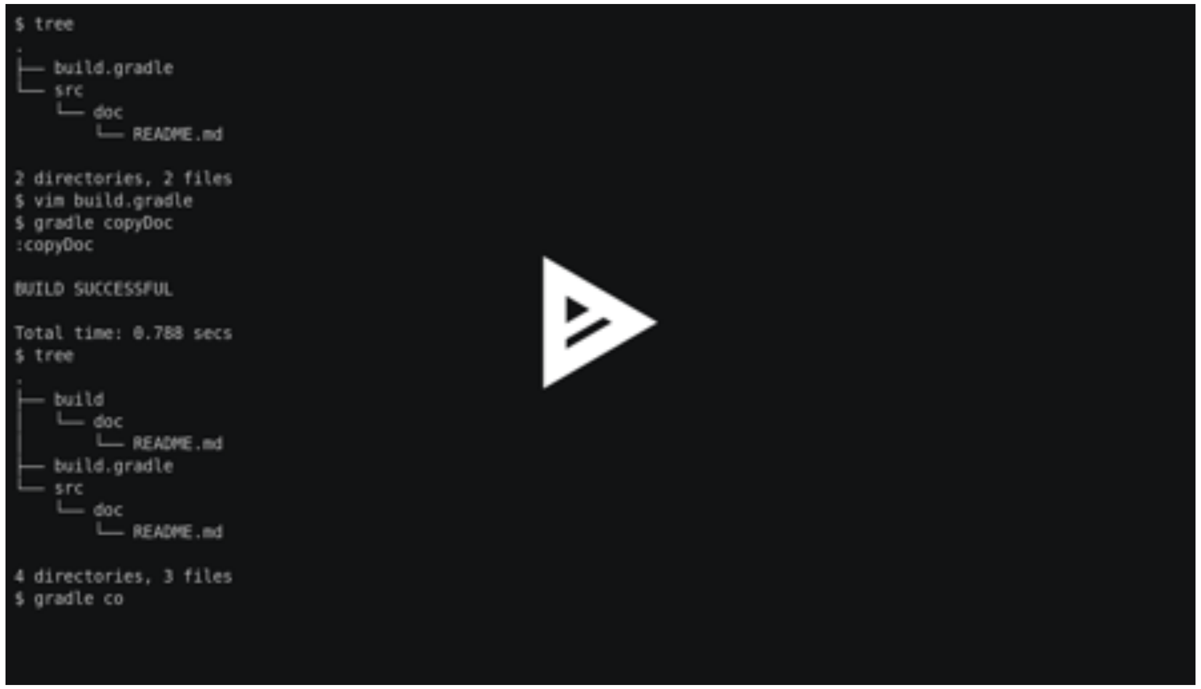
- Each tasks have task types
- `DefaultTask` is used if not defined
- Built-in tasks: `Copy`, `Exec`, `Zip`, `Jar`, etc
    - Listed in the DSL reference
- Can implement a custom one

```
task copyDoc(type: Copy) {
    from 'src/doc'
    into 'build/doc'
}
```

Inputs and Outputs

- Tasks can declare their input and output files and folders
- Task execution is skipped if task is ‘up-to-date’
  - Same input
  - Untouched output

```
task copyDoc(type: Copy) {
    from 'src/doc'
    into 'build/doc'
}
```



```
$ tree
.
├── build.gradle
└── src
    └── doc
        └── README.md

2 directories, 2 files
$ vim build.gradle
$ gradle copyDoc
:copyDoc

BUILD SUCCESSFUL

Total time: 0.788 secs
$ tree
.
├── build
│   └── doc
│       └── README.md
├── build.gradle
└── src
    └── doc
        └── README.md

4 directories, 3 files
$ gradle co
```

Plugins

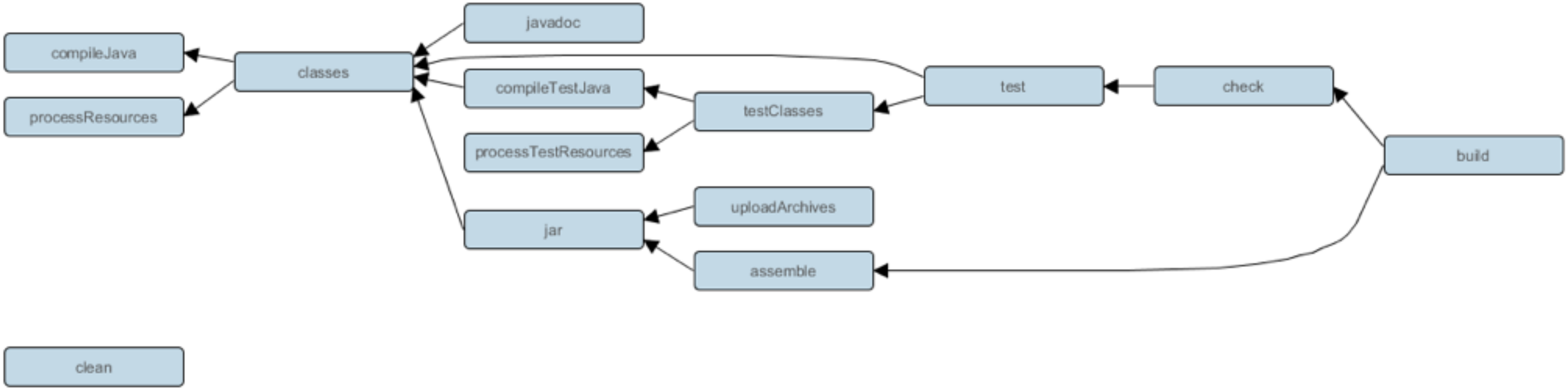
- Plugins contribute tasks, domain objects, conventions and object extensions

- Gradle core functionality is already organised into plugins

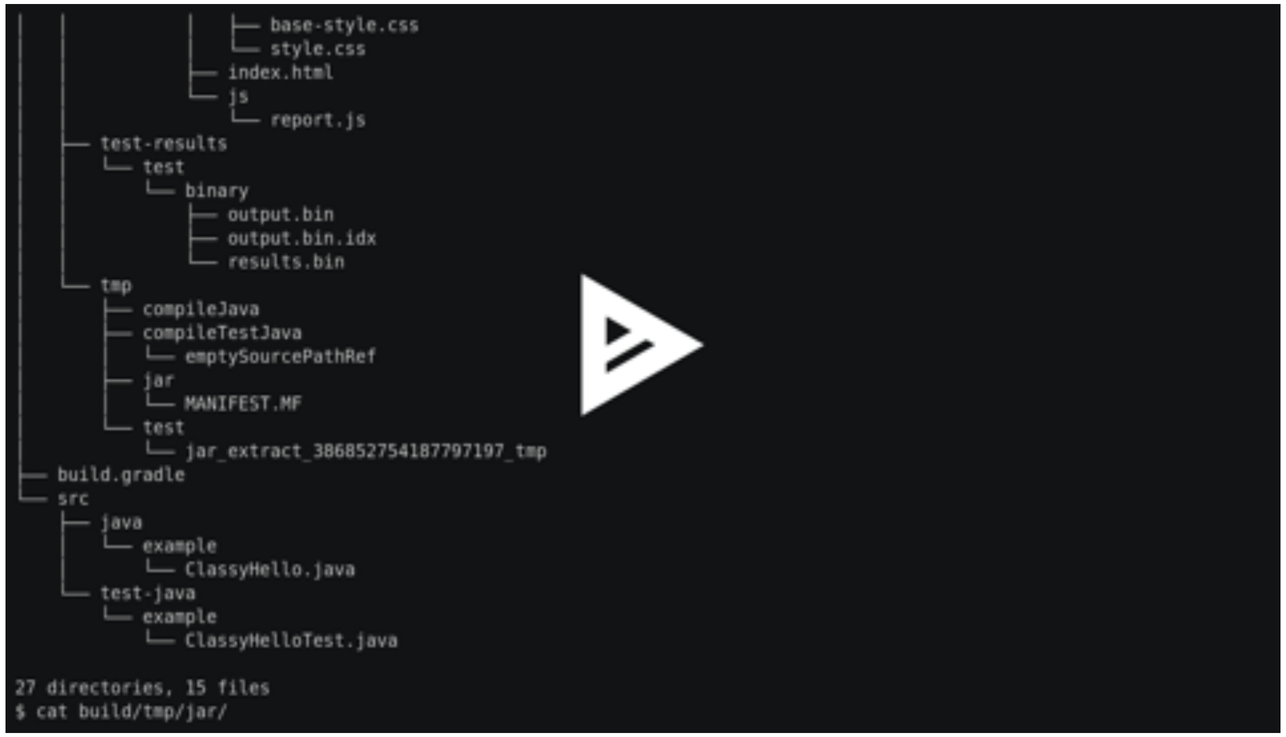
- ~800 external plugins are available in the Gradle plugin portal

Java plugin

```
apply plugin: 'java'
```



Java plugin - project customization

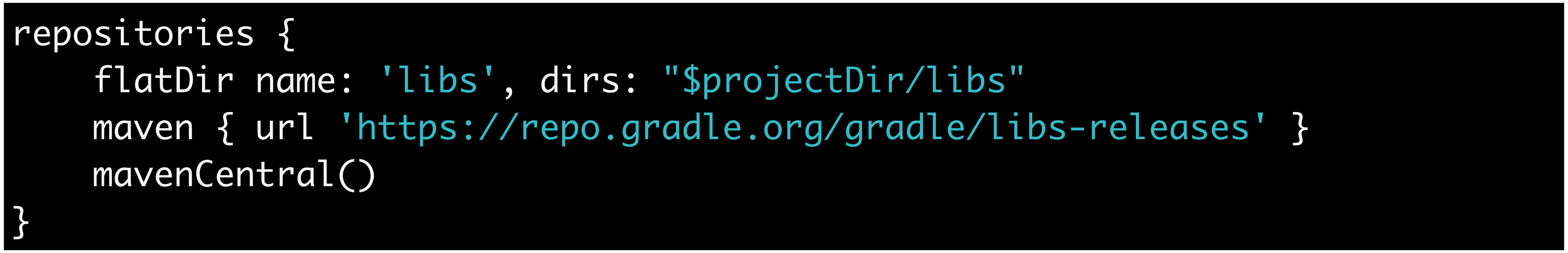


```
│   │           │   ├── base-style.css
│   │           │   └── style.css
│   │           ├── index.html
│   │           └── js
│   │               └── report.js
│   ├── test-results
│   │   └── test
│   │       └── binary
│   │           ├── output.bin
│   │           ├── output.bin.idx
│   │           └── results.bin
│   └── tmp
│       ├── compileJava
│       ├── compileTestJava
│       │   └── emptySourcePathRef
│       ├── jar
│       │   └── MANIFEST.MF
│       └── test
│           └── jar_extract_386852754187797197_tmp
├── build.gradle
└── src
    ├── java
    │   └── example
    │       └── ClassyHello.java
    └── test-java
        └── example
            └── ClassyHelloTest.java

27 directories, 15 files
$ cat build/tmp/jar/
```

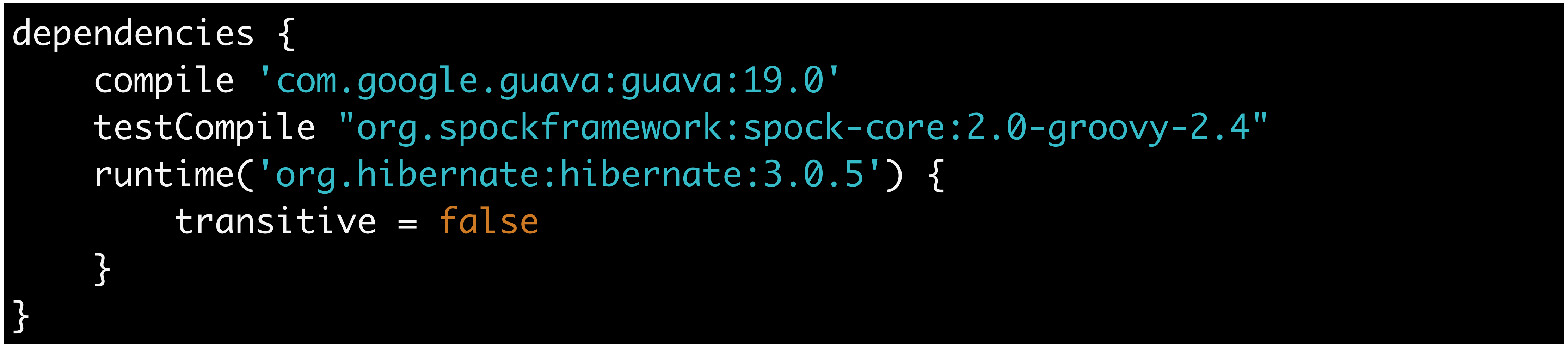
Dependency management

- Transitive dependency management
- Support for non-managed dependencies
- Customisable dependency resolution
- Integration with existing dependency management infrastructure

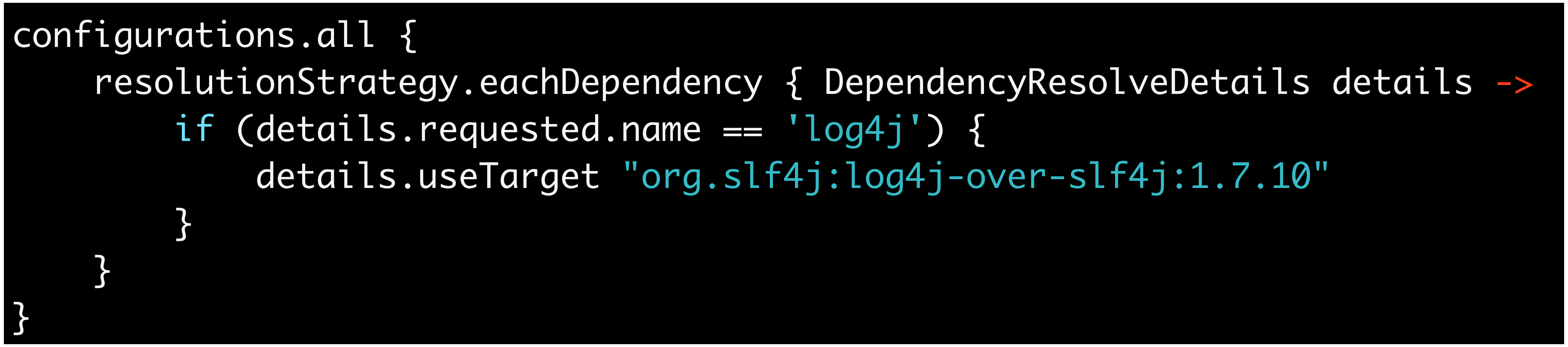
Dependency management



```
repositories {
    flatDir name: 'libs', dirs: "$projectDir/libs"
    maven { url 'https://repo.gradle.org/gradle/libs-releases' }
    mavenCentral()
}
```



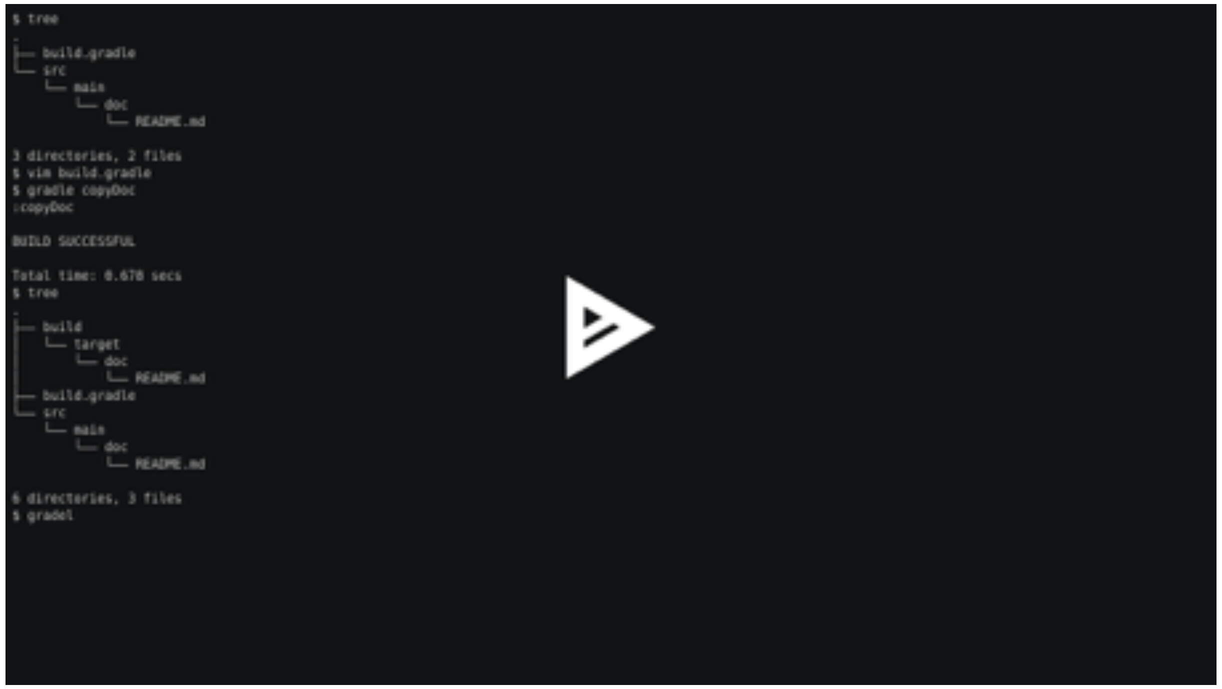
```
dependencies {
    compile 'com.google.guava:guava:19.0'
    testCompile "org.spockframework:spock-core:2.0-groovy-2.4"
    runtime('org.hibernate:hibernate:3.0.5') {
        transitive = false
    }
}
```



```
configurations.all {
    resolutionStrategy.eachDependency { DependencyResolveDetails details ->
        if (details.requested.name == 'log4j') {
            details.useTarget "org.slf4j:log4j-over-slf4j:1.7.10"
        }
    }
}
```

Gradle wrapper

- A script to download and execute Gradle
- It can be checked into VCS along with the project
- Build users don't have to download the Gradle to compile the projects
- Upgrading Gradle is transparent to the build users



```
$ tree
.
├── build.gradle
└── src
    └── main
        └── doc
            └── README.md

3 directories, 2 files
$ vim build.gradle
$ gradle copyDoc
:copyDoc

BUILD SUCCESSFUL

Total time: 0.678 secs
$ tree
.
├── build
│   └── target
│       └── doc
│           └── README.md
├── build.gradle
└── src
    └── main
        └── doc
            └── README.md

6 directories, 3 files
$ gradel
```

IDE integration

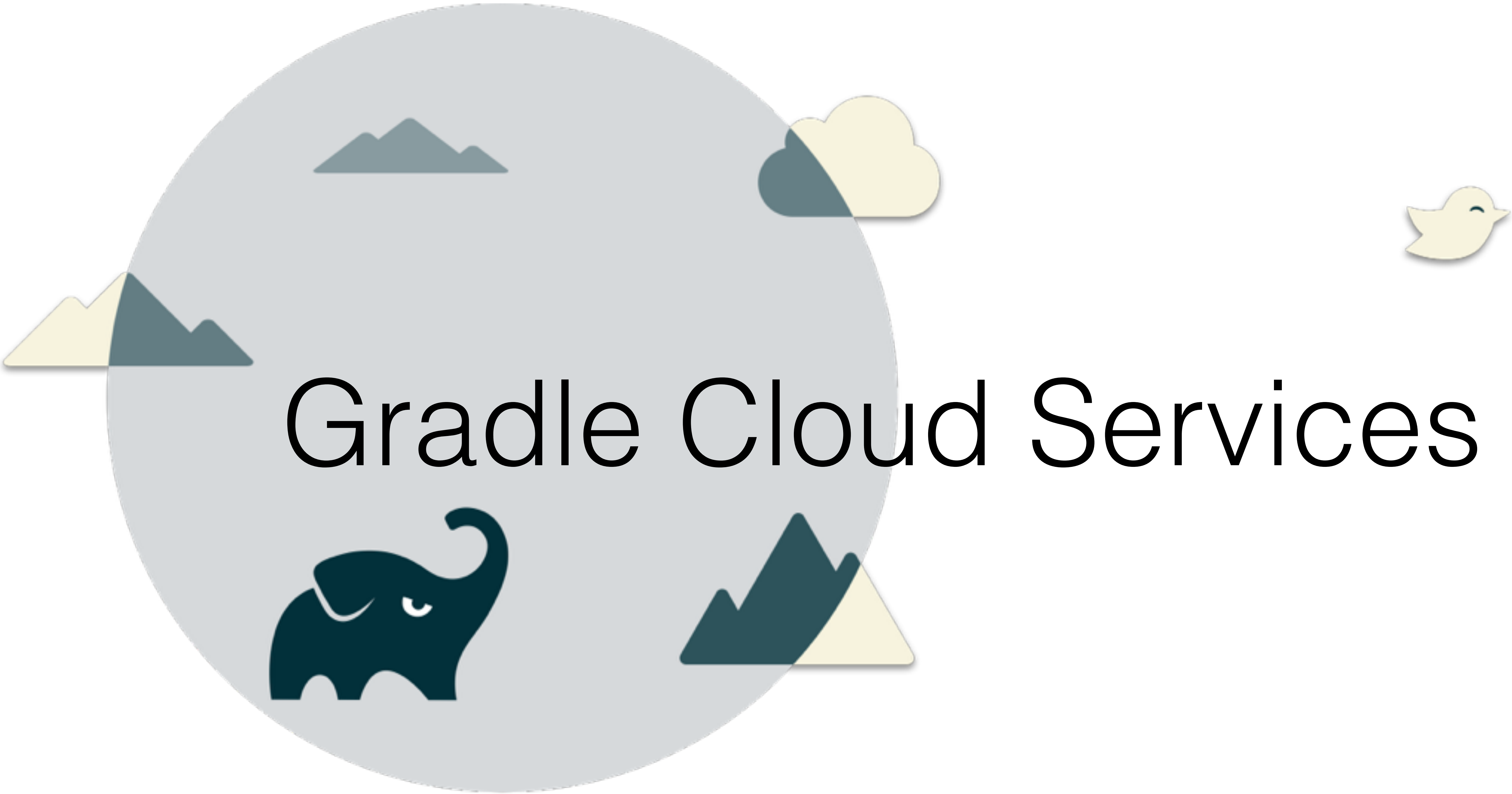
Buildship: Eclipse plugins for Gradle

- Developed by Gradle Inc.
    - My main focus :)
- Top-level Eclipse project
- Pre-installed in 3 packages since Eclipse 4.5
- Vision: single source of build logic

Buildship hands-on

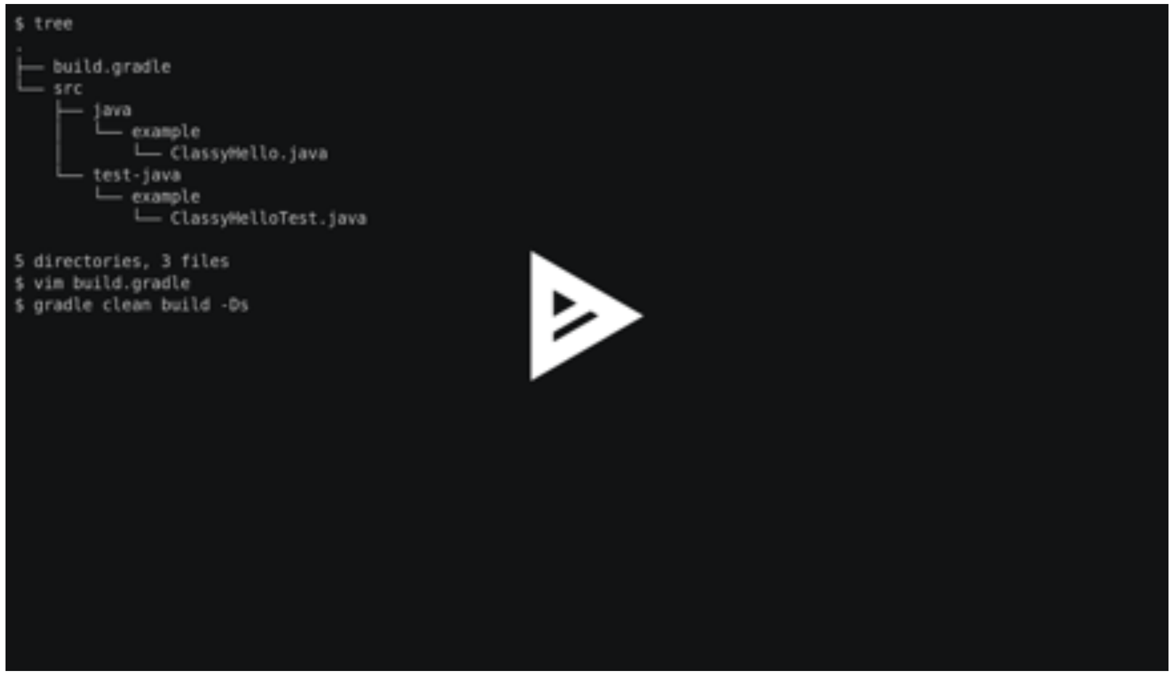
- Project import and synchronisation
- Tasks and Executions view
- Test execution

Gradle Cloud Services



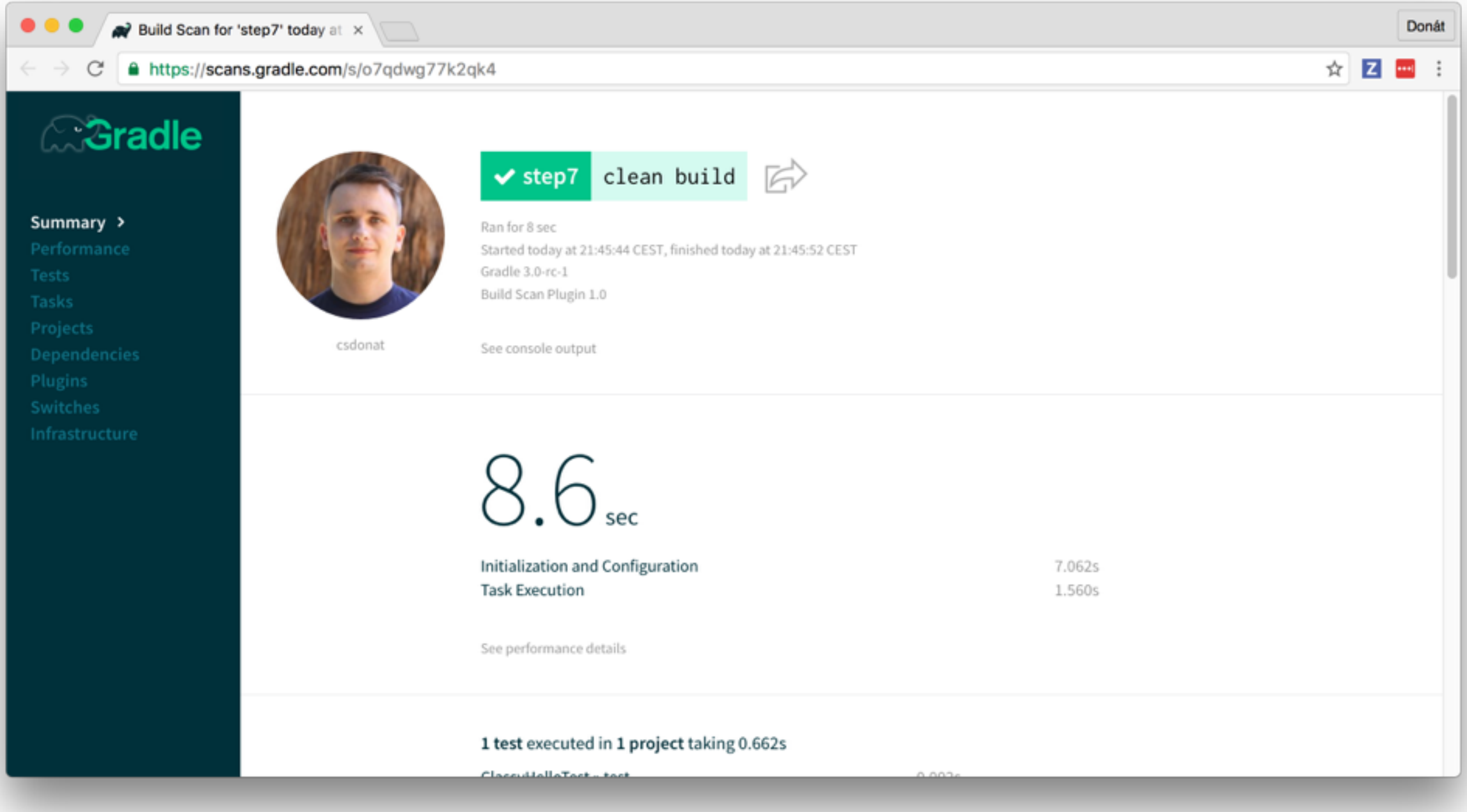
Gradle Cloud Services

- Gradle = build tool + cloud services
- Build scans: shareable build insights



```
$ tree
.
├── build.gradle
└── src
    ├── java
    │   └── example
    │       └── ClassyHello.java
    └── test-java
        └── example
            └── ClassyHelloTest.java

5 directories, 3 files
$ vim build.gradle
$ gradle clean build -Ds
```

Build scans



Ran for 8 sec
Started today at 21:45:44 CEST, finished today at 21:45:52 CEST
Gradle 3.0-rc-1
Build Scan Plugin 1.0

See console output

8.6 sec

Initialization and Configuration 7.062s
Task Execution 1.560s

See performance details

1 test executed in **1 project** taking 0.662s

There’s so much more

- Publishing artifacts
- Gradle Kotlin script
- Composite builds
- Gradle daemon
- Writing plugins and existing plugin ecosystem (Netflix OSS)
- Java 9 support and compiler avoidance
- TestKit: build script testing
- Multi-project builds
- Native builds
- Framework x support (play, android)
- Task cache
- Continuous build

Get in touch with us

- Gradle forums: discuss.gradle.org
- Buildship Bugzilla: https://bugs.eclipse.org/bugs/buglist.cgi?product=Buildship
- Contact me: donat@gradle.com or follow @donatcsikos