

Bazel Release Announcements (2023)

The goal of this document is to edit Bazel release notes and prepare the release announcement. Anyone is welcome to suggest improvements and add information. After a Bazel release, the publisher converts the content to Markdown and publishes it on the [Bazel blog](#) (send a PR to [bazelbuild/bazel-blog](#)).

How to review the notes?

- Rule of thumb: Imagine someone who just started to use Bazel. They read the announcement, they want to know what's important and understand what the points are about.
 - If you mention a native rule, a function, a flag, or an important concept, **add a link to the documentation**. It's very rare that an item doesn't need any link to the Bazel documentation.
 - Prefer versioned links whenever possible:
`https://bazel.build/versions/6.0.0/foo/bar` instead of
`https://bazel.build/foo/bar`.
 - Avoid showing urls, prefer a link on a real text. Avoid anchors like "click here" and instead make the anchored text self-descriptive: think as if the text was being printed and the link wasn't available.
 - Fix the formatting if needed. Start the line with an uppercase, end it with a point.
 - Use backquotes around code, symbols, flags, or any word containing an underscore.
 - Review the [Release Notes guidelines](#). They contain important advice on the phrasing to keep the document consistent.
-

Bazel 7.0.0 LTS

[Release notes for all releases after 7.0 can be found here.](#)

Bazel 6.4.0 LTS

Release Manager: Hee Cha

Reviewers: Xudong Yang Yun Peng

Status: Approved

Tracking issue: [#19035](#)

Bazel 6.4.0 is a minor LTS release. It is fully backward compatible with Bazel 6.0 and contains selected changes by the Bazel community and Google engineers.

General

- Add support for more workspace boundary files to bash completion ([#19281](#))
- Ignore Starlark options on commands with ``allowResidue = False`` ([#19417](#))
- Print Passed and Failed methods in detailed test summary ([#19505](#))
- Keep leading zero in formatted date ([#19694](#))
- Merge rule and aspect validation output groups ([#19745](#))
- Bazel release process: Fix push justification. ([#19768](#))
- Clear runfiles environment variables for ``bazel run`` ([#19606](#))

Android

- Add a Starlark flag that allows disabling proguard. This will be useful for testing later. ([#19179](#))

C++ / Objective-C

- Fix a bug where frozen targets list was mutated while expanding env attribute ([#19053](#))
- Advertise CcInfo from cc_import ([#19086](#)) ([#19088](#))
- Enable cc toolchain resolution when cross compiling to windows arm64. ([#19198](#))
- Add Starlark implementation for several CcCommon methods. ([#19076](#))
- Rename ``cc_test_wrapper`` to ``cc_test`` ([#19231](#))
- Add ``additional_linker_inputs`` option to ``cc_library`` rule ([#19264](#))
- Add `--incompatible_disable_objc_library_transition` ([#19393](#))
- Wrong include path to Clang 16 on Windows ([#19430](#))
- Remove default `-s` flag from macOS libtool invocation ([#19454](#))
- Only use ``/showIncludes`` if supported ([#19521](#))
- Advertise CcInfo from cc_proto_library ([#19534](#))
- Update unknown Xcode version error message and provide an environment variable to force re-evaluation ([#19540](#))
- cc_library: propagate data dependencies via implementation_deps. ([#19590](#))
- Error on potential unsupported ``/showIncludes`` lines ([#19611](#))
- Flip `--experimental_cc_implementation_deps` ([#19751](#))
- Bump c++ standard to c++14 per default ([#19794](#))
- Collect debug info context from implementation deps ([#19836](#))

Configurability

- Performance improvement: ``--allow_analysis_cache_discard=false`` aborts the build if changed build flags would otherwise discard the analysis cache ([#19503](#))

- Performance, correctness improvement: add `diff_against_dynamic_baseline` option to `--experimental_output_directory_naming_scheme` (#19514)
- Improve error when a label is provided in `config_setting`'s `values` (#19484)
- Do not allow `applicable_licenses` on `platform` (#19426)
- Raise an early error on invalid labels in transitions inputs/outputs (#19764)
- Fix unconditional Skyframe invalidation with `--lockfile_mode=update`

External Dependencies / Bzlmod

- Merge `use_repo` buildifier fixups into a single command (#19134)
- Ensure that extension unique names followed by `~` are prefix-free (#19164)
- Lockfile updates & fixes (#19153)
 - A new attribute `'environ'` is added to module extension to allow depending on environment variables.
 - The module extension will be re-evaluated in response to changes in the files it depends on.
 - Lockfile version bump will prompt users to run it in `'lockfile_mode=update'`.
- Friendlier error message for `'bazel_dep'`s without `'version'` (#19196)
- Use `'debugPrint'` instead of `'str'` for `'fail'` arguments (#19283)
- Download `'BazelRegistryJson'` only once per registry (#19300)
- Make module extension tag's `'debugPrint'` useful for error messages (#19285)
- Intern repository mapping entries (#19293)
- Do not rerun module extensions when only imports or locations change (#19284)
- Fetch `'RepoSpecs'` in parallel (#19354)
- Make `'MODULE.bazel.lock'` deterministic (#19370)
- Ensure lockfile is updated after reset to pre-build state (#19371)
- Add visionOS support (#19436)
- Cherry pick platform dependent lockfile (#19498)
 - New attributes `'os'` and `'arch'` added to module extension to signify its reliance on the operating system or system architecture.
 - When either the `'os'` attribute, the `'arch'` attribute, or both are set to `'true'`, separate module extensions will be stored in the lockfile for each distinct combination of operating system and architecture.
- Print dep chain leading to a module that is in error (#19543)
- Show fetch progress for the `'mod'` command (#19542)
- Explain the use of `'str(Label(...))'` in the docs (#19554)
- Inject builtin modules at the end of the `MODULE.bazel` file (#19573)
- Use case-insensitive comparison for Windows paths in `'runfiles.bash'` (#19626)
- Show test labels in summaries in display form (#19625)
- Remove stale extension entries from lockfile (#19683)
- Bzlmod lockfile: fix pretty printing for attributes (#19691)
- Fixed a case where the `MODULE.bazel.lock` file contains user specific paths (#19698)
- Consider RCs equivalent to release for `'bazel_compatibility'` (#19689)
- Remove stale extension entries from lockfile if module order changes (#19730)

- Make lockfile's `RepoSpec` attributes more readable ([#19748](#))
- Fix handling of non-ASCII characters in archive entry file names ([#19765](#))
- Fix crash when `environ` contains duplicate entries ([#19827](#))

Java

- Add toolchain type for Java bootstrap runtime ([#19220](#))
- Add formatted timestamp entries to volatile workspace status file ([#19499](#))
- Update java_tools to v12.7 ([#19522](#))
- Use `Label` in `@bazel_tools//tools/jdk` macros ([#19675](#))
- Update rules_java to 5.5.1 ([#19701](#))
- Fix Java compilation for jdk21 ([#19735](#))
- Handle synthetic method parameters entries that don't have names ([#19758](#))

Performance

- Add profiling for Bzlmod operations ([#19313](#))
- Intern empty `Depset`s ([#19443](#))
- Optimize classpath pre-processing in java_stub_template.txt ([#19491](#))
- Also apply `NestedSet` optimizations to `Depset` ([#19492](#))

Query

- Fix valid json when using jsonproto output in queries with new `--output=streamed_jsonproto` implementation. ([#19226](#))
- Add `--consistent_labels` flag to all query commands ([#19567](#))

Remote Cache and Execution

- Support multiple remote execution digest functions ([#19042](#))
- Add the `--remote_require_cached` flag ([#19075](#))
- BLAKE3 can be used as a digest function with the `--digest_function=blake3` startup flag ([#19191](#))
- Ensure disk cache root exists ([#19225](#))
- Retry on `javax.net.ssl.SSLException ... BAD_DECRYPT` ([#19346](#))
- build-runfiles: remove temporary file prior to creating it ([#19386](#))
- Take the `no-remote-exec` tag into account when computing the action salt ([#19457](#))
- Mark tool inputs in the execution log. ([#19483](#))
- Add output name to `CacheNotFoundException` ([#19452](#))
- Fix output materialized as symlink when building without the bytes. ([#19739](#))
- Add blake3 NEON instructions on linux arm64 ([#19804](#))

Rules API

- Added a new top-level symbol `PackageSpecificationInfo`, a provider for `package_group` targets. It has a `contains` method for allowlist checking. ([#19422](#), [#19425](#))
- Always fail on unknown attributes ([#19404](#))
- Add `--incompatible_merge_fixed_and_default_shell_env` ([#19319](#))
- Include name in `repr` of exported `rule`'s ([#19229](#))

Acknowledgements:

This release contains contributions from many people at Google, as well as Andreas Herrmann, Brentley Jones, Chirag Ramani, Ed Schouten, Fabian Meumertzheim, George Gensure, Julio Merino, Keith Smiley, Matt Mackay, NelsonLi0701, Nicholas Junge, Orion Hodson, Roman Salvador, Ted Kaplan, Thi Doan, and Tyler Williams.

Bazel 6.3.2 LTS

Release Manager: Hee Cha

Reviewers: Yun Peng Xudong Yang

Status: Approved

Tracking issue: [#19165](#)

Bazel 6.3.2 is a patch LTS release. It is fully backward compatible with Bazel 6.0 and contains selected changes by the Bazel community and Google engineers.

- Ensure that extension unique names followed by `~`` are prefix-free ([#19167](#))
- Module extensions can now specify a list of environment variables to watch through the `environ` parameter.
- The module lockfile can now deal with missing fields due to Bazel version upgrades. ([#19105](#))
- Module extensions are now re-evaluated if any labels they depend on change, similar to repo rules. ([#19068](#))

This release contains contributions from many people at Google, as well as Fabian Meumertzheim.

Bazel 6.3.1 LTS

Release Manager: Hee Cha

Reviewers: Yun Peng Xudong Yang

Status: Approved ▾

Tracking issue: [#19045](#)

Bazel 6.3.1 is a patch LTS release. It is fully backward compatible with Bazel 6.0 and contains selected changes by the Bazel community and Google engineers.

- Disabled lockfile support by default due to issue reports ([#19105](#) and [#19068](#)). Will be re-enabled in the next minor release.
- Mark isolated extension usages as experimental ([#19021](#))
- Fix a bug where frozen targets list was mutated while expanding env attribute ([#19046](#))
- Advertise CcInfo from cc_import ([#19086](#))
- Update java_tools to v12.6 ([#19091](#))

This release contains contributions from many people at Google, as well as Brentley Jones and Fabian Meumertzheim.

Bazel 6.3.0 LTS

Release Manager: Hee Cha

Reviewers: Xudong Yang

Status: Approved ▾

Tracking issue: [#18368](#)

Bazel 6.3.0 is a minor LTS release. It is fully backward compatible with Bazel 6.0 and contains selected changes by the Bazel community and Google engineers.

Highlights

- The new `bazel mod` command allows users to inspect their external dependency graph in Bzlmod. See [documentation](#) for details.
- Full lockfile support has been added to Bzlmod and is now enabled by default. See [documentation](#) for details.

General

- Options specified on the pseudo-command ``common`` in ``.rc`` files are now ignored by commands that do not support them as long as they are valid options for *any* Bazel command. Previously, commands that did not support all options given for ``common`` would fail to run. These previous semantics of ``common`` are now available via the new ``always`` pseudo-command. ([#18609](#))
- Improve error on invalid ``-//foo`` and ``-@repo//foo`` options ([#18516](#))
- Adjust `--top_level_targets_for_symlinks``: ``bazel-bin``, ``bazel-testlogs``, etc. no longer mysteriously disappear ([#18916](#))
- Fix Xcode 14.3 compatibility ([#18490](#))
- Remove option to disable ``ForkJoinPool``. ([#18791](#))

Android

- Fix D8 dex merger failure when a synthetic class is placed on a different shard than its container class ([#16368](#))
- Add a new provider for injecting native libs in `android_binary` for `android_binary` Starlarkification ([#18753](#))
- Add a new provider for passing dex related artifacts in `android_binary` for `android_binary` Starlarkification ([#18899](#))

Build Event Protocol

- Add `ActionCacheStatistics` to BEP ([#18914](#))

C++ / Objective-C

- Make `cpp` file extensions case sensitive again ([#18552](#))
- Add changes to `cc_shared_library` from head to 6.3 ([#18606](#))
- Make `grep_includes` optional inside `cc_common.register_linkstamp_compile_action` ([#18823](#))
- Add implementation deps support for Objective-C ([#18372](#))
- Add `external_include_paths` feature on Windows toolchain ([#18654](#))
- Additional source inputs can now be specified for compilation in `cc_library` targets using the `additional_compiler_inputs` attribute, and these inputs can be used in the `$(location)` function. ([#18882](#))
- Fix VS 2022 autodetection ([#18960](#))

Coverage

- Expose `metadata_files` parameter in `coverage_common.instrumented_files_info` ([#18838](#))
- Check if json.gz files exist, not the gcov version. ([#18889](#))
- Add flag `--experimental_collect_code_coverage_for_generated_files` to enable coverage collection for generated files. ([#18664](#))
- Fix split post-processing of LLVM-based coverage ([#18737](#))

External Dependencies / Bzlmod

- Overrides specified by non-root modules no longer cause an error, and are silently ignored instead. ([#18388](#))
- The `--credential_helper` flag is now stable, and works for repo fetching.
- Module extension usages can now be specified to be `_isolated_`. Each isolated usage causes a separate evaluation of the extension. See [documentation](#) for more details.
- Report percentual download progress in repository rules ([#18471](#))
- Ignore hash string casing ([#18414](#))
- Include actual MODULE.bazel location in stack traces ([#18612](#))
- Fix WORKSPACE toolchain resolution with `--enable_bzlmod` ([#18649](#))
- The REPO.bazel and MODULE.bazel files are now also considered as workspace boundary markers. ([#18787](#))
- Yanked module versions no longer contribute dependency requirements or emit `DEBUG` messages for `print()` statements ([#18908](#))
- Report dev/non-dev deps imported via non-dev/dev usages ([#18922](#))
- Identify isolated extensions by exported name ([#18923](#))

Java

- Pass version to `java_runtimes` created by `local_java_repository` ([#18415](#))
- Teach ijar about dynamic constants ([#18729](#))
- Update to latest turbine version ([#18803](#))
- Update Error Prone to 2.20.0 ([#18885](#))
- Update java_tools to v12.5 ([#18868](#))
- Disable UseCorrectAssertInTests by default ([#18948](#))

Local Execution

- Use `local_termination_grace_seconds` when testing `LinuxSandbox` availability ([#18568](#))

Query

- Add `jsonproto` option to `query --output` flag ([#18438](#))
- `query --output=proto --order_output=deps` now returns targets in topological order (previously there was no ordering). ([#18870](#))

Remote Execution

- Prevent CredentialHelperEnvironment crash when invoking Bazel outside of a workspace. ([#18430](#))
- Use wall-time for credential helper invalidation ([#18413](#))
- Move credential helper setup into `remote_helpers.sh` so it can be reused by other shell tests. ([#18453](#))
- Support remote symlink outputs when building without the bytes. ([#18476](#))
- Enrich local BEP upload errors with file path and digest possible. ([#18481](#))
- Extend the credential helper default timeout to 10s. ([#18527](#))
- Ignore all errors when writing to stdin of a credential helper. ([#18540](#))
- Implement failure circuit breaker ([#18541](#))
- Add `ServerCapabilities` into `RemoteExecutionClient` ([#18442](#))
- `RemoteExecutionService`: support `output_symlinks` in `ActionResult` ([#18441](#))
- `RemoteExecutionService`: `Action.Command` to set `output_paths` ([#18440](#))
- Add ``ActionExecutionMetadata`` as a parameter to ``ActionInputPrefetcher#prefetchFiles``. ([#18656](#))
- Use `failure_rate` instead of failure count for circuit breaker ([#18559](#))
- Update `ignored_error` logic for circuit breaker ([#18662](#))
- Don't rewind the build if invocation id stays the same ([#18670](#))
- Fix potential memory leak in UI when BwoB. ([#18659](#))
- Properly handle invalid credential files ([#18779](#))
- Report remote execution messages as events ([#18780](#))
- Wait for outputs downloads before emitting local BEP events that reference these outputs. ([#18815](#))
- Fix non-declared symlink issue for local actions when BwoB. ([#18817](#))
- Download directory output for test actions ([#18846](#))

Starlark / Build Language

- Perform builtins injection for WORKSPACE-loaded bzl files. ([#18819](#))

Testing

- Fix test setup script to convey test exit code correctly ([#18514](#))
- Set ``GTEST_SHARD_STATUS_FILE`` in test setup ([#18482](#))

- Actually check if `TEST_SHARD_STATUS_FILE` has been touched ([#18418](#))
- test-setup.sh: Attempt to raise the original signal once more ([#18932](#))

Acknowledgements

This release contains contributions from many people at Google, as well as amishra-u, Andreas Herrmann, Andy Hamon, andyrinne12, Benjamin Lee, Benjamin Peterson, Brentley Jones, Chirag Ramani, Christopher Rydell, Daniel Wagner-Hall, Ed Schouten, Fabian Brandstetter, Fabian Meumertzheim, Greg, Ivan Golub, Jon Landis, Kai Zhang, Keith Smiley, Iripoché, Rasrack, Son Luong Ngoc, Takeo Sawada, Vertexwahn, Yannic.

Bazel 6.2.1 LTS

Release Manager: Pavan Singh Hee Cha

Reviewers: Yun Peng

Status: Approved

Tracking issue: [#18502](#)

Bazel 6.2.1 is a patch LTS release. It is fully backward compatible with Bazel 6.0 and contains selected changes by the Bazel community and Google engineers.

General

- Fix bazel run test_target exit code issue ([#18512](#))
- Use extension rather than local names in ModuleExtensionMetadata ([#18535](#))

Acknowledgements

This release contains contributions from many people at Google, as well as Fabian Meumertzheim.

Bazel 6.2.0 LTS

Release Manager: Kshyanashree Mohapatra

Reviewers: Xudong Yang Yun Peng

Status: Approved

Tracking issue: [#17695](#)

Bazel 6.2.0 is a minor LTS release. It is fully backward compatible with Bazel 6.0 and contains selected changes by the Bazel community and Google engineers.

Highlights

- Added preliminary support for a lockfile for Bzlmod. It's disabled by default; use `--lockfile_mode=update` to enable it. Check flag documentation [here](#). This lockfile contains only Bazel module information; it does not involve module extensions. ([#18143](#))

General

- `@foo` labels can now be used on the command line as the top-level target (that is, `bazel build @foo` now works). Double-dot syntax is now forbidden (`bazel build ../foo` will no longer work). ([#17945](#))
- The `aquery` and `cquery` commands now respect the `--query_file` flag just like the `query` command. ([#17823](#))
- Patch `zlib` to fix compatibility with latest Xcode ([#18000](#))
- `profile`: add `profile_finish_ts` ([#18129](#))
- Use `ctime` in file digest cache key ([#18105](#))
- Include cause when reporting `ActionExecutionException` ([#18185](#))
- Check for the nullness of `AspectValue`. ([#18186](#))
- Force the Bazel server Java runtime to use the root locale ([#17733](#))
- Fix message generation of `ActionExecutionException` ([#18257](#))

Android

- Added coverage metric support to `android_local_test` ([#17467](#))
- Include `build-tools/X.Y.Z/{lib,lib64}/**` in the minimal set of SDK files used by the Android integration tests. ([#17826](#))
- Fix worker and multiplex workers for `DexBuilder` and `Desugar` actions ([#17965](#))

C++ / Objective-C

- Fix `wasm` dynamic library extension crash ([#17765](#))
- Enable C++ deps pruning on Windows when `PARSE_SHOWINCLUDES` is available. ([#17957](#))
- Add attribute `provides = [CcInfo]` to `_cc_proto_aspect` ([#17979](#))
- Expose `cc_proto_aspect` as a top level symbol. ([#17980](#))

CLI

- Handle ``SIGWINCH`` under ``bazel run //some:test`` (i.e. test rules) correctly. ([#17760](#))
- Use less subshells and ``tee`s` in running tests with ``bazel run``. ([#17869](#))

Configurability

- Allow multiple matching `select` branches if they resolve to the same value ([#18066](#))
- `cquery --output-starlark`: better alias support ([#17786](#))
- Fix crashes on `select()`able `target_compatible_with` attributes. ([#18135](#))
- Add "did you mean?" suggestions for typoed rule attribute names ([#17855](#))
- Turn `applicable_licenses` on `platform()` into a no-op. ([#18089](#))

External Dependencies

- Added ``native.module_name()`` and ``native.module_version()`` to allow BUILD macro authors to acquire information about which Bazel module the current repo is associated with. ([#17893](#))
- Added a new ``max_compatibility_level`` attribute to the ``bazel_dep`` directive, which allows version selection to upgrade a dependency up to the specified compatibility level. ([#18178](#))
- Added ``module_ctx.is_dev_dependency``, allowing an extension to see whether a tag is specified to be a dev dependency. ([#17934](#))
- Added ``module_ctx.extension_metadata``, allowing the extension to specify the list of repos the root module can ``use_repo`` on. If the root module violates this list, Bazel will print out a Buildozer command to automatically fix the `use_repo` directive. ([#18174](#))
- Fixed an issue where WORKSPACE and WORKSPACE-loaded .bzl files couldn't see the Bzlmod root module's mappings when Bzlmod is enabled. ([#17818](#))
- Added suggestions to invalid tag class errors. ([#17896](#))
- Workspace relative paths are now supported in `--override_module`. ([#17906](#))
- Fail early if ``use_extension`` has a bad label ([#17926](#))
- Added ``dev_dependency`` to ``register_{execution_platforms,toolchains}``. ([#18259](#))

Java

- Silence `setlocale` warnings in Java stub ([#17741](#))

- Split darwin `java_tools` to x86_64 and arm64 ([#18197](#))
- Upgrade Azul JDK 17 archives. ([#17852](#))
- Add `version` to `JavaRuntimeInfo` ([#17913](#))
- Update default `java_toolchain.bzl` to fix a regression ([#18225](#))
- Updated `java_tools` to v12.3 to fix BazelTestRunner compatibility with jre8 ([#18328](#))

Python

- Removed temporary module space created for zip-based binaries ([#17764](#))
- Set envvar for runfiles manifest, not runfiles dir, when using a manifest ([#18133](#))

Remote Execution

- Skip empty directories instead of throwing in the prefetcher. ([#17718](#))
- Remove `actionId` from `'RemoteFileArtifactValue'`. ([#17724](#))
- Correctly handle templated tree artifacts in the prefetcher. ([#17735](#))
- [credentialhelper] Ignore failing to provide request ([#17745](#))
- [remote] upload: treat `'ALREADY_EXISTS'` as success ([#17732](#))
- Fix data race in prefetcher. ([#17744](#))
- Clear all remote metadata if any of them are evicted from remote cache ([#17770](#))
- Fix seeking of empty chunkers. ([#17830](#))
- Rework `ByteStreamUploader` early return logic. ([#17832](#))
- Avoid unnecessary copying when building Merkle trees. ([#17930](#))
- Cache Merkle trees for tree artifacts. ([#17998](#))
- Deduplicate concurrent computations of the same Merkle tree. ([#18015](#))
- Add exception message to 'failed to create output directory' ([#18067](#))
- Prevent failures creating output directories ([#18062](#))
- Gracefully handle output symlinks with BwoB ([#18106](#))
- Automatically retry the build if encountered remote cache eviction error ([#18171](#))
- Allow remote retry max delay to be user configurable ([#18061](#))
- Rename `--experimental_remote_grpc_log` to `--remote_grpc_log` ([#18193](#))
- Construct `TreeArtifactValues` on multiple threads. ([#18194](#))
- Fix crash on multiple output symlinks with BwoB ([#18226](#))

Starlark / Build Language

- Relax JSON parser, ensure duplicate keys are overridden ([#17868](#))

Acknowledgements

This release contains contributions from many people at Google, as well as Adam Lavin, Alessandro Patti, Ben Lee, Benjamin Lee, Benjamin Peterson, Brentley Jones, Ed Schouten, Ezekiel Warren, Fabian Meumertzheim, George Gensure, Joel Jeske, Julio Merino, Konstantin Erman, Patrick Balestra, Son Luong Ngoc, Yannic Bonenberger.

Bazel 4.2.4 LTS

Release Manager: Kshyanashree Mohapatra

Reviewers: Xudong Yang Yun Peng

Status: Approved ▾

Tracking issue: [#18033](#)

Bazel 4.2.4 is a patch LTS release. It is fully backward compatible with Bazel 4.0 and contains selected changes by the Bazel community and Google engineers.

General

- Bump minimum supported macOS versions to 10.13 ([#18047](#))
- Patch zlib to fix compatibility with Xcode 14.3 ([#18041](#))

Acknowledgements

This release contains contributions from many people at Google, as well as Daniel Wagner-Hall.

Bazel 6.1.2 LTS

Release Manager: Kshyanashree Mohapatra

Reviewers: Xudong Yang Yun Peng

Status: Approved ▾

Tracking issue: [#18031](#)

Bazel 6.1.2 is a patch LTS release. It is fully backward compatible with Bazel 6.0 and contains selected changes by the Bazel community and Google engineers.

General

- Patch zlib to fix compatibility with Xcode 14.3 ([#18039](#))
- Use ctime in file digest cache key ([#18101](#))
- Turn applicable_licenses on platforms into a no-op ([#18082](#))

Acknowledgements

This release contains contributions from many people at Google, as well as Fabian Meumertzheim.

Bazel 5.4.1 LTS

Release Manager: Kshyanashree Mohapatra

Reviewers: Xudong Yang Yun Peng

Status: Approved ▾

Tracking issue: [#18032](#)

Bazel 5.4.1 is a patch LTS release. It is fully backward compatible with Bazel 5.0 and contains selected changes by the Bazel community and Google engineers.

General

- Bump minimum supported macOS versions to 10.13 ([#17977](#))
- Patch zlib to fix compatibility with Xcode 14.3 ([#17996](#))
- Use ctime in file digest cache key ([#18115](#))

Acknowledgements

This release contains contributions from many people at Google, as well as Daniel Wagner-Hall and Fabian Meumertzheim.

Bazel 6.1.1 LTS

Release Manager: Kshyanashree Mohapatra

Reviewers: Xudong Yang Yun Peng

Status: Approved ▾

Tracking issue: [#17212](#) (no new issue for 6.1.1)

Bazel 6.1.1 is a patch LTS release. It is fully backward compatible with Bazel 6.0 and contains selected changes (for bug fixes) by the Bazel community and Google engineers.

C++ / Objective-C

- Revert addition of `-dead_strip` in default opt link flags for darwin ([#17686](#))
- Fix wasm dynamic library extension crash ([#17698](#))

Acknowledgements

This release contains contributions from many people at Google, as well as Ezekiel Warren, Fabian Meumertzheim, and Keith Smiley.

Bazel 6.1.0 LTS

Release Manager: Kshyanashree Mohapatra

Reviewers: Xudong Yang Yun Peng

Status: Approved

Tracking issue: [#17212](#)

Bazel 6.1.0 is a minor LTS release. It is fully backward compatible with Bazel 6.0 and contains selected changes by the Bazel community and Google engineers.

General

- Do not count tests as failed that have not started ([#17269](#))
- Add an `--incompatible_strict_conflict_checks` alias for `--experimental_strict_conflict_checks``. ([#17589](#))
- Rerun the artifact conflict check when `--incompatible_strict_conflict_checks` changes. ([#17592](#))
- Add exec transition to Java version of `py_test` ([#17594](#))

Android

- Multiplex worker fixes and tests for Android busybox tools ([#17371](#))
- Make desugar dependencies deterministic ([#17217](#))

Apple / Xcode

- Allow the timeout value for execute calls on a mac to be set via an environment variable

C++ / Objective-C

- Use `xcrun` to invoke `install_name_tool` ([#17253](#))
- Add sanitizer features to `unix_cc_toolchain_config` ([#17263](#))
- Rulesets can start using the new optional form of `find_cpp_toolchain` now and benefit from the toolchain actually being optional once that has been implemented in a future version of Bazel. ([#17331](#))
- Enable `archive_param_file` feature by default for Windows ([#17344](#))
- Only try to create groups of test actions in the ui. ([#17343](#))
- Find `'libtool'` when using `'BAZEL_USE_CPP_ONLY_TOOLCHAIN'`. ([#17327](#))
- Add `-dead_strip` in default opt link flags for darwin ([#17312](#))
- Fix a `cc_binary` bug for `cc_shared_library` on Windows which prevented `cc_shared_libraries` from working across packages ([#17445](#))
- Remove O1 from sanitizer feature flag defaults ([#17439](#))
- Allow `-v` to libtool ([#17459](#))
- Fix RPATHs for cc toolchain solib when sibling layout is used ([#17506](#))
- Add `--host_features` flag ([#17528](#))
- Only fetch `@remote_coverage_tools` when collecting coverage ([#17512](#), [#17287](#))
- Bump minimum supported macOS versions to 10.13 ([#17570](#))
- Allow `.wasm` cc executable and dynamic library extension ([#17440](#))

Configurability

- Do not clear `--platforms` on no-op change to `--cpu` ([#17273](#))
- Set `--experimental_action_listeners` to default in `'exec'` config ([#17284](#))
- Fix Bazel 6.0 crash regression when incompatible target skipping mixes with `--auto_cpu_environment_group` ([#17613](#))

Coverage

- Emit LLVM coverage for source file paths with a `'tmp'` segment ([#17324](#))

External Dependencies

- Fix `'git_repository'` with the latest version of Git. ([#17247](#))
- Add `'SpellChecker'` suggestions for common Bzlmod errors ([#17216](#))
- Accept tildes in `--override_module` ([#17313](#))
- Test and fix root symlink edge case in runfiles library ([#17365](#))
- Fix Bash `'runfiles_current_repository'` for tools ([#17329](#))
- Fix label unambiguous canonical form to correctly report non-visible repo names ([#17326](#))

- Prettify labels in action progress messages with Bzlmod ([#17278](#))
- Do not recommend `shallow_since` for `git_repository` ([#17356](#))
- Bump version number in runfiles.bash init code ([#17539](#))
- Properly report repo fetch progress during main repo mapping computation ([#17544](#))

Java

- Register JavaCompileActionContext in Bazel ([#17325](#))
- Fix Java coverage collection with Java 8 runtime ([#17491](#))
- Make AutoBazelRepositoryProcessor compatible with Java 8 ([#17504](#))

Local Execution

- Added support for user-specified resource types in the resource manager. The available amount of extra resources can be specified using the new `--local_extra_resources=<resourcename>=<amount>` command line flag, which is analogous to the existing `--local_cpu_resources` and `--local_memory_resources` flags. ([#17229](#))
- Allow Java coverage collection for external targets. ([#17360](#))
- Add support for wrapping system streams in `WorkRequestHandler` ([#17583](#))

Remote Execution

- Fix an issue where some intermediate outputs are downloaded with minimal mode when doing `bazel test` ([#17234](#))
- Make download progress not jumpy by always emitting trailing zero ([#17290](#))
- Upload all logs in BEP even with minimal upload ([#17299](#))
- Do the AC integrity check for disk part of the combined cache. ([#17309](#))
- Improve BES uploader performance by not querying remote cache but always using bytestream protocol ([#17352](#))
- Improve error message for concurrent modifications ([#17396](#))
- Make bazel coverage work with minimal mode ([#17397](#))
- Make Bazel more responsive and use less memory when `--jobs` is high ([#17398](#))
- Remove usage of gRPC Context cancellation in the remote execution client. ([#17438](#))
- Reduce flakiness on Windows for BwOB tests ([#17513](#))
- Exit with code 39 if remote cache evicted blobs that Bazel need during an invocation ([#17496](#))
- Fix symlink file creation overhead ([#17488](#))
- Allow following builds to continue when remote cache evicted files ([#17538](#), [#17601](#), [#17605](#))
- Report background downloads for BwOB ([#17619](#))

- Add flag `--experimental_writable_outputs` to allow Bazel generate writable outputs ([#17617](#))

Starlark / Build Language

- Allow ``map_each`` to return ``None`` in ``TemplateDict#add_joined`` ([#17288](#))
- Allow ``TemplateDict#map_each`` callback to return a list of strings ([#17306](#))
- Added ``native.package_relative_label`` function ([#17435](#))

Acknowledgements

This release contains contributions from many people at Google, as well as Ben Lee, Brentley Jones, Daniel Wagner, Ed Schouten, Ezekiel Warren, Fabian Meumertzheim, Keith Smiley, Krzysztof Naglik, Mathieu Olivari, Ted Kaplan, UebelAndre, Ozio85, Yannic, lunch-glide-pepper, and yuzhy8701.

Bazel 5.4.0 LTS

Release Manager: Kshyanashree Mohapatra

Reviewers: Xudong Yang [LGTM] Yun Peng Ivo Ristovski List

Tracking issue: [#16663](#)

Bazel 5.4.0 is a minor LTS release. It is fully backward compatible with Bazel 5.0 and contains selected changes by the Bazel community and Google engineers.

- Added ``rlocationpath(s)`` functions suitable for the ``Rlocation`` function offered by runfiles libraries for smoother transition from Bazel 5.x.x to Bazel 6.0.0 ([#16668](#))
- ``package(default_package_metadata=[...])`` is now the preferred alternative to ``default_applicable_licenses``. ``default_applicable_licenses`` will be removed in a future release. ([#16892](#))
- Fixed hanging issue when Bazel fails to upload action inputs. ([#16819](#))
- Upgraded google-auth-library-oauth2-http dependencies and fixed transitive dependency on opencensus-contrib-http-util 0.31.0. ([#15639](#))
- Fixed coverage generation for C++ when multiple files with the same name are present. ([#16672](#))
- Moved `analysis_test` into `testing.analysis_test`, an experimental function intended to be used only by the rules_testing repository. The move makes it easier to support both Bazel 5.4.0 and Bazel 6.0.0. ([#16702](#))
- Keep credentials obtained from a credential helper cached across build commands. ([#16884](#))
- Add `'toolchain'` parameter to `actions.{run,run_shell}` ([#16964](#))

Acknowledgements

This release contains contributions from many people at Google, as well as Andreas Fuchs, Benjamin Peterson, Brentley Jones, Dan Fleming, Danny Wolf, Emil Kattainen, Fabian Meumertzheim, Juh-Roch, Keith Smiley, Krzysztof Naglik, Niyas Sait, Noa Resare, Oliver Eikemeier, Peter Mounce, Philipp Schrader, Ryan Beasley, Thi Doãn, Yannic, Zhongpeng Lin.

Bazel 6.0.0 LTS

Release manager: Kshyanashree Mohapatra

Reviewers: Xudong Yang, Yun Peng, Lyra Levin

Tracking issue: [#16159](#)

Bazel 6.0 is a major LTS release. It contains new features and backwards incompatible changes.

Highlights

- [Bzlmod](#), the new external dependency system in Bazel, is now generally available (though still disabled by default). Use `--enable_bzlmod` to enable it.
- Toolchain types may now be [optional](#), in addition to mandatory.
- The author of a `.bzl` file can now call `visibility()` to declare a [load visibility](#) for that file. Load visibility is a new concept that governs which other `.bzl` and `BUILD` files may load `.bzl` code.

General

- **[Incompatible]** `genrule` is switched to use exec transition instead of host. This can break targets with hardcoded output paths. To avoid using hardcoded paths use make variables, see [Predefined source/output path variables](#).
- The Build Event Protocol now contains file digests and sizes along with the file name and URI.
- Introduced a new canonical label literal syntax `@@canonical_repo_name//foo:bar` that skips repository mapping. When Bzlmod is enabled, calling `str()` on a Label object will return literals of this form.
- **[Incompatible]** Added a new flag `--incompatible_unambiguous_label_stringification` (defaults to `true`), which causes labels in the main repo to stringify into unambiguous forms starting with an `@`. ([#15916](#))
- The `some` function in `query` (<https://bazel.build/versions/6.0.0/query/language#some>) now contains an optional parameter that specifies the maximum number of targets to return.

Android

- D8 is used for dexing and Java 8 desugaring.
- Bazel uses the D8 jar from Maven instead of the Android SDK.
- D8 dependency updated to v. 3.3.28.
- Worker mode for D8 desugaring and dexing is supported.
- `--persistent_multiplex_android_tools`` added to enable persistent multiplex workers for Android resource processing, dexing, and desugaring.
- Merging "uses-permissions" tags in Android manifests available with `--merge_android_manifest_permissions``
- Start of DX removal: Migrate `main_dex_list_creator`` to D8.
- Deleted the `--extra_proguard_specs`` Blaze flag.

Apple / Xcode

- Bazel now matches remote and local xcode versions by the most granular version.
- Introduced an experimental static library linking API under `apple_common.link_multi_arch_static_library``.
- Bazel now no longer includes system headers on macOS in coverage reports ([#14969](#)).

C++ / Objective-C

- Bazel no longer stamps `cc_common.link`` actions for tool dependencies.
- Switched `cc_test``, `cc_binary`` and `cc_library`` implementations to Starlark.
- `cc_test`` will now link statically when `_targeting_ Windows` regardless of host platform (rather than always linking statically when Windows is the `_host_`).
- ****[Incompatible]**** Split up the C++ archive from the C++ link action and set `CppArchive`` as mnemonic.
- ****[Incompatible]**** `objc_library`` now requires `CcInfo`` in its deps. If this breaks you, add empty `CcInfo()` to your rule.
- The `@bazel_tools//tools/cpp:compiler`` flag now has the value `gcc`` if the configured compiler is detected to be gcc rather than the generic value `compiler``. A branch for `gcc`` may have to be added to `select`` statements that do not have a default case that handles gcc appropriately.
- The `@bazel_tools//tools/cpp:compiler`` flag now has the value `clang`` for the auto-configured Xcode toolchain rather than the generic value `compiler``. A branch for `clang`` may have to be added to select statements that do not have a default case that handles this toolchain appropriately.
- ****[Incompatible]**** `include_prefix`` and `strip_include_prefix`` for cc rules will no longer accept an empty string (see [code](#)); use a dot instead. ([#17039](#))

Configurability

- `cquery`'s new [--output=files`] (https://bazel.build/docs/cquery#files-output) lists the output files that targets produce.`
- `[genquery.scope`]` (<https://bazel.build/reference/be/general#genquery.scope>) is [faster] (<https://github.com/bazelbuild/bazel/commit/e3f319e19dc4e828c78b76cab8aaea3f38b488e8>).
- New `--experimental_output_directory_naming_scheme=diff_against_baseline`` [may] (<https://github.com/bazelbuild/bazel/issues/14023#issuecomment-1093216554>) improve build speed for builds that use [transitions] (<https://bazel.build/extending/config#user-defined-transitions>).
- [Incompatible target skipping] (<https://bazel.build/extending/platforms#skipping-incompatible-targets>) [works] (<https://github.com/bazelbuild/bazel/issues/14867>) when dependencies lack platform-appropriate toolchains.
- Enable `--use_top_level_targets_for_symlinks`` so targets with [transitions] (<https://bazel.build/extending/config#user-defined-transitions>) are more likely to appear in ``bazel-out/.../bin``.
- ``bazel config`` now only reports info from the last build. To compare configurations across multiple builds, redirect ``bazel config`` output to a file and run your favorite diff tool.
- ``cquery`` proto formatted output now includes more information about configurations. This deprecates the ``configuration`` field of ``AnalysisProtosV2.ConfiguredTarget``, and adds a new field, ``configuration_id``, to be used instead.
- **[Incompatible]** `[--incompatible_override_toolchain_transition`]` (<https://github.com/bazelbuild/bazel/issues/11584>) is now always set and will be removed in the future. ``--noincompatible_override_toolchain_transition`` has no effect and the value of the ``incompatible_use_toolchain_transition`` parameter in ``aspect()`` and ``rule()`` builtins is ignored.
- `[`select()`) (https://bazel.build/docs/configurable-attributes): improved support for [`constraint_value`] (https://bazel.build/reference/be/platform#constraint_value) with new [alias] (https://bazel.build/reference/be/general#alias) and [specialization] (https://bazel.build/reference/be/functions#select) support`
- `[`select()`) (https://bazel.build/docs/configurable-attributes) [no longer] (https://github.com/bazelbuild/bazel/commit/2570fe1119b2462ba10790d3bedf53699baecdf5) reports inaccurate "Duplicate label" errors.`
- Bazel now correctly configures shell actions for multiplatform builds.
- **[Incompatible]** `--incompatible_use_platforms_repo_for_constraints` is now always set and will be removed in the future. ([#8622s](#))

External Dependencies

- **[Incompatible]** ``--noincompatible_disable_managed_directories``, and with that, ``workspace(managed_directories=)`` is not supported anymore.

- The ``get_child`` method of ``path`` now accepts an arbitrary number of relative path strings as positional arguments.
- The ``new_git_repository`` repository rule has been deprecated in favor of ``git_repository``. ([#16064](#))

Java

- Packaging support for deploy JAR embedded JDK files (hermetic Java).
- Make ``ijar`` / ``java_import`` preserve classes with ``@kotlin.Metadata`` annotations.
- Singlejar accepts runtime ``Created-By`` field.
- Change singlejar metadata to report ``Created-By`` Bazel.
- Record hermetic packaged JDK modules file size in deploy JAR manifest ``JDK-Lib-Modules-Size`` attribute.

Local Execution

- Made multiple improvements to dynamic execution to manage local machine resources: ``--experimental_dynamic_exclude_tools``, ``--experimental_dynamic_load_factor``, support for ignoring certain errors in one branch, and better prioritization of local-only actions.
- Fixed a race condition in dynamic execution that caused occasional crashes.
- The Linux sandbox can now run in a chroot jail only allowing access to allowlisted directories ([#13279](#))
- Added more data to the profile: system load average (``--experimental_collect_load_average_in_profiler``), worker memory usage (``--experimental_collect_worker_data_in_profiler``), system network usage (``--experimental_collect_system_network_usage``), and system CPU/memory metrics.
- Added option to include spawn metrics in execution log (``--experimental_execution_log_spawn_metrics``)
- It is now possible to use pseudoterminals in the Linux sandbox ([#14072](#))
- Added the ``--incompatible_sandbox_hermetic_tmp`` flag that makes sandboxed actions have separate ``/tmp`` directories (Linux only). ([#16336](#))

Packaging

- Further deprecation and removal of ``pkg_tar``. Stop supporting legacy use of ``files`` attribute, where it could be a list of labels instead of a map of paths to labels.
- Removed ``--incompatible_no_build_defs_pkg`` flag. It never fulfilled its purpose because ``--all_incompatible_changes`` would never set it. The last rule it gated (``pkg_tar``) is scheduled to be removed in Bazel 6.x.
- ``pkg_tar(syminks)`` has been removed. Users needing that feature should migrate to ``@rules_pkg``.
- Remove ``mtime`` options from ``pkg_tar``. Users should migrate to ``@rules_pkg``.
- The legacy ``pkg_tar`` no longer supports the ability to untar and repackage an input tar file (``deps`` attribute). Users needing that capability must switch to ``rules_pkg``.

Persistent Workers

- Added support for remote persistent workers ([#16362](#))
- Added `--experimental_worker_multiplex_sandboxing`` flag that controls whether to sandbox multiplex workers that support it.
- Added `--experimental_total_worker_memory_limit_mb`` flag that enables evicting idle workers when workers use too much memory.
- Added `--experimental_worker_as_resource`` that prevents creation of workers before they have the resources to run.
- Added verbosity field to worker protocol for better debugging.
- Added `--experimental_worker_strict_flagfiles`` flag that makes improper use of `@`-files in worker definitions a failure.
- Worker directories are now only created when actually needed.
- Made the JSON worker protocol newline-delimited ([#14219](#))
- The worker JSON protocol now ignores unknown fields.
- **[Incompatible]** Flag `--experimental_worker_allow_json_protocol`` removed (defaulted to true).

Python

- Add ``six`` to deps of ``has_services=1`` ``py_proto_library`s`.
- Add `--incompatible_build_transitive_python_runfiles`` alias. ([#16303](#)). ****This will be enabled next release****.
- Added ``CurrentRepository`` method to ``bazel_tools`` Python runfiles library for Bzlmod support ([#16341](#)).
- Make `--incompatible_disallow_legacy_py_provider`` a no-op; using the modern ``PyInfo`` provider is now always required.
- Remove potentially misleading warning about potential Python version mismatch ([#15783](#)).
- Reduce import overhead during bootstrapping ([#15572](#)).
- Set ``PYTHONSAFEPATH`` to prevent escaping runfiles ([#15701](#)).
- Allow larger transitive dependencies by deduplicating ``PYTHONPATH`` ([#15648](#)).
- Optimize memory usage when building Python zip files ([#15037](#)).
- Fix: ``py_test`` now collect coverage for C++ binaries in data ([#15098](#)).
- Fix: Allow spaces in Windows path to binary ([#14851](#)).
- Fix: Allow Windows launcher to customize executable name ([#14343](#)).
- Fix: ``py_binary`` can now be invoked through an ``sh_binary`` ([#11997](#)).
- Various preparations to drop Python 2 support and re-implement rules in Starlark ([#15684](#), [#15897](#)).

Remote Execution

- Add support for fetching RPC credentials from credential helper.

- The deprecated `--remote_allow_symlink_upload` flag has been removed. Symlinks in local action outputs are always permitted, even with remote caching. Whether they're uploaded as symlinks or as the files/directories they point to is still determined by the `--incompatible_remote_symlinks` flag.
- **[Incompatible]** `--incompatible_remote_downloader_send_all_headers` is flipped to `true`. ([#16356](#))
- **[Incompatible]** `GrpcRemoteDownloader` only includes relevant headers instead of sending all credentials (see [security advisory](#)).
- **[Incompatible]** `--incompatible_remote_results_ignore_disk` is flipped to `true`. ([#15148](#))
- Improve `--remote_download_minimal` by correctly downloading necessary outputs for `bazel run` and `bazel test`.
- Add flag `--experimental_remote_mark_tool_inputs` to support remote persistent workers.
- **[Incompatible]** Deprecated flag `--remote_allow_symlink_upload` is deleted.
- Add flag `--experimental_remote_build_event_upload` to control how Bazel upload local outputs referenced by BEP to remote cache.

Starlark / Build Language

- `\a \b \f \w` escape sequences are now allowed in Starlark.
- `provider()` now allows you to specify a custom constructor (`init`) for performing pre-processing and validation of field values. For more information, see [Custom initialization of providers](#).
- **[Incompatible]** The `--incompatible_existing_rules_immutable_view` flag has been flipped to true. ([#13907](#))
- Starlark test rules can use the new `inherited_environment` parameter of `testing.TestEnvironment` to specify environment variables whose values should be inherited from the shell environment.
- Aspects can now define and use exec groups using the same API as rules.
- **[Incompatible]** `analysis_test` moved into `testing.analysis_test`.
- `SourceManifestAction` now supports `Action.content`.
- Added `struct`, `json`, `proto`, and `depset` to the starlark environment of Bazel's `cquery` (`--output=starlark`) command
- **[Incompatible]** The `name` parameter is removed from the `rule` function. ([#16301](#))
- **[Incompatible]** In `package_group`'s [packages](#) attribute, the syntax `"//..."` now refers to all packages in the same repository as the package group, rather than all packages everywhere. The new item `"public"` can be used instead to obtain the old behavior. In `bazel query --output=proto` (and `--output=xml`), the `packages` attribute now serializes with the leading double slash included (for instance, `//foo/bar/...` instead of `foo/bar/...`). See issues [#16323](#), [#16355](#), and [#16391](#) for more information and for the flags to temporarily opt out of these changes.

- ``package(default_package_metadata=[...])`` is now the preferred alternative to ``default_applicable_licenses``. ``default_applicable_licenses`` will be removed in a future release.
- Aspects can now have parameters of types: [integer](#) and [boolean](#)
- Command-line aspects can now have parameters of types: string, integer and boolean. ``-aspects_parameters`` flag is introduced to pass their values [\[commit\]](#).

Acknowledgments

This release contains contributions from many people at Google, as well as Adam Azarchs, Adam Singer, Adam Wolf, Albert Lloveras, Alessandro Patti, Alex Eagle, Alex Scott, Alex Torok, Andreas Fuchs, Andreas Herrmann, Andrew Katson, Andrew Klotz, arun.sampathkumar, arunkumar9t2, Ast-x64, Benedek Thaler, Benjamin Lee, Benjamin Peterson, Bo Zhang, Bohdan Vanieiev, Bradley Burns, Brandon Duffany, Brandon Jacklyn, Brentley Jones, Chad Miller, Charles-Francois Natali, Chris Clearwater, Chris Fredrickson, Christopher Rydell, Christopher Sauer, ckiffel, Cristian Hancila, crydell-ericsson, Dan Fleming, Daniel Wagner-Hall, Danny Wolf, David Ostrovsky, David Sanderson, Denys Kurylenko, dhmemi, Dimi Shahbaz, divanorama, Ed Schouten, Emil Kattainen, Eric Song, Eric Wendelin, Fabian Brandstetter, Fabian Meumertzheim, Fredrik Medley, George Prekas, gkgoat1, gkorlam, Greg Roodt, Gregory Fong, Grzegorz Lukasik, Halil Sener, Hannes Kufler, homuler, Igor Nazarenko, James Broadhead, Jason Tan, Jay Bazuzi, Jeremy Volkman, jheaff1, Jiawen Chen, Joel Williamson, John Laxson, John Millikin, Jon Shea, Jonathan Gerrish, juanchoviedo, Kaiqin Chen, Keith Smiley, Ken Micklas, Kevin Lin, Kirill Zabelin, Kiron, Krishna Ersson, Krzysztof Naglik, lihu, Luc Bertrand, Luis Fernando Pino Duque, Marek uppa, Matt Clarkson, Matt Mackay, Michael P. Nitowski, Mikhail Balabin, Mostyn Bramley-Moore, Nick Korostelev, Nitesh Anandan, Niyas Sait, Noa Resare, Olek Wojnar, Oscar Bonilla, Patrick Balestra, Paul Tarjan, Peter Mounce, Philipp Schrader, Pras Velagapudi, Rahul Butani, Rajeshwar Reddy T, Rifqi Mulya Fahmi, Roman Salvador, rustberry, Ryan Beasley, Ryan Schmidt, Sahin Yort, Saleem Abdulrasool, Sara Adams, Shuai Zhang, Simon Bjorklen, Son Luong Ngoc, Stephan Wolski, Steve Vermeulen, Stiopa Koltsov, Takeo Sawada, Tao Wang, Ted Kaplan, Tetsuo Kiso, Thi Doan, Thomas Chen, Thomas Zayouna, ThomasCJY, Timothe Peignier, Tom de Goede, Tomas Volf, UebelAndre, Ulf Adams, Ulrik Falklof, vardaro, Vasilios Pantazopoulos, Vladimir Tagakov, Waleed Khan, William Muir, Xavier Bonaventura, Xiangquan Xiao, Yannic Bonenberger, Yesudeep Mangalapilly, Yuval K, Zhongpeng Lin.

For older releases please see the archive docs:

- [All releases until EOY 2022 \(Bazel 5.3 LTS\)](#)
- [All releases until EOY 2021 \(Bazel 4.2 LTS\)](#)