Bazel Release Announcements

Archive of announcements up to EOY 2022

New doc: Bazel Release Announcements

Bazel 4.2.3 LTS

Release Manager: Kshyanashreem
Reviewers: Xudong Yang Yun Peng

Bazel 4.2.3 is patch LTS release. It is fully backward compatible with Bazel 4.0 and contains selected changes(i.e security fix) by the Bazel community and Google engineers.

Remote execution:

- Update GrpcRemoteDownloader to only include relevant headers.
  (#PR 16439)

Bazel 5.3.2 LTS

Release Manager: Kshyanashreem
Reviewers: Xudong Yang Yun Peng

Bazel 5.3.2 is patch LTS release. It is fully backward compatible with Bazel 5.0 and contains selected changes by the Bazel community and Google engineers.

Remote execution:

- Update GrpcRemoteDownloader to only include relevant headers.(16439)
- Avoid unnecessary iteration on action inputs.(16118)
- Revert to using `matches` instead of `find` with `remote_download_regex`(16476)

Acknowledgments:

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

Bazel 5.3.1 LTS

Release Manager: Kshyanashreem
Reviewers: Xudong Yang Yun Peng

Bazel 5.3.1 is patch LTS release. It is fully backward compatible with Bazel 5.0 and contains selected changes by the Bazel community and Google engineers.

C++

- Fix local execution of external dynamically linked cc_* targets (16214)

Remote Execution

- Do not crash on URIs without a host component. (16184)
- Add profiler task for calling a credential helper. (16226)

Configurability

- add change to allow blaze info to skip Starlark build settings that start with --no prefix (16258)

Acknowledgments

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, Gowroji Sunil, Juh-Roch, Keith Smiley, Niyas Sait, Noa Resare, Oliver Eikemeier, Peter Mounce, Philipp Schrader, Ryan Beasley, Thi Doãn, Xùdōng Yáng, Yannic, Zhongpeng Lin, amberdixon, hvadehra, kshyanashree, oquenchil.

Bazel 5.3 LTS

Release manager: kshyanashreem
Tracking issue: #15685
Reviewers: Xudong Yang Yun Peng Radhika Advani Jason Dobies Lyra Levin

Bazel 5.3 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.

General

- junitrunner: Report suppressed and ignored tests as skipped (13717)

- Print remote execution message when the action times out (15710)
- Preserve --experimental_allow_unresolved_symlinks in exec cfg (15702)

Apple / Xcode

- Define cc-compiler-darwin in Xcode toolchain (14796)
- Fix alwayslink in objc_import (15313)

C++

- Fix fail message construction in cc_shared_library (14697)
- Optionally enable LLVM profile continuous mode (15166)
- Increase osx_cc_configure timeouts (15877)
- Make cpp assembly file extensions case sensitive again (14131)
- Fix rpath for binaries in external repositories(16008)

Configurability

- Allow string_list flags to be set via repeated flag uses (14911)
- Add --output=files mode to cquery (15552)

Coverage

- g++ fpermissive compilation error for strdup on musl when building from source(15729)
- target pattern file: allow comments(15903)

External Dependencies

- Unify URL/URLs parameter code across http_archive, http_file, http_jar(15408)
- Add is_root struct field to bazel_module (15792)
- Add util for finding credential helper to use (15707)
- [credential helper] Add types to communicate with the subprocess (15803)
- Fix behavior of `print()` in module extensions (15795)
- Add is_root struct field to bazel_module (15792)
- [remote] Improve .netrc test in RemoteModuleTest (15902)
- Fail the build on repository rule errors in a module extension (15796)
- Add factory for creating paths relative to well-known roots (15805)
- Replace module.{execution_platforms,toolchains}_to_register with register {execution_platforms,toolchains}(15829)
- Accept tildes in --override_repository (15417)
- http_archive, unrecognised file mode for [Content_Types].xml: 0x80 (9236)
- Fix and test for null value bug when using --repo_env(15618)
- Wire up credential helper to command-line flag(s)(15947)

Java

- Fix string formatting when java_home path is missing. (14686)

Remote Execution

- Chunker: Always seek on the uncompressed stream. (15669)
- RemoteExecutionService: fix outputs not being uploaded (15823)
- Add a flag to force Bazel to download certain artifacts when using --remote_download_minimal(15638)
- [credential helper] Implement invoking credential helper as subprocess(15861)
- Replace uses of cfg = "host" with cfg = "exec" (15785)
- [credential helper] Add parser for flag syntax (15906)
- Docs should mention the new no-remote-cache-upload tag (14518)
- Add CommandLinePathFactory to CommandEnvironment (15905)
- Remote: Fix performance regression in "upload missing inputs" (15890)
- Add netrc support to --bes_backend (15930)
- Move newCredentialHelperProvider into GoogleAuthUtils (15941)
- Propagate the error message when a credential helper fails.(16012)
- Fix an issue that `incompatible_remote_build_event_upload_respect_no_…(16023)
- Refactor combined cache.(16039)

Starlark

- Let starlark executable rules specify their environment (15232)

Acknowledgments

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

Bazel 5.2 LTS

Release manager: ckolli5

Tracking issue: #15210
Reviewers: Xudong Yang, Yun Peng, Radhika Advani, Jason Dobies

Bazel 5.2 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.

General

- Fix running Bazel on Apple Silicon machine under QEMU (15223)
- Catch NumberFormatException while trying to parse thread id (15434)
- Improve the --sandbox_debug error message (15437)
- Set keywords on appropriate lifecycle events (15440)

Apple / Xcode

- Configure Apple crosstool to return a complete target triple that includes minimum OS version and target environment (15266)
- Fix android emulator darwin_arm64 select (15445)

C++

- Remove -U_FORTIFY_SOURCE when thin_lto is enabled (15433)
- Revert "Fixes incorrect install names on darwin platforms (15450)
- Filter libtool warning about table of contents (15448)
- SolibSymlinkAction does not need exec platform or properties (15474)
- Add feature to produce serialized diagnostics files (15600)

Configurability

- Fix cache leak when applying transitions when only a rule's attributes change (15404)
- Fix --use_top_level_targets_for_symlinks with aliases (15446)
- config doesn't error on duplicate --define values (15473)

Coverage

- Collect C++ lcov coverage if runtime object not in runfiles (15299)
- Collect coverage from cc_binary data deps of java_test (15216)
- Collect coverage from cc_binary data deps of py_test (15298)
- Make coverage --combined_report=lcov skip incompatible tests (15471)

External Dependencies

- Bzlmod: throw on json parse exception (15429)
- Add support for .ar archives (and .deb files) (15218)
- Bump the limit of Bazel install base size (15585)
- Upgrade zlib to 1.2.12 (15534)

- Fix a bug that outputs of no-remote actions when used in combination with a disk-cache (15453)

Java

- Include jdk.crypto.mscapi in minimized Windows embedded JDK (15342)

Python

- Update PythonZipper action to use CommandLineItem.CapturingMapFn (15472)

Remote Execution

- Unify sandbox/remote handling of empty TreeArtifact inputs (15449)
- Improved zstd compression support for remote cache/execution (15372)
- Upgrade Google Auth Version (15383)
- Fix checking remote cache for omitted files in buildevent file (15405)
- Add a flag to expose undeclared test outputs in unzipped form (15431)
- Record additional profiling information for remotely executed actions (15443)
- Fix downloading remote execution output files inside output dirs (15444)

Starlark

- Let Starlark tests inherit env variables (15217)

Acknowledgments

This release contains contributions from many people at Google, as well as Andreas Herrmann, Ben Lee, Benjamin Peterson, Brentley Jones, Dan Fleming, Denis Kurylenko, Fabian Meumertzheim, Keith Smiley, Ken Micklas, Mostyn Bramley-Moore, Niyas Sait, Oscar Bonilla, Son Luong Ngoc, Thi Doãn, Yannic, Yuval K, Zhongpeng Lin.

Bazel 5.1 LTS

Release manager: wyv
Tracking issue: #14714

Bazel 5.1 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.

General

- Bazel now finds runfiles in directories that are themselves runfiles. (#14737)
- Symlinks are no longer resolved for `--sandbox_base`. (#14748)
- Flipped `--experimental_worker_allow_json_protocol` to true. (#14749)
- Enabled native support for Windows on arm64. (#14794)

- Added a new `subpackages()` built-in helper function. (#14780)
- Fixed an issue where Bazel could erroneously report a test passes in coverage mode without actually running the test. (#14836)
- Bazel now uses the new Java API `ProcessHandle` to get the PID. (#14842)

Apple / Xcode

- Added support for the `tvos_sim_arm64` toolchain. (#14779)
- Fixed `ctx.fragments.apple.single_arch_cpu` returning incorrect cpu for tools when host cpu and exec cpu are different. (#14751)
- Fixed the default CPU for macOS and iOS. (#14923)
- Corrected `cpu` and `os` values of some Apple-related `local_config_cc_toolchains` targets. (#14995)
- Fixed conflicting actions error when specifying `--host_macos_minimum_os`. (#15068)
- `osx_cc_wrapper` now only expands existing response files. (#15090)

C++

- Added an experimental version of `cc_shared_library`. (#14773)
- Removed uses of `-lstdc++` on Darwin (#14750) and BSD (#14860).
- Added the default `solib` directory to the `rpath` for `cc_import`s with transitions. (#14757)
- `rpath` entries are now normalized to guard against missing default `solib` dir. (#14929)
- `--experimental_cc_implementation_deps` now propagates into exec configs. (#14753)

Configurability

- `alias()` can now `select()` directly on `constraint_value()`. (#14754)
- Added a helper method `use_cpp_toolchain()` to depend on the cc toolchain type. (#14795)
- `Label` instances are now allowed as keys in `select`. (#14755)

Coverage

- Bazel now no longer includes system headers on macOS in coverage reports (#14969).
- Exposed `CoverageOutputGenerator` on a Fragment. (#14997)

External Dependencies

- `UrlRewriter` can now load credentials from `.netrc`. (#14834)
- Added an `arch` field to `repository_os`. (#14835)
- Added support for `WORKSPACE.bzlmod`. (#14813)
- Multiple `use_extension`s are now allowed on the same module extension. (#14945)

- Added `--experimental_repository_cache_urls_as_default_canonical_id` to help detect broken repository URLs. (#14989)
- Added Starlark dependencies to the package `//external`. (#14991)
- Added support for decompressing zstd tar archives in repository rules. (#15087)

Java

- Bazel now ignores a missing `include` directory in JDK distributions. (#14832)

Protocol Buffers

- `protocOpts()` is now publicly accessible. (#14952)

Remote Execution

- The remote module now only waits for background tasks spawned from remote execution. (#14752)
- Postponed the block waiting in `afterCommand` to `BlockWaitingModule`. (#14833)
- Bazel now handles early return of compressed blob uploads. (#14885)
- Changed the default Merkle tree cache size to 1000. (#14984)
- Actions are no longer considered successful and cached if outputs were not created. (#15071)
- Fixed certain crashes by InterruptedException when dynamic execution is enabled. (#15091)

Starlark

- Added new `removeprefix`/`removesuffix` methods to strings. (#14899)

Acknowledgments

This release contains contributions from many people at Google, as well as Andreas Herrmann, Ben Lee, Benjamin Peterson, Brentley Jones, Dan Fleming, Denis Kurylenko, Fabian Meumertzheim, Keith Smiley, Ken Micklas, Mostyn Bramley-Moore, Niyas Sait, Oscar Bonilla, Son Luong Ngoc, Thi Doãn, Yannic, Yuval K, Zhongpeng Lin.

Bazel 5.0 LTS

Release manager: wyv, pcloudy
Tracking issue: #14013

Bazel 5.0 is a major LTS release. It contains new features and backwards incompatible changes.

Highlights

- `--experimental_enable_bzlmod` enables the new external dependency subsystem, Bzlmod.
- The minimum required JDK to run Bazel itself is now OpenJDK 11. Note that compiling binaries using JDK 8 and lower is still supported.
- Java rules now use toolchain resolution. (#7849)
- Starlark now permits `def` statements to be nested (closures) and supports lambda expressions (anonymous functions).

General

- `//visibility:legacy_public` has been removed.
- The `--all_incompatible_changes` flag is now a no-op.
- Specifying a target pattern underneath a directory specified by `.bazelignore` will now emit a warning, not an error.
- The `test` and `coverage` commands no longer return 3 when a test action fails because of a system error. Instead, the exit code reflects the type of system error.
- Bazel will no longer create a `bazel-out` symlink if `--symlink_prefix` is specified: the directory pointed to via the `bazel-out` symlink is accessible via `${symlink_prefix}-out`.
- Removed flag `--experimental_no_product_name_out_symlink`: it is always true.
- Removed `--action_graph` from the `dump` command.
- Removed `--{experimental_,}json_trace_compression`; its value is determined by the profile name.
- Removed `--experimental_profile_cpu_usage`; it is effectively always true.
- `--legacy_dynamic_scheduler` is now a no-op.
- Flipped `--trim_test_configuration` to true. (#6842)
- When using `--allow_analysis_failures` (for example, via bazel-skylib's `analysistest` with `expect_failure = True`), analysis-time failures in aspect implementation functions will now be propagated and saved in `AnalysisFailureInfo`, just like analysis-time failures in rules.
- Added `--experimental_reuse_sandbox_directories` to reuse already-created non-worker sandboxes with cleanup.
- Renamed `--experimental_run_validations` to `--run_validations` and flipped its default to true.
- Removed `--experimental_forward_instrumented_files_info_by_default`, now that this behavior is the default.
- When using MemoryProfiler with multiple GCs via the `--memory_profile_stable_heap_parameters` flag, we do a more precise calculation of heap used at the end of the build. This will generally result in lower values.
- Added support to length-delimited protos as undeclared output annotations.
- Added experimental flag `--experimental_dynamic_skip_first_build` to skip local execution in dynamic execution until there has been a successful build. This allows doing a first clean build from remote only, filling the cache.

Android

- The minimum Android build tools version for the Android rules is now 30.0.0.
- The Android rules' `--use_singlejar_apkbuilder` is now a no-op. SingleJar will always be used to build APKs.
- `--apk_signing_method` now accepts the value `v4`.
- Proguard configs generated from aapt2 link step now exclude path location comments for better determinism.
- Removed obsolete `--incompatible_prohibit_aapt1`.

Apple / Xcode

- `--apple_sdk` has been deleted. It is a no-op.
- The `--incompatible_disable_native_apple_binary_rule` flag has been added which disables the native `apple_binary` rule. Users who need to use `apple_binary` directly (if they cannot use one of the more specific Apple rules) should load it from https://github.com/bazelbuild/rules_apple.
- Allowing the lipo operations to be conditional in the `linkMultiArchBinary` API for Apple binaries. Single architecture slices are now returned through AppleBinaryOutput and the Starlark API.
- If `--experimental_prefer_mutual_xcode` is passed, Bazel will choose the local default (instead of the newest mutually available version) if it's available both locally and remotely.

Build Event Protocol

- In the build event stream, `BuildMetrics.TargetMetrics.targets_loaded` is no longer populated. Its value was always mostly meaningless. `BuildMetrics.TargetMetrics.targets_configured` and `BuildMetrics.ActionSummary.actions_created` now include configured aspect data.
- `--bep_publish_used_heap_size_post_build` is removed. Use `--memory_profile=/dev/null` instead; the `used_heap_size_post_build` field in BEP is now populated when the `--memory_profile` flag is set.
- Added `--bes_header` flag to pass extra headers to the BES server.
- BEP now includes test suite expansions.
- BEP now includes all files from successful actions in requested output groups. Previously, an output group's files were excluded if any file in the output group was not produced due to a failing action. Users can expect BEP output to be larger for failed builds.
- In BEP, `TargetComplete.output_group` has a new field `incomplete` indicating that the `file_sets` field is missing one or more declared artifacts whose generating actions failed.
- When `--experimental_bep_target_summary` is enabled, BEP contains a new event, `TargetSummary`. For each top-level configured target, this event aggregates the results for building the configured target, tests run for that configured target (if any),

and building all top-level aspects applied to that configured target. This flag will eventually default to enabled.

C++ / Objective-C

- The Starlark method `generate_dsym` in `objc` fragment has been deleted. Please use the equivalent `apple_generate_dsym` in `cpp` fragment instead.
- Removed `--incompatible_objc_compile_info_migration`. (#10854)
- Removed `--incompatible_objc_provider_remove_compile_info`. (#11359)
- Made gcov optional in `cc_toolchain` tools.
- Added a new flag, `--incompatible_enable_cc_test_feature` which switches from the use of build variables to the feature of the same name.

Configurability

- Added `--incompatible_enforce_config_setting_visibility` to make `config_setting` honor the `visibility` attribute (defaulting to `//visibility:public`) (#12932)
- The flag `--toolchain_resolution_debug` now takes a regex argument, which is used to check which toolchain types or targets should have debug info printed. You may use `.*` as an argument to keep the current behavior of debugging every toolchain type / target.

Coverage

- All (instead of just C++) source files are now filtered for coverage output according to `--instrumentation_filter` and `--instrument_test_targets`.
- `genrule.srcs` is now considered a source attribute for coverage.
- Forward coverage-instrumented files from non-tool dependencies by default.
- `label_keyed_string_dict` attributes are now considered when gathering instrumented files for coverage.
- Java branch coverage now applies Jacoco's coverage filters. (#12696)

Java

- Removed `javac_jar` from JavaToolchainInfo.
- Flipped `--incompatible_disallow_resource_jars`, disabling `resouce_jars` attribute on java_library. (#13221)
- Flipped `--incompatible_java_common_parameters`, removing `host_javabase` parameter from `java_common.compile`. (#12373)
- Flipped `--incompatible_use_toolchain_resolution_for_java_rules`. Flags `--javabase` and `--java_toolchain` are noop. (#7849)
- Only native libraries supported by C++ targets in `deps` and `runtime_deps` are collected. No longer collecting `.so` files (#13043) or libraries in the `data` attribute. (#13550)
- `JavaToolchainInfo.jvm_opt` returns a `depset` instead of a list.
- Added a `_direct_source_jars` output group to Java related targets.

- Added an `enable_annotation_processing` option to `java_common.compile`, which can be used to disable any annotation processors passed to `plugins` or in `exported_plugins` of `deps`.
- Removed `java_lite_proto_library.strict_deps` attribute.
- The Starlark implementations of `java_library` and `java_plugin` are now used instead of their native counterparts.
- `java_plugins` can now only be used in the `plugins` and `exported_plugins` attributes. (#14012)
- Javac workers now support cancellation.
- Javac workers now cache the bootclasspath classes.

Packaging

- `pkg_deb` is no longer part of `@bazel_tools//build_defs/pkg:pkg.bzl`. Use https://github.com/bazelbuild/rules_pkg/tree/main/pkg instead.
- Dropped fragile xz support from built in `pkg_tar`. Users requiring xz compression should switch to `bazelbuild/rules_pkg`.

Persistent Workers

- Multiplex persistent workers can now use the JSON protocol (#7998).
- Updated worker protocol to allow request cancellation, and added `--experimental_worker_cancellation` flag to turn on cancellation for workers that implement it (#614).

Python

- The `--incompatible_load_python_rules_from_bzl` flag is now a no-op.
- Relax restriction for "-" in the package name for Python sources. Now `py_binary` and `py_test` targets can have a main source file with "-" in the path.

Remote Execution

- Disk cache can now be used in remote execution. (#13852)
- Merkle tree calculation can be cached with --experimental_remote_merkle_tree_cache. The cache size is unlimited by default but can be controlled by --experimental_remote_merkle_tree_cache_size. (#13879)
- File uploads are scheduled into background if --experimental_remote_cache_async is set. (#13655)
- --remote_max_connections applies to gRPC connections as well. (#14202)
- Remote metadata is saved into action cache if --exprimental_action_cache_store_output_metadata is set. (#13604)
- Data transfers between Bazel and remote server are compressed via zstd if --experimental_remote_cache_compression is set. (#14041)
- Added support for tag no-remote-cache-upload. (#14338)
- Bug fixes and UX improvements.

Query

- Query `output=xml/proto/location` for source files will now show the location of line 1 of the source file (as the new default) instead of its location in the BUILD file.
- Query `--order_output=auto` will now sort lexicographically. However, when `somepath` is used as a top level function (e.g. `query 'somepath(a, b)'`), it will continue to output in dependency order. If you do not want the lexicographical output ordering, specify another `--order_output` value (`no`, `deps` or `full`) based on what ordering you require.
- cquery `--noimplicit_deps` now correctly filters out resolved `cc_toolchain`s.

Starlark / Build Language

- Starlark now permits `def` statements to be nested (closures) and supports lambda expressions (anonymous functions). Recursion is still not permitted.
- `Args.add_all` and `Args.add_joined` can now accept closures in `map_each` if explicitly enabled via the `allow_closure` parameter. Beware that closures may retain large data structures into the execution phase.
- dict.setdefault(key, ...) now fails if dict is frozen, even if it already contains key. This is an incompatible API change. (#12642)
- The deprecated "relative_to_caller_repository" parameter has been removed from the `Label` constructor.
- The undocumented `ctx.expand` feature no longer exists.
- Add `runfiles.merge_all()` for merging a sequence of runfiles objects.
- `runfiles.merge()` and `merge_all()` now respect `--nested_set_depth_limit`. If you hit the depth limit because you were calling `merge()` in a loop, use `merge_all()` on a sequence of runfiles objects instead.
- `native.existing_rule` now returns select values in a form that is accepted by rule instantiation. This is a breaking API change, though the fallout is expected to be small.
- Added `--incompatible_existing_rules_immutable_view` flag to make the `native.existing_rule` and `native.existing_rules` functions more efficient by returning immutable, lightweight dict-like view objects instead of mutable dicts.
- Flipped `--incompatible_disable_depset_items` (#9017). The flag is now a no-op.

Acknowledgments

This release contains contributions from many people at Google, as well as Adam Liddell, Alex Eagle, Andrew Katson, Anthony Pratti, Artem V. Navrotskiy, Austin Schuh, Benedek Thaler, Benjamin Lee, Benjamin Peterson, Brandon Jacklyn, Brentley Jones, bromano, Cameron Mulhern, Christopher Sauer, Cristian Hancila, Dan Bamikiya, Daniel McCarney, Daniel Wagner-Hall, Danny Wolf, Dave Nicponski, David Cummings, David Ostrovsky, Delwin9999, Denys Kurylenko, Dmitry Ivankov, dorranh, ecngtng, Ed Schouten, Eitan Adler, Elliotte Rusty Harold, Eric Cousineau, Ethan Steinberg, Fabian Meumertzheim, FaBrand, Felix Ehrenpfort, Finn Ball, frazze-jobb, Fredrik Medley, Garrett Holmstrom, Gautam Korlam,

George Gensure, goodspark, Grzegorz Lukasik, hvadehra, Ikko Ashimine, Jesse Chan, Joe Lencioni, Johannes Abt, John Laxson, Jonathan Schear, Justus Tumacder, Keith Smiley, kekxv, Kevin Hogeland, Lauri Peltonen, Liu Liu, László Csomor, Marc Zych, Mark Karpov, Masoud Koleini, Mathieu Olivari, Matt Mackay, Mauricio Galindo, Max Liu, Menny Even Danan, Michael Chinen, Nathaniel Brough, Nick Korostelev, Niek Peeters, Nikolay Shelukhin, odisseus, Oleh Stolyar, Olek Wojnar, Olle Lundberg, Omar Zuniga, Paul Gschwendtner, Peter Kasting, Philipp Schrader, Pras Velagapudi, Qais Patankar, Rabi Shanker Guha, Rai, Ron Braunstein, Ryan Beasley, samhowes, Samuel Giddins, Sebastian Olsson, Sergey Tyurin, Steve Siano, steve-the-bayesian, Stiopa Koltsov, tatiana, Tetsuo Kiso, Thi Doãn, Thomas Carmet, ThomasCJY, Timothe Peignier, Timothy Klim, Tobi, Torgil Svensson, Trustin Lee, Ulf Adams, Ulrik Falklof, Uri Baghin, Vaidas Pilkauskas, Vertexwahn, wisechengyi, Wren Turkal, Xavier Bonaventura, Yannic Bonenberger, Yury Evtikhov, Yuval Kaplan, [zqzzq].

For older releases please see the archive docs:

- All releases until EOY 2021 (Bazel 4.2 LTS)