Maximize developer productivity

With fast and reliable builds

Labs & slides

https://gradl.es/mdpw

Schedule

- 10min Introduction
- 35min Workshop
- 10min Pause ☕
- 50min Workshop
- 15min Summary

System Requirements

- JDK 8
- JAVA_HOME environment variable pointing to JDK 8
- IDE or text editor of your choice

Setup

Gradle

- Open a command prompt
- Move to workshop folder
- Move to `setup/gradle`
- Run a build to download all dependencies
  `./gradlew clean build`

Maven

- Open a command prompt
- Move to workshop folder
- Move to `setup/maven`
- Run a build to download all dependencies
  `./mvnw clean verify`

In many industries, there are engineering disciplines dedicated to the practice of making production efficient and effective, such as chemical engineering and industrial or process engineering.

Build engineering is of similar importance when it comes to the production of software but is not yet professionalized.

Current harmful patterns

- Anecdote-driven prioritization
- Build engineering is prioritized by complaints, not data
- No vision

How we fix it

1. Within organizations
    - More resources (people, tools)
    - Organizational/cultural changes
2. As an industry
    - Vendors need to provide better solutions
    - Software manufacturing as a discipline

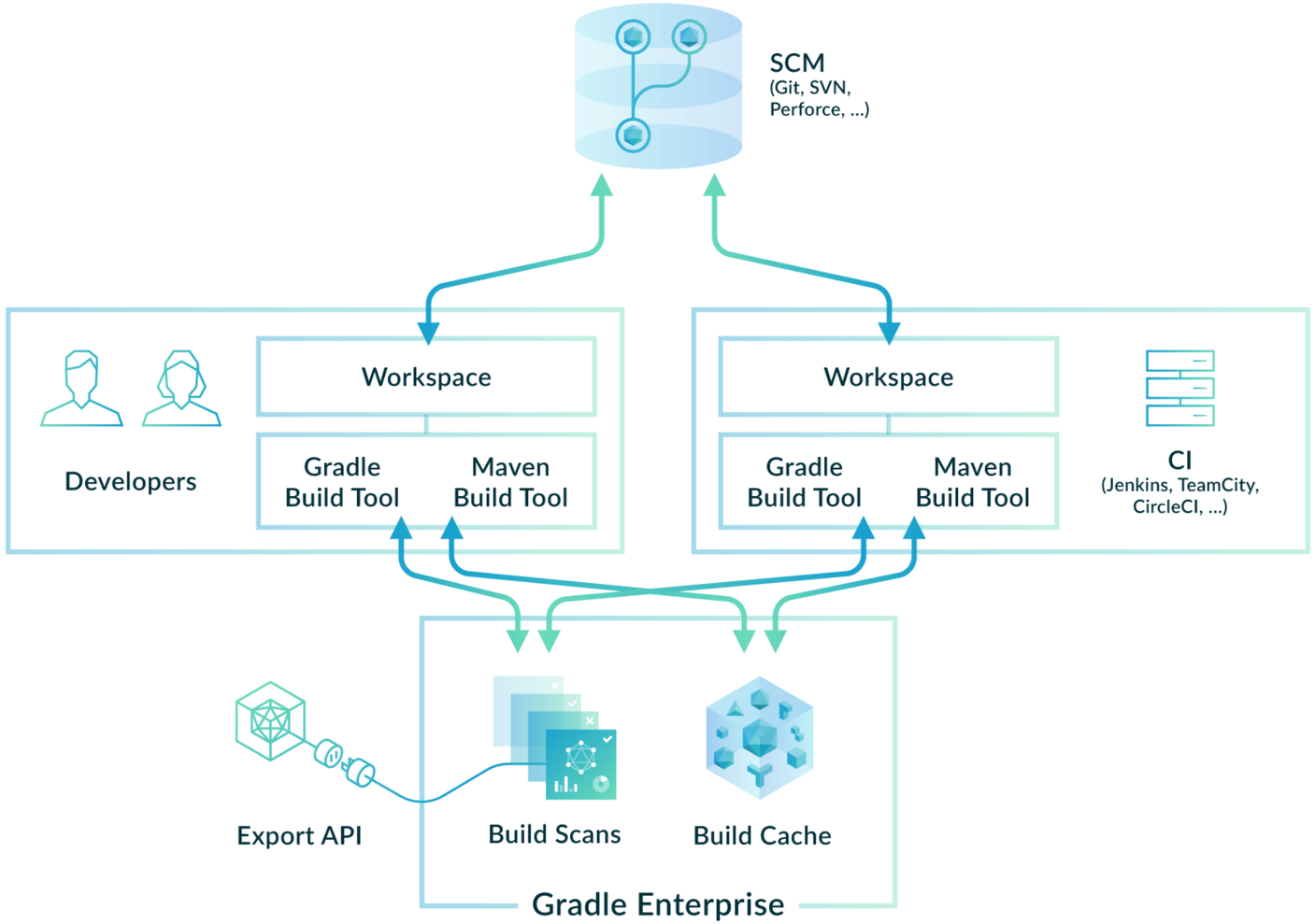
How vendors can help

- Faster builds (caching, distributed builds)
- Build analytics
  - Optimize machinery
  - Optimize behavior/collaboration
- Faster mean time to repair

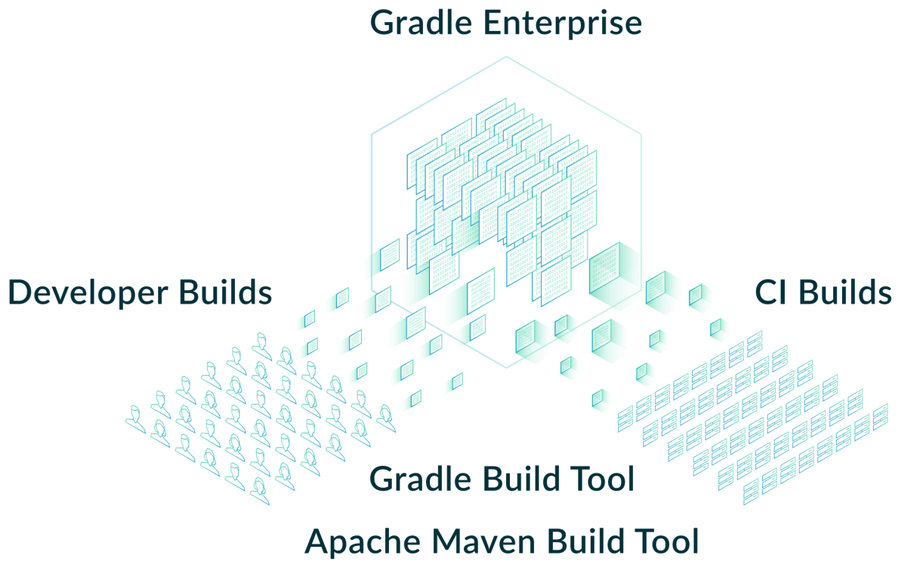
Build automation

Development productivity
(Gradle & Maven)

What is Gradle Enterprise



Gradle Enterprise Benefits

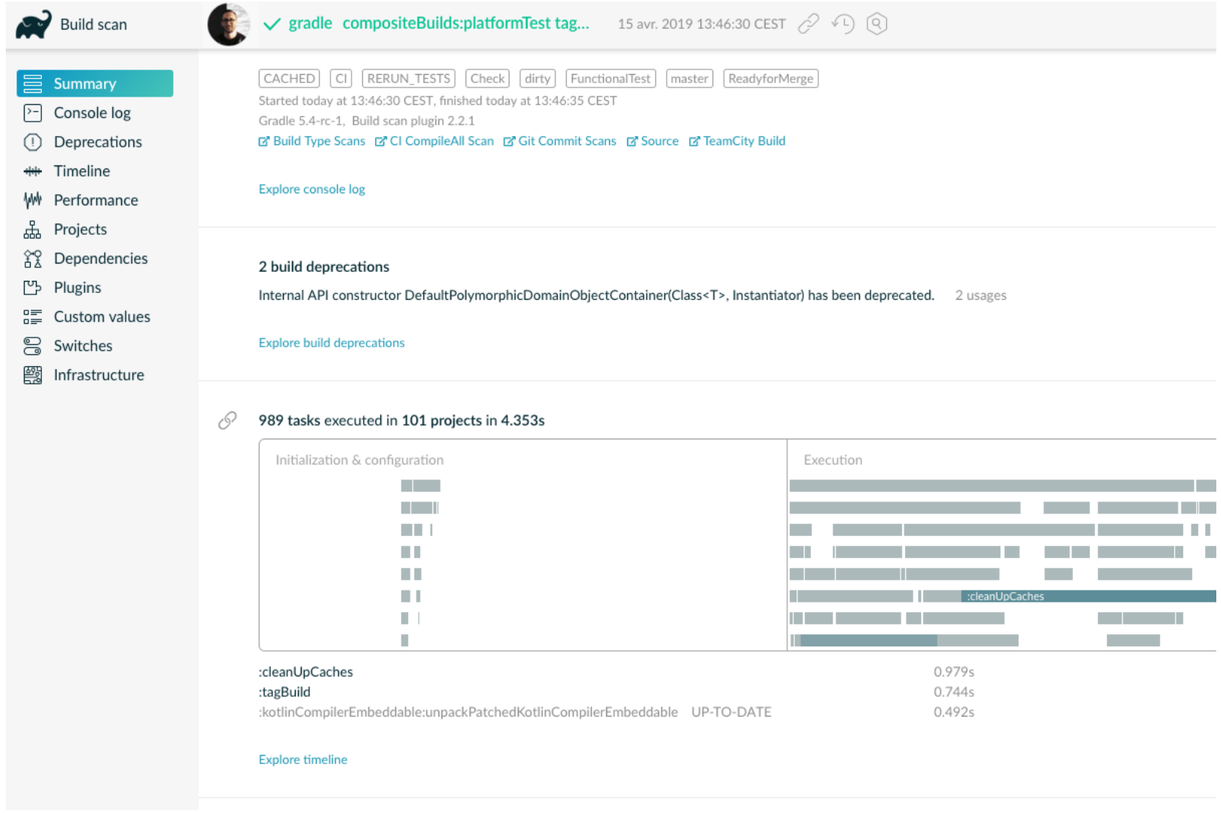


Make and keep builds fast
recapture millions of dollars in developer productivity

Improve reliability
get the data to quickly find and fix build reliability issues

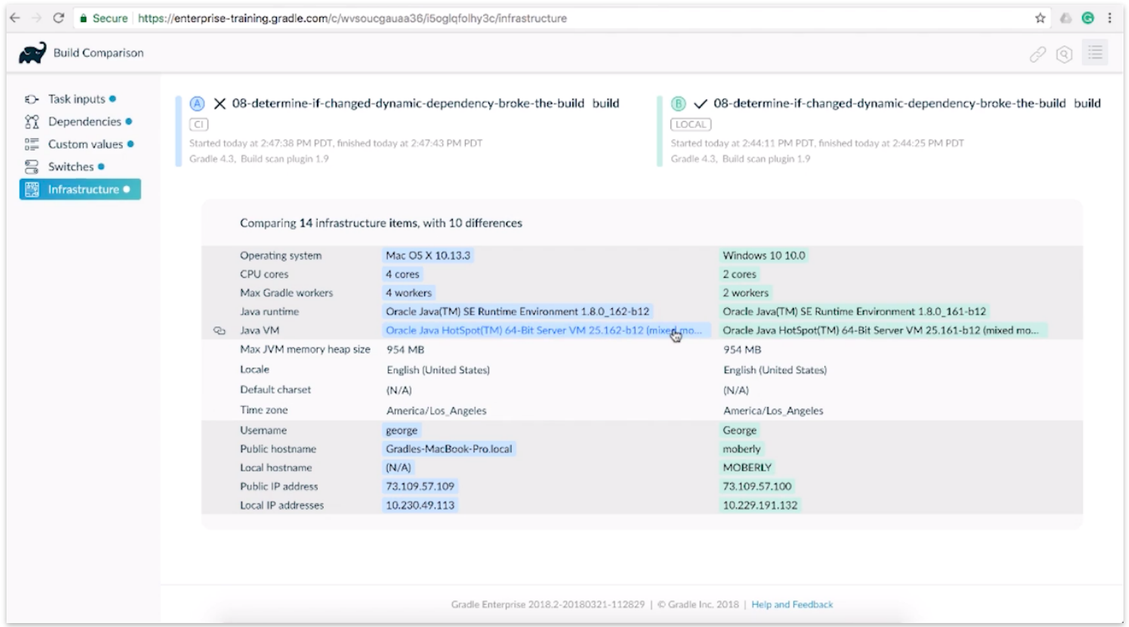
Accelerate debugging
developers fix build issues faster without calling the build team

Build scans



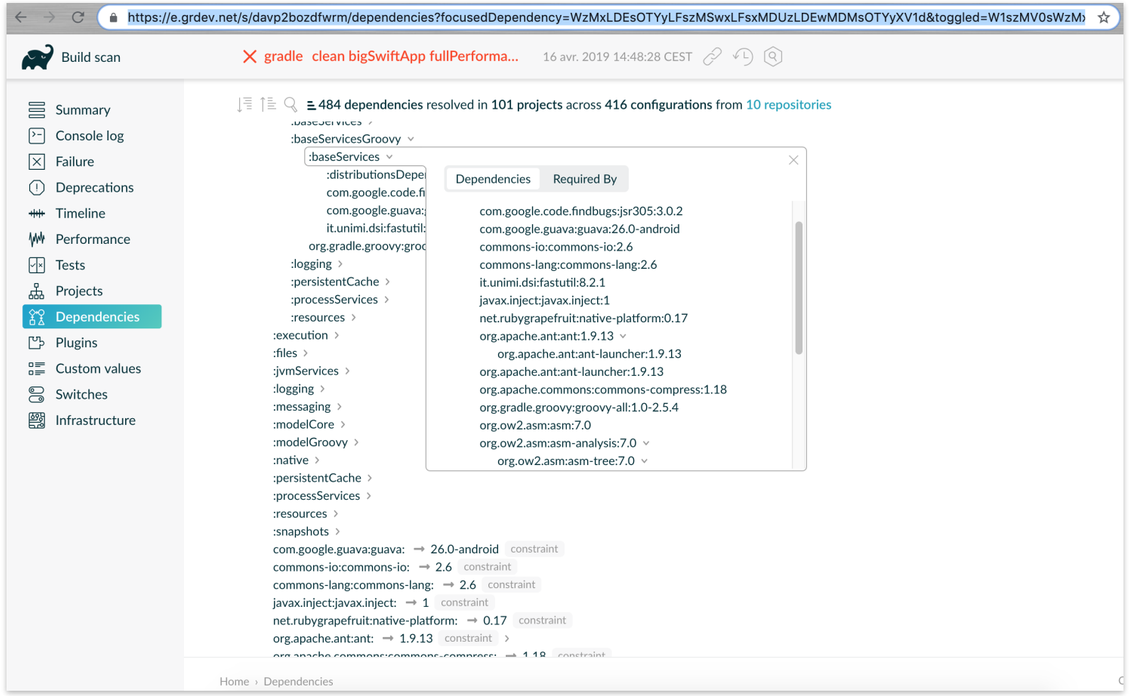
Avoid build reliability problems and huge developer frustration

Build scans



Faster debugging of build failures

Build scans



Easier collaboration with deep-linking

Build scans (Gradle)

build.gradle

```
plugins {
  id "com.gradle.build-scan" version "2.3"
}

buildScan {
  server "https://enterprise-training.gradle.com"
  publishAlways()
}
```

Build scans (Maven)

Requires Gradle Enterprise Maven extension

```
<extensions>
  <extension>
    <groupId>com.gradle</groupId>
    <artifactId>gradle-enterprise-maven-extension</artifactId>
    <version>1.1.2</version>
  </extension>
</extensions>
```

Per project: *«project»/.mvn/extensions.xml*
Per user: put JAR in *«maven-home»/lib/ext/*

Build scans (Maven)

«project»/.mvn/gradle-enterprise.xml

```
<gradleEnterprise>
  <server>
    <url>https://enterprise-training.gradle.com</url>
  </server>
  <buildScan>
    <publish>ALWAYS</publish>
  </buildScan>
</gradleEnterprise>
```

Build scans credentials

- URL: https://enterprise-training.gradle.com/scans
- Username: `attendee`
- Password: `gradle`

Lab 01

Creating a build scan

Lab 01-creating-a-build-scan

Gradle

- Run a build to publish a build scan
  `./gradlew build`
- Follow the link in the console output to view your build scan
- Find the answers to the following questions
  - What is the warning in the console output?
  - What version of `commons-lang` is used by the build?
  - Was the `publishing` plugin applied to the build?
  - What OS and JVM versions was the build run with?
  - Was the build cache enabled?

Maven

- Run a build to publish a build scan
  `./mvnw clean test`
- Follow the link in the console output to view your build scan
- Find the answers to the following questions
  - What version of the compile plugin was used?
  - How much memory was used and how much time was spent on Garbage collection?
  - How much time was spent calculating the execution plan?

scans.gradle.com

Free to use build scan service for Gradle and Maven

Extending build scans

Build scans can be annotated with extra data

- Tags (CI, local, dirty, branch, ...)
- Values (Commit ID, Build number, ...)
- Links (CI build URL, GitHub URL, ...)

Can also be used as search criteria

Extending build scans (Gradle)

build.gradle

```
def ciServer = System.getenv().containsKey('CI')

buildScan {
  tag(ciServer ? "CI" : "local")
  value("buildNumber", 2)
  link("CI Build", "http://ci-server/build/2")
}
```

Or via system properties:

`./gradlew build -Dscan.tag.CI`

Extending build scans (Maven)

pom.xml

```
<plugins>
  <plugin>
    <groupId>org.gradle</groupId>
    <artifactId>gradle-enterprise-maven-extension</artifactId>
    <version>1.1.2</version>
    <configuration>
      <gradleEnterprise>
        <buildScan>
          <tags><tag>CI</tag></tags>
          <links><link><name>CI Build<name><url>http://ci-server/build/
          <values><value><name>buildNumber<name><value>2</value></value
        </buildScan>
      </gradleEnterprise>
    </configuration>
```

Or via system properties:

```
./mvnw package -Dscan.tag.CI
```

Extending build scans (Maven)

Complex logic can be implemented as a Groovy script

```
<plugin>
  <groupId>org.codehaus.gmavenplus</groupId>
  <artifactId>gmavenplus-plugin</artifactId>
  <version>1.6.2</version>
  <executions>
    <execution>
      <id>build-scan-user-data</id>
      <phase>validate</phase>
      <goals>
        <goal>execute</goal>
      </goals>
    </execution>
  </executions>
  ...
```

Extending build scans (Maven)

```
  <configuration>
    <scripts>
      <script>file:///${project.basedir}/buildScanUserData.groovy</scri
    </scripts>
  </configuration>
  <dependencies>
    <dependency>
      <groupId>org.codehaus.groovy</groupId>
      <artifactId>groovy-all</artifactId>
      <version>2.5.6</version>
      <type>pom</type>
      <scope>runtime</scope>
    </dependency>
  </dependencies>
</plugin>
```

Extending build scans (Maven)

buildScanUserData.groovy

```
def buildScan =
  session.lookup("com.gradle.maven.extension.api.scan.BuildScanApi")

def ci = System.getenv('CI')
if (ci) {
  buildScan.tag('CI')
} else {
  buildScan.tag('Local')
}
```

Lab 02

Extending build scans

Lab 02-extending-build-scans

Gradle

- Open `build.gradle`, find the code adding the tags, values and links
- Run a build to publish a build scan
  `./gradlew build`
- Follow the link in the console output to view your build scan
- Find the tags, links and values in your build scan
- Run another build adding a tag at build time
  `./gradlew build -Dscan.tag.TEST`
- Go to the scan list and find your scan with this tag

Maven

- Open `.mvn/gradle-enterprise.xml`, find the code adding the tags, values and links
- Open `pom.xml`, find the code configuring use of a Groovy script
- Open `buildScanUserData.groovy`
- Run a build to publish a build scan
  `./mvnw clean test`
- Follow the link in the console output to view your build scan
- Find the tags, links and values in your build scan
- Run another build adding a tag at build time
  `./mvnw clean test -Dscan.tag.TEST`
- Go to the scan list and find your scan with this tag

What is a build cache?

When the inputs have not changed, the outputs can be reused from a previous run.

Cacheable tasks/goal executions

Gradle Compile/Maven Compile

- Source Files
- Compile Classpath
- Java version
- Java compiler configuration
- …

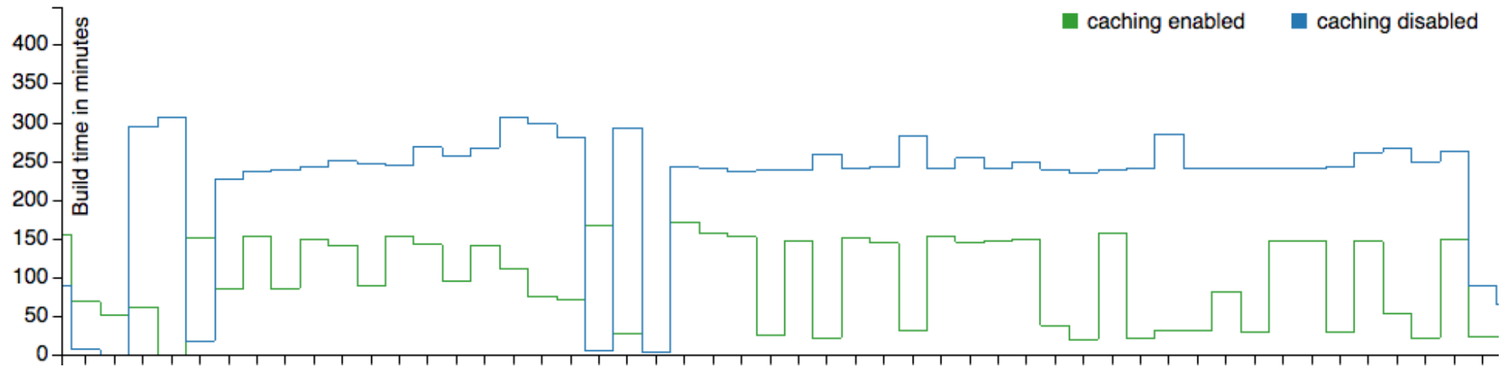
Gradle Test/Maven Surefire

- Source Files
- Runtime Classpath
- Java version
- Java compiler configuration
- …

Caching is a generic feature and applies to all tasks/goals.

For some tasks/goals caching has no benefits (e.g. clean, copy).

Gradle CI builds

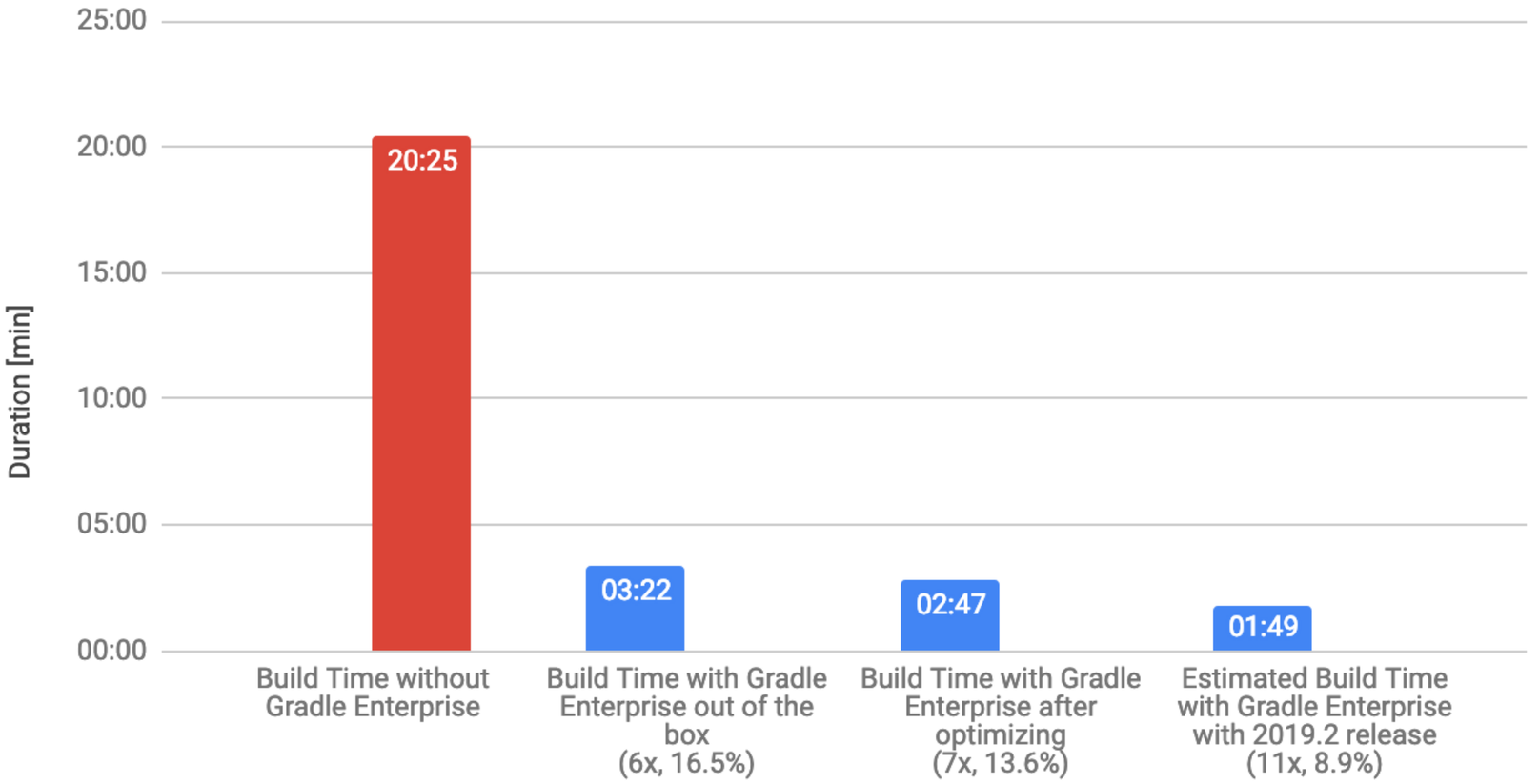


Build times are **> 80%** faster for Gradle core.

Dramatically better caching results due to build scans.

Spring Boot build time for compile and unit tests (fully cached)

https://spring.io/projects/spring-boot



Build caching with Maven

Only supported for clean builds

Local build cache

- Uses cache in directory on local machine
- Speeds up development for single developer or build agent
- Reuses build results when switching branches locally

Enabling the cache (Gradle)

- Built-in
- This build invocation only: `--build-cache` command line option
- All build invocations: `org.gradle.caching=true` in `gradle.properties`

```
./gradlew --build-cache clean assemble
```

Enabling the cache (Maven)

Requires Gradle Enterprise Maven extension

```
<extensions>
  <extension>
    <groupId>com.gradle</groupId>
    <artifactId>gradle-enterprise-maven-extension</artifactId>
    <version>1.1.2</version>
  </extension>
</extensions>
```

Per project: *«project»/.mvn/extensions.xml*

Per user: put JAR in *«maven-home»/lib/ext/*

Lab 03

Using the local build cache

Lab 03-local-cache

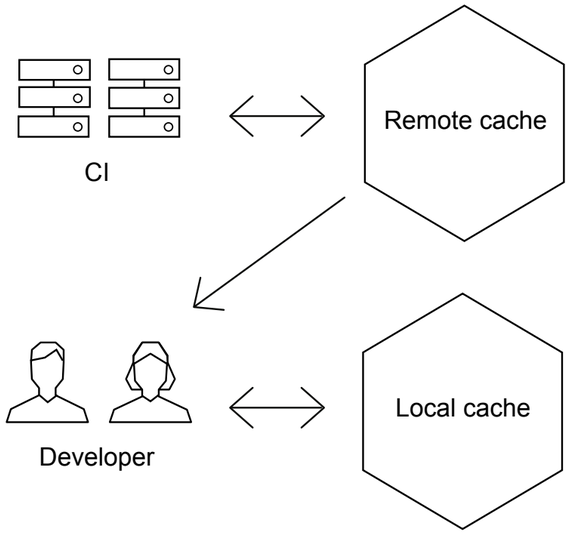
Gradle

- Open `gradle.properties`, note that caching is enabled
  - verbose console is enabled to make seeing the effects of caching easier
- Build the project
  `./gradlew build`
- Build again with clean, notice in the build scan the compilation and test task outcomes are `FROM-CACHE`
  `./gradlew clean build`
- Build again with clean and build cache disabled, notice the much longer build time
  `./gradlew clean build --no-build-cache`

Maven

- Open `.mvn/gradle-enterprise.xml`, note that local cache is not configured (enabled by default)
- Build the project
  `./mvnw clean test`
- Build again, notice in the build scan the compilation and test goal outcomes are `FROM-CACHE`
  `./mvnw clean test`
- Execute just a clean
  `./mvnw clean`
- Then execute tests and notice the cache was not used
  `./mvnw test`
- Build again with build cache disabled, notice the much longer build time
  `./mvnw clean test -Dgradle.cache.local.enabled=false`

Remote build cache



- Shared among different machines
- Speeds up development for the whole team
- Reuses build results among CI agents/jobs and individual developers

Remote cache configuration (Gradle)

settings.gradle

```
buildCache {
    remote(HttpBuildCache) {
        url = 'https://enterprise-training.gradle.com/cache/'
    }
}
```

Conditional cache configuration (Gradle)

settings.gradle

```
def ciServer = System.getenv().containsKey('CI')

buildCache {
    local {
        enabled = !ciServer
    }
    remote(HttpBuildCache) {
        url = 'https://enterprise-training.gradle.com/cache/'
        push = ciServer
    }
}
```

Remote cache configuration (Maven)

```
<gradleEnterprise>
  <server>
    <url>https://enterprise-training.gradle.com</url>
  </server>
</gradleEnterprise>
```

Per project: *«project»/.mvn/gradle-enterprise.xml*

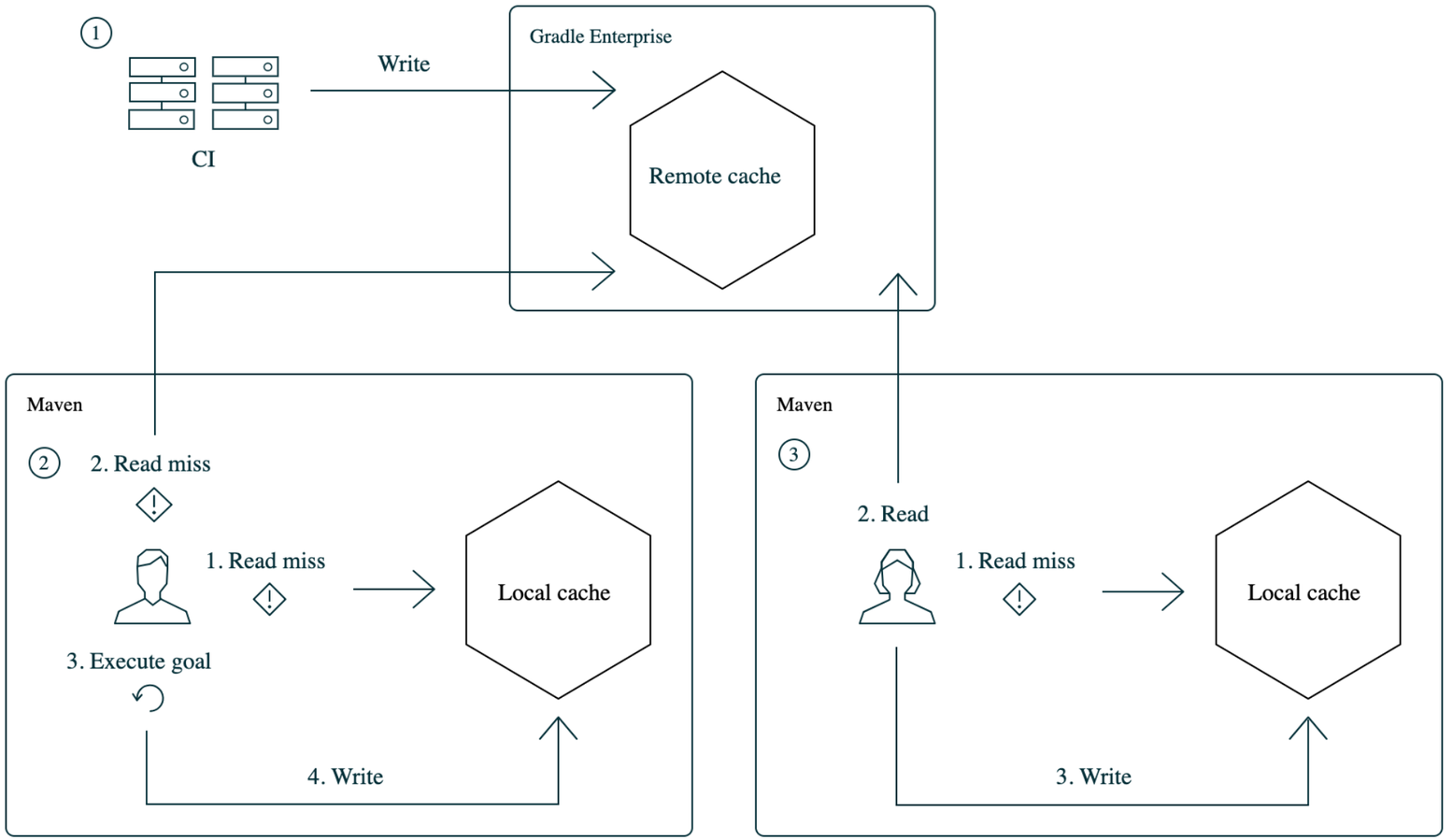
Per user: *~/.m2/gradle-enterprise.xml*.

Conditional cache configuration (Maven)

```
<gradleEnterprise>
  <server>
    <url>https://enterprise-training.gradle.com</url>
  </server>
  <buildCache>
    <remote>
      <storeEnabled>false</enabled>
    </remote>
  </buildCache>
</gradleEnterprise>
```

```
./mvnw clean assemble -Dgradle.cache.remote.storeEnabled=true
```

Common build cache topology



CI reads/writes from/to remote cache. Developers read from remote cache.

Lab 04

Using the remote build cache

Lab 04-remote-cache

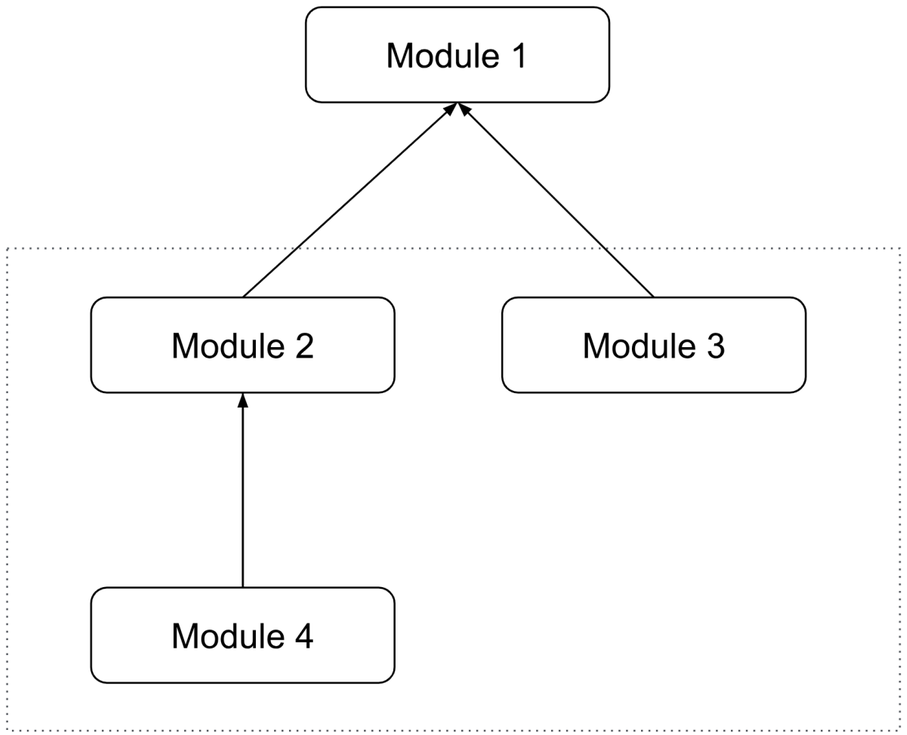
Gradle

- Open `settings.gradle`, note that the remote cache is enabled but not the local cache
- Build the project
  `./gradlew build`
- Go to Performance > Build cache in the build scan. If you have cache hits:
  - Find the `test` task in the Timeline view and inspect its details. Click the “Origin” button (the cube next to the X) to see the build scan of the originating build
  - Make a change to
    `src/test/java/example/ExampleTest.java`
  - Build your project again, notice that tests had to be recompiled but production classes were reused
- If you do not have any cache hits, ask the instructor on how to diagnose the misses with inputs comaprison

Maven

- Open `.mvn/gradle-enterprise.xml`, note that the remote cache is enabled but not the local cache
- Build the project
  `./mvnw clean test`
- Go to Performance > Build cache in the build scan. If you have cache hits:
  - Find the `test` goal in the Timeline view and inspect its details. Click the “View origin scan” link to see the build scan of the originating build
  - Make a change to
    `src/test/java/example/ExampleTest.java`
  - Build your project again, notice that tests had to be recompiled but production classes were reused
- If you do not have any cache hits, ask the instructor on how to diagnose the misses with inputs comaprison

Java compile avoidance



Implementation Change (Non-ABI)

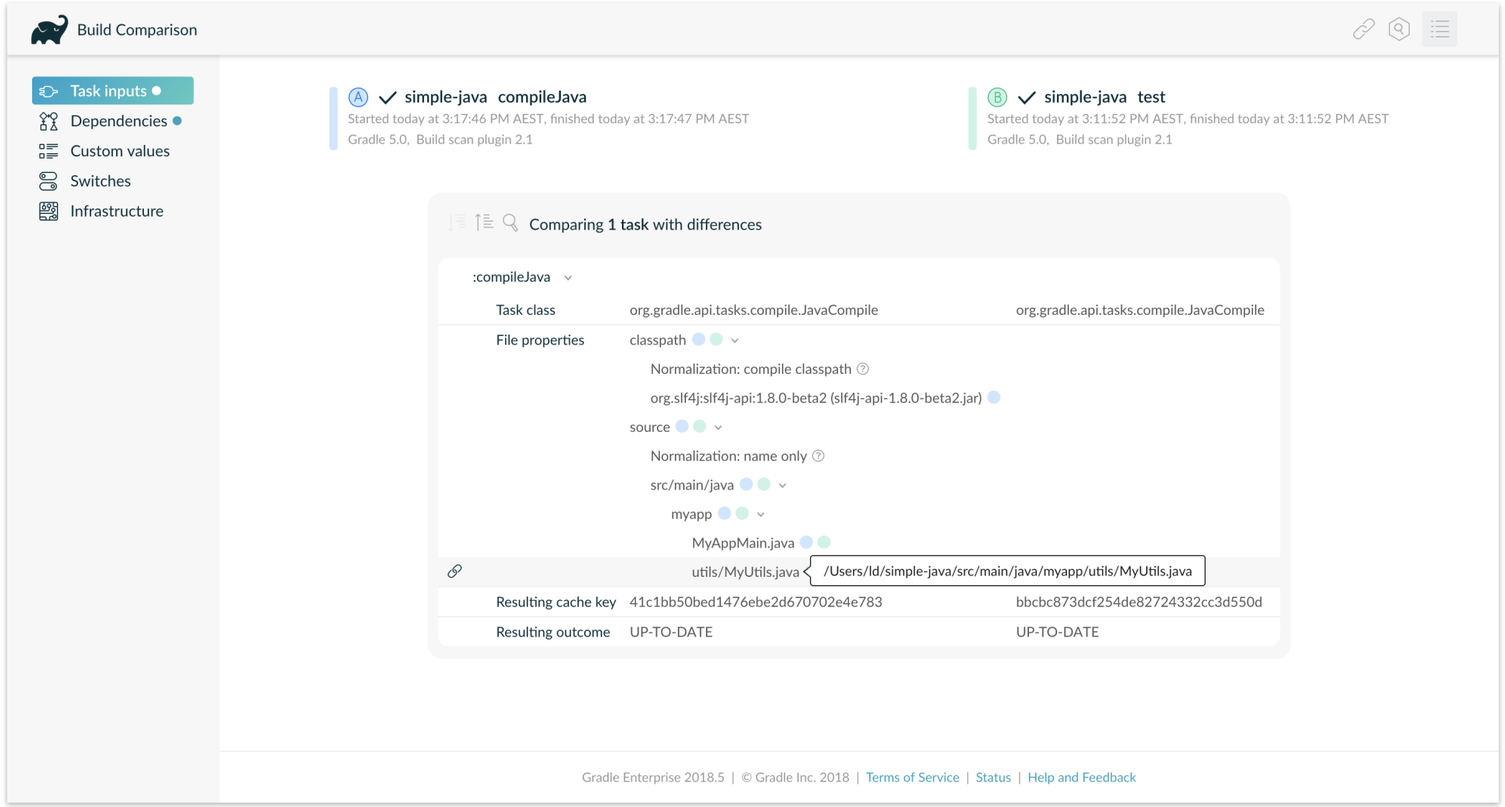
Compile cache key remains unchanged (Cache Hit)

Tests are executed as runtime classpath changed

Input volatility

- Inputs need to be stable and portable
- Common problems:
  - Timestamps
  - Absolute file paths
  - Non-deterministic ordering

Debugging volatile inputs



Gradle Enterprise task / goal inputs comparison visualizes changes in inputs

Runtime classpath normalization

Inconsequential & volatile files on runtime classpaths can be ignored.

More cache hits and faster builds.

Runtime classpath normalization (Gradle)

build.gradle

```
normalization {
    runtimeClasspath {
        ignore 'build-info.properties'
    }
}
```

Runtime classpath normalization (Maven)

pom.xml

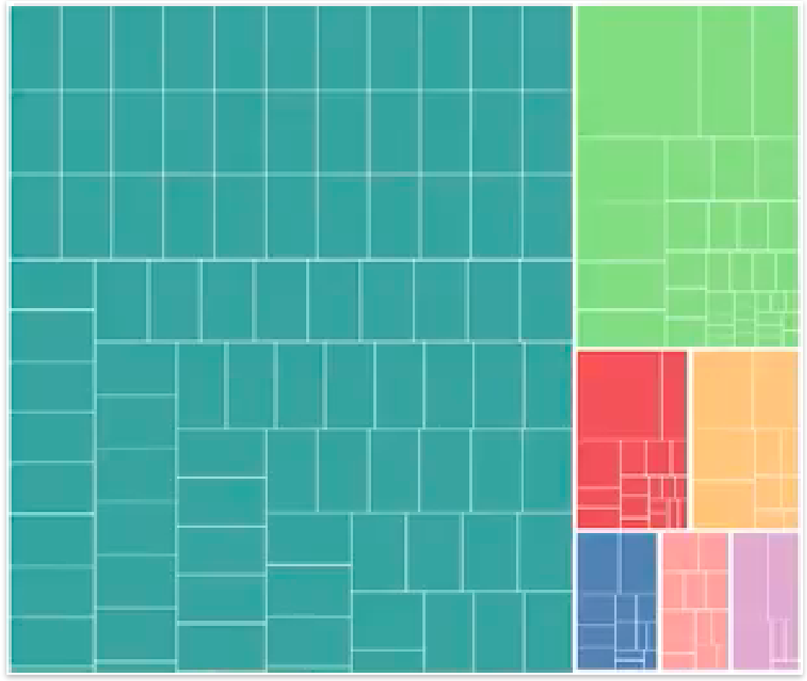
```
<plugin>
  <groupId>com.gradle</groupId>
  <artifactId>gradle-enterprise-maven-extension</artifactId>
  <version>1.1.2</version>
  <configuration>
    <gradleEnterprise>
      <normalization>
        <runtimeClassPath>
          <ignoredFiles>
            <ignoredFile>META-INF/build.properties</ignoredFile>
          </ignoredFiles>
        </runtimeClassPath>
      </normalization>
    </gradleEnterprise>
  </configuration>
```

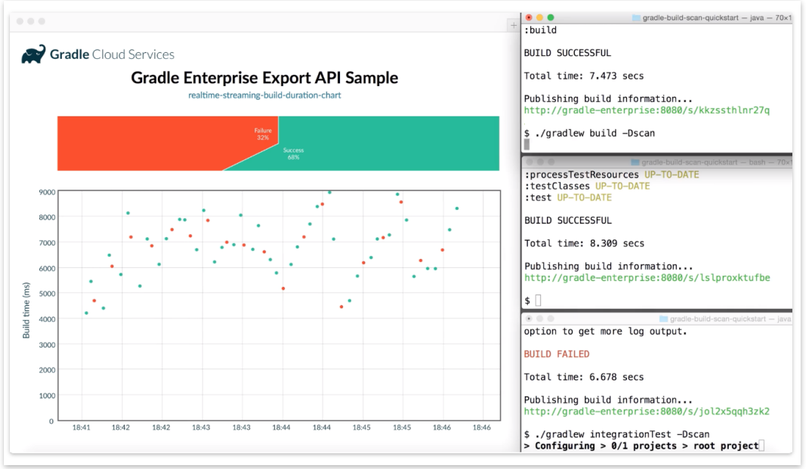
Performance monitoring

Eliminate performance regressions due to a constantly changing environment

Do your own build data analysis

Gradle Enterprise exposes build scan data for real time consumption via API





Export API credentials

- Username: `attendee`
- Password: `gradle`

Lab 05

Live dashboard of build activity

Lab 05-live-dashboard

- Open `index.html` in Chrome or Firefox
- Run some builds and watch them appear on the chart in real-time

Resources

- Gradle Enterprise docs and tutorials: https://docs.gradle.com
- Build Scan Plugin User Manual: https://docs.gradle.com/build-scan-plugin
- Maven Extension User Manual: https://docs.gradle.com/enterprise/maven-extension
- Export API Manual: https://docs.gradle.com/enterprise/export-api
- Try out build scans for Maven and Gradle for free: https://scans.gradle.com

Workshop Survey

https://gradl.es/mdpw-survey

https://gradle.com/training

Gradle Enterprise

gradle.com/enterprise