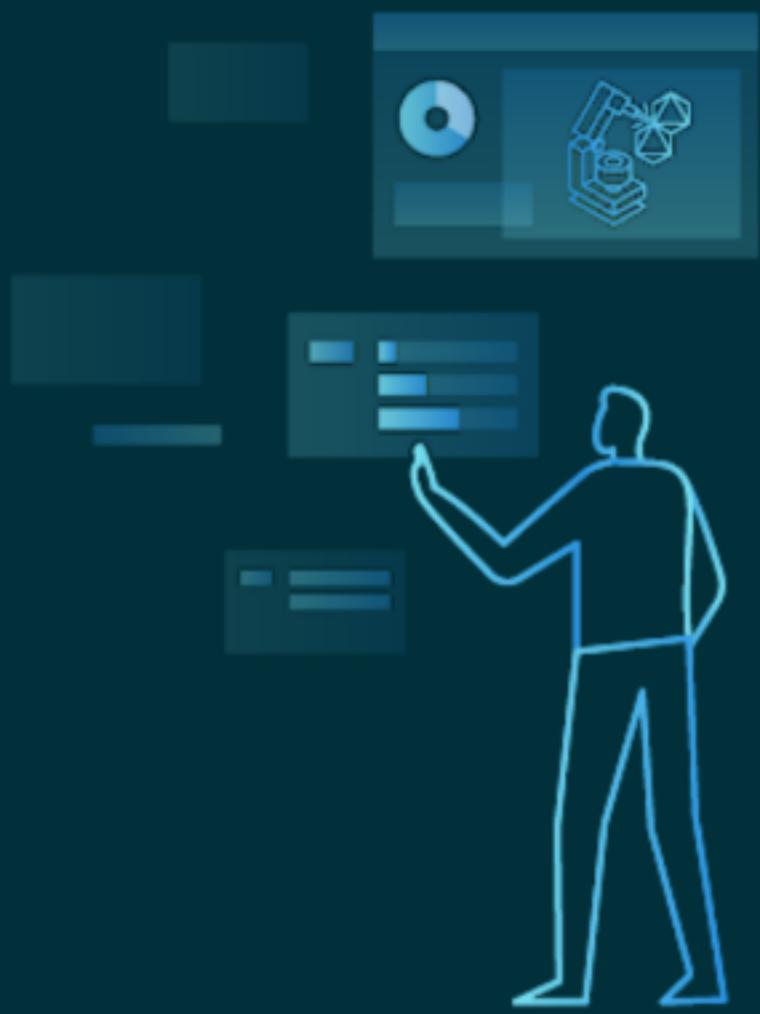


Harnessing Data for DevProd Decisions

A Case Study

Louis Jacomet
DPE Summit 2024





Develocity Reporting & Visualization



What is DRV?

- Leverages Develocity APIs
- Pre-computed dashboard and queries to visualize the Develocity data
- See [Blog post](#) announcing the feature

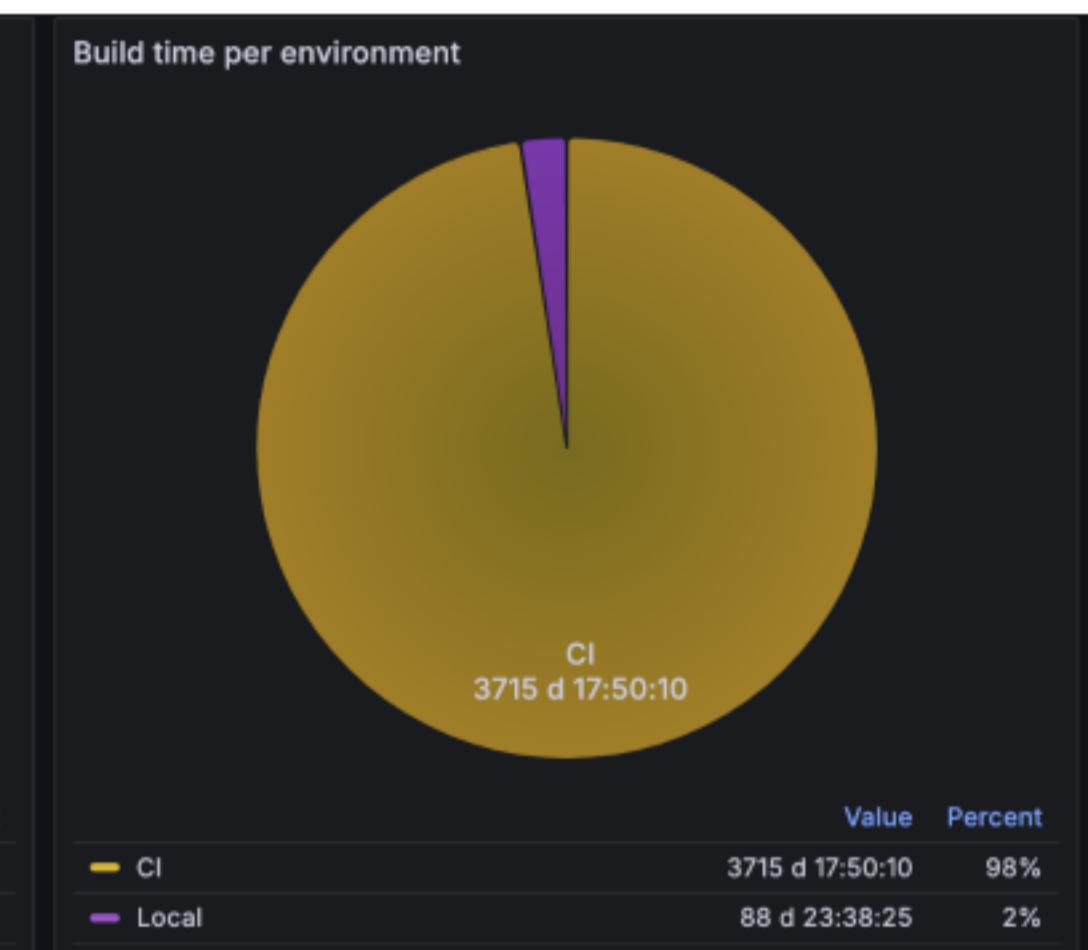
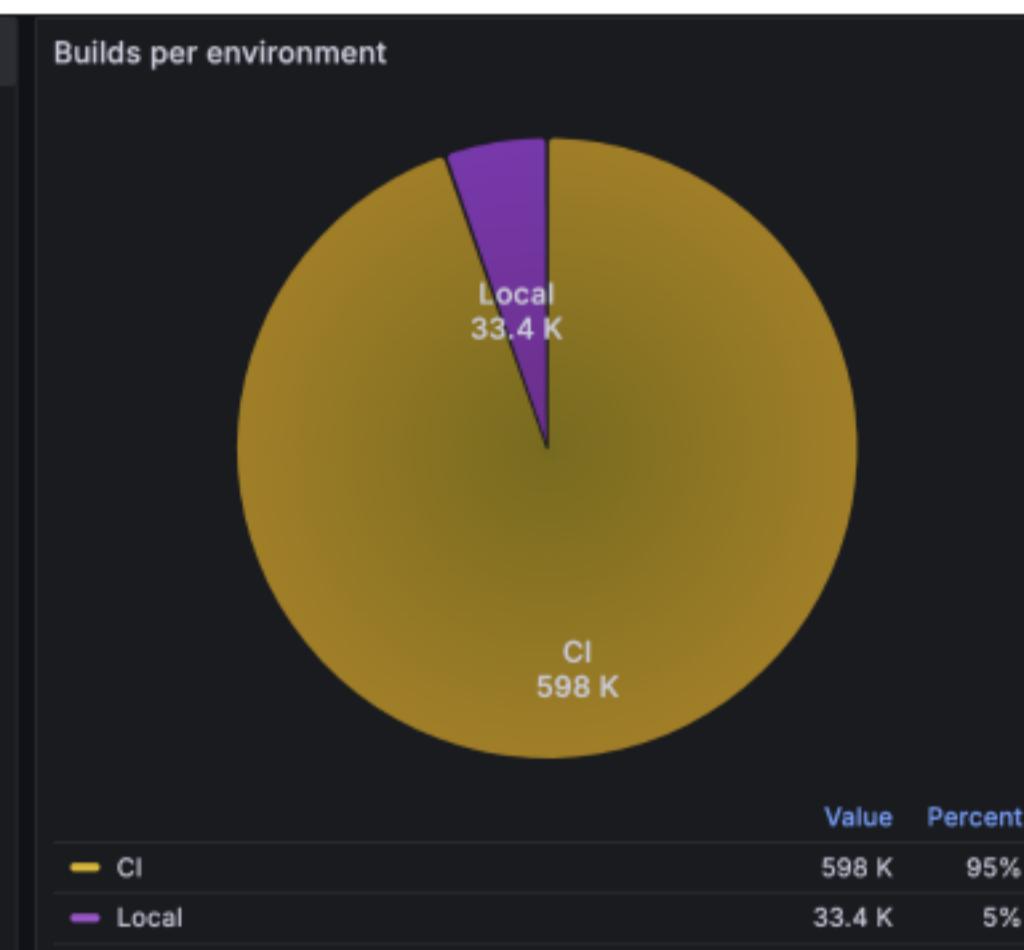
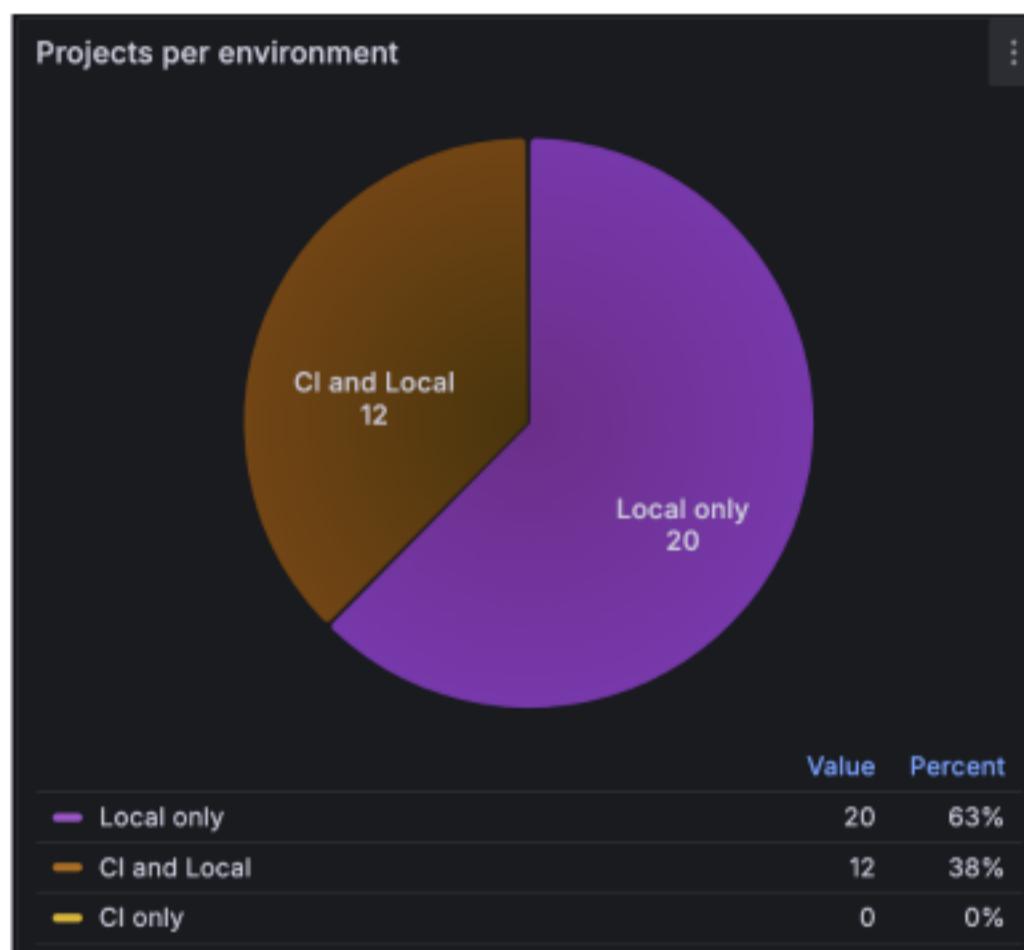
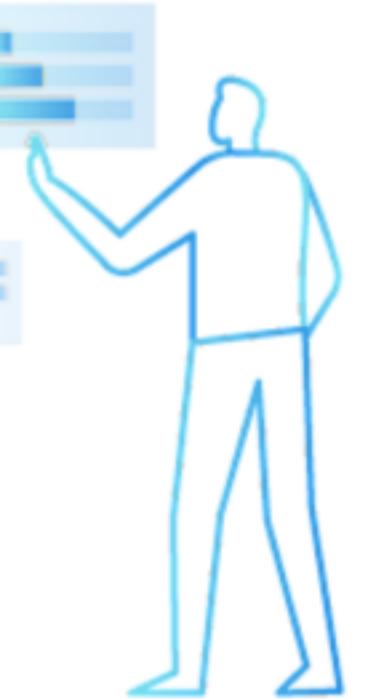




Available data for Gradle's own build







Using 7 days of data for the rest of the presentation



Volume based analysis

Invoked tasks per type

▼ Details per invoked task/goal type

All builds

Invoked task/goal type	Build tool	Environments	Invocations ↓	Execution time
org.gradle.api.tasks.bundling.Jar	Gradle	CI, Local	13.3 Mil	6 d 08:50:46
gradlebuild.basics.tasks.ClasspathManifest	Gradle	CI, Local	10.2 Mil	2 d 10:08:10
org.gradle.api.tasks.compile.JavaCompile	Gradle	CI, Local	8.54 Mil	14 d 01:48:27
org.gradle.docs.samples.internal.tasks.InstallSample	Gradle	CI, Local	6.86 Mil	1 d 13:18:11
org.gradle.language.jvm.tasks.ProcessResources	Gradle	CI, Local	5.86 Mil	16:49:54
org.jetbrains.kotlin.gradle.tasks.KotlinCompile	Gradle	CI, Local	2.96 Mil	9 d 23:52:13

CI builds

Invoked task/goal type	Build tool	Environments	Invocations ↓	Execution time
org.gradle.api.tasks.bundling.Jar	Gradle	CI	12.7 Mil	6 d 07:41:06
gradlebuild.basics.tasks.ClasspathManifest	Gradle	CI	9.73 Mil	2 d 09:22:42
org.gradle.api.tasks.compile.JavaCompile	Gradle	CI	8.17 Mil	13 d 09:32:26
org.gradle.docs.samples.internal.tasks.InstallSample	Gradle	CI	6.40 Mil	1 d 11:09:26
org.gradle.language.jvm.tasks.ProcessResources	Gradle	CI	5.65 Mil	16:34:21
org.jetbrains.kotlin.gradle.tasks.KotlinCompile	Gradle	CI	2.86 Mil	9 d 15:26:05



Are these volumes coherent?

- Some tasks run a very high number of times
- Can we confirm this makes sense?



InstallSample task

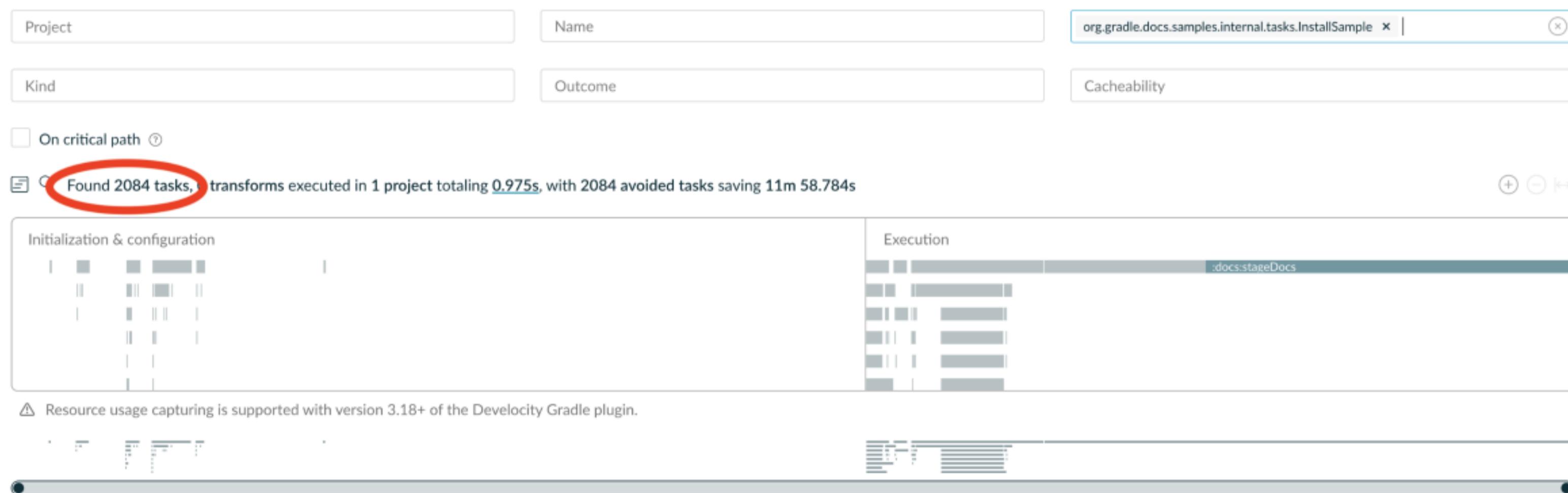
▼ Details per invoked task/goal type

All builds

Invoked task/goal type	Build tool	Environments	Invocations ↓	Execution time	Avg execution time
org.gradle.api.tasks.bundling.Jar	Gradle	CI, Local	13.3 Mil	6 d 08:50:46	00:00:00
gradlebuild.basics.tasks.ClasspathManifest	Gradle	CI, Local	10.2 Mil	2 d 10:08:10	00:00:00
org.gradle.api.tasks.compile.JavaCompile	Gradle	CI, Local	8.54 Mil	14 d 01:48:27	00:00:00
org.gradle.docs.samples.internal.tasks.InstallSample	Gradle	CI, Local	6.86 Mil	1 d 13:18:11	00:00:00
org.gradle.language.jvm.tasks.ProcessResources	Gradle	CI, Local	5.86 Mil	16:49:54	00:00:00
org.jetbrains.kotlin.gradle.tasks.KotlinCompile	Gradle	CI, Local	2.96 Mil	9 d 23:52:13	00:00:00



How often does it run in a single build?



- Indicates we run 3k documentation builds in a week!



Execution time based analysis

Gradle's testing setup

- Gradle has lots of build-based tests
- They run in different configurations:
 - With the daemon
 - Without the daemon
 - With configuration cache
 - With isolated projects



Embedded integration tests

- Additional test setup that runs the test in-process with the build daemon
- Faster to start, easier to debug
- Have corner cases where they just don't work
- Source of confusion for devs
 - Comes up in developer satisfaction surveys



On CI?

- quickTest uses the embedded mode
- platformTest uses the "forking" mode
- Adds one "stage" to the CI pipeline



DRV data - Core tests

CI builds				
Requested tasks/goals ↑	OS	Builds	Build time	Avg build time
clean configuration-cache:quickTest	Windo	101	07:54:50	00:04:42
clean configuration-cache:quickTest	Linux	44	04:04:29	00:05:33
clean core:platformTest	Windo	92	09:01:05	00:05:52
clean core:platformTest	Linux	18	01:14:26	00:04:08
clean core:platformTest	macOS	4	00:02:58	00:00:44
clean core:quickTest	Windo	68	05:59:39	00:05:17
clean core:quickTest	Linux	44	03:53:40	00:05:18



DRV data - DM tests

clean dependency-management:platformTest declarative	Linux	4	00:39:49	00:09:57
clean dependency-management:platformTest declarative	Linux	7	00:49:52	00:07:07
clean dependency-management:platformTest stdlib-kotlin	Linux	6	00:04:39	00:00:46
clean dependency-management:quickTest	Windows	119	12:31:19	00:06:18
clean dependency-management:quickTest declarative	Linux	7	00:52:12	00:07:27
clean dependency-management:quickTest declarative	Linux	30	03:14:40	00:06:29
clean dependency-management:quickTest stdlib-kotlin	Linux	7	00:09:56	00:01:25



Follow up

- DRV gives us the data to back a decision with regards to embedded tests
- If this mode is not faster, is there any benefit to it?



Negative cache savings

- When caching gets in the way
- Restoring from cache takes more time than running the task



Negative cache savings

All builds	Invoked task/goal type	Build tool	Environments	Invocations	Execution time	BC savings ↑
	gradlebuild.basics.tasks.ClasspathManifest	Gradle	CI, Local	9.64 Mil	2 d 07:59:00	09:53:25 ago
	org.gradle.kotlin.dsl.provider.plugins.precompiled.i	Gradle	CI, Local	1.11 Mil	01:54:47	00:47:44 ago
	org.gradle.kotlin.dsl.provider.plugins.precompiled.i	Gradle	CI, Local	1.11 Mil	02:26:32	00:18:47 ago



Follow up

- Disable caching of ClasspathManifest task



Dependency downloads

Gradle CI strategy

- Fleet of long lived agents
 - Includes process killing and files clean-ups
- Fleet of on-demand agents
 - Start fresh on agent activation
 - Remain available as long as there is work available



Overall downloads data

Total downloads	Total download size	Total download time	Total wall-clock download time
1.64 Mil	578 GiB	1 d 23:23:34	19:24:09



Per requested tasks

Requested tasks/goals	Build tool	Environments	Builds ↓	Downloads	Avg downloads	Download size	Avg download size	Download time
clean quickFeedbackCrossVersion	Gradle	CI	524	118 K	225	31.5 GiB	61.6 MiB	03:38:48
clean docs:docsTest	Gradle	CI	383	126 K	329	35.7 GiB	95.4 MiB	03:03:31
clean docs:docsTest docs:checkS	Gradle	CI	255	42.1 K	165	13.2 GiB	52.8 MiB	01:05:28
clean sanityCheck	Gradle	CI	223	36.9 K	165	34.0 GiB	156 MiB	00:56:09
clean sanityCheck test	Gradle	CI	162	12.0 K	73.8	20.6 GiB	130 MiB	00:23:32
clean tooling-api:quickTest	Gradle	CI	119	23.1 K	194	6.37 GiB	54.8 MiB	00:36:25

Requested tasks/goals	Build tool	Environments	Builds ↓	Downloads	Avg downloads	Download size	Avg download size	Download time
clean quickFeedbackCrossVersion	Gradle	CI	524	118 K	225	31.5 GiB	61.6 MiB	03:38:48
clean docs:docsTest	Gradle	CI	383	126 K	329	35.7 GiB	95.4 MiB	03:03:31
clean docs:docsTest docs:checkS	Gradle	CI	255	42.1 K	165	13.2 GiB	52.8 MiB	01:05:28
clean sanityCheck	Gradle	CI	223	36.9 K	165	34.0 GiB	156 MiB	00:56:09
clean sanityCheck test	Gradle	CI	162	12.0 K	73.8	20.6 GiB	130 MiB	00:23:32
clean tooling-api:quickTest	Gradle	CI	119	23.1 K	194	6.37 GiB	54.8 MiB	00:36:25

Requested tasks/goals	Build tool	Environments	Builds ↓	Downloads	Avg downloads	Download size	Avg download size	Download time
	Gradle	Local	66	5.08 K	76.9	1.53 GiB	23.7 MiB	03:25:51
sanityCheck	Gradle	Local	19	657	34.6	2.31 GiB	125 MiB	00:15:01
sC	Gradle	Local	7	208	29.7	660 MiB	94.3 MiB	00:02:35
install	Gradle	Local	6	74	12.3	636 KiB	106 KiB	00:00:49
	Gradle	Local	5	71	14.2	158 MiB	31.7 MiB	00:00:34
saChe	Gradle	Local	5	8	1.60	522 MiB	104 MiB	00:00:37



"No downloads" strategy

- Only a fraction of our builds do downloads
- Example with :docs:docsTest builds:
 - 1.86k builds per week
 - Only 323 download 1 file or more



Thank you for your attention!



```
speaker {  
    name = "Louis Jacomet"  
    title = "Senior Lead Software Engineer"  
    twitter = "@ljacomet"  
    mastodon = "@ljacomet@foojay.social"  
    github = "ljacomet"  
}
```



2024 Developer Survey by Gradle



Help Us Improve
YOUR Developer Experience!



* Ask your team, too!

