

Draft etcd 3.7.0 Release Blog

Title

etcd 3.7.0 is Now Available

Outline

- Released
 - Major features
 - Download it here
 - Also includes new bbolt, raft
 - Link to install docs
- Features
 - RangeStream
 - Keys-only optimization
 - Performance improvements
 - faster leases
 - interval splitting
 - Unix sockets
 - v3store startup
 - etcdctl timeouts
 - JWT user setting
 - ClientLogLevel
 - new watch metrics
 - highlight: non-blocking client creation (also breaking change)
 - ???
- Upgrade Nodes
 - Deprecating experimental features
 - v2store completely gone
 - Protobuf overhaul
 - ???
- Contributors
 - Leads
 - Other Contributors
 - New Contributors

Text

Announcing etcd v3.7.0

NO FURTHER EDITS HERE, TEXT MOVING TO PR

By **SIG etcd leads** | Wednesday, July 01, 2026

TODO: fix release date

The `date` in this post's front matter is a **placeholder** (2026-07-01).

etcd blog: Update it to the actual etcd v3.7.0 GA date before publishing, then remove this callout.

mirror article: comment out the *date*: front matter and initially merge the article into the Kubernetes k/website repo as *draft: true*

Table of Contents

- [Introduction](#)
- [Major features](#)
- [Features](#)
 - [RangeStream](#)
 - [Keys-only range optimization](#)
 - [Faster, more reliable leases](#)
 - [Interval splitting](#)
 - [Unix socket support](#)
 - [Bootstrap from v3store](#)
 - [etcdctl timeouts](#)
 - [Setting the JWT directly](#)
 - [New watch metrics](#)
- [Performance improvements](#)
- [Upgrading](#)
 - [Deprecating experimental features](#)
 - [v2store completely gone](#)
 - [Non-blocking client creation](#)
 - [Protobuf overhaul](#)
 - [Multiarch containers](#)
 - [Other API changes](#)
- [Bbolt v1.5.0](#)

- [Raft v3.7.0](#)
- [Dependencies](#)
- [Release cadence](#)
- [Contributors](#)

Introduction

Today, SIG etcd is releasing [etcd v3.7.0](#), the latest minor release of the popular distributed key-value store and core Kubernetes component. v3.7 ships the long-requested RangeStream feature, delivers several other performance improvements, removes the last remnants of the legacy v2store, and completes a major protobuf overhaul.

You can download etcd v3.7.0 here:

- [Source code](#)
- [Binaries](#)
- [Official container images](#)

This release also includes new versions of the two core etcd dependencies, [bbolt v1.5.0](#) and [raft v3.7.0](#).

For instructions on installing etcd, see the [install documentation](#). For the full list of changes, see the [etcd v3.7 changelog](#).

A heartfelt thank you to all the contributors who made this release possible!

Major features

The headline changes in v3.7.0:

- [RangeStream](#) — stream large result sets in chunks instead of buffering the whole response.
- ~~[Keys-only range optimization](#) — serve `keys-only` ranges from the in-memory index, skipping backend reads.~~
- **Keys-only range requests, faster and more reliable leases**, and a range of other [performance improvements](#).
- etcd now boots entirely from v3store, eliminating a long-standing dependency on the legacy v2 store
- A completed [protobuf overhaul](#), replacing outdated protobuf libraries with fully supported ones.

- etcd v3.7 ships with bbolt v1.5.0 and raft v3.7.0.

Features

Performance improvements

v3.7 delivers multiple performance improvements, both for the Kubernetes control plane and for other use cases. In Kubernetes scalability tests, etcd consumes up to 50% less CPU time for operation.

RangeStream

In etcd v3.6 and earlier, it is challenging to work with requests that return large result sets. The database would buffer the full result set before sending, leading to unpredictable latency and memory usage, both on the server and the client. [The RangeStream RPC](#) lets calling applications accept result sets in chunks, reducing latency and making buffering memory usage more predictable.

Instructions on how to use RangeStream [in gRPC calls](#) and [in etcdctl](#) can be found in the etcd documentation. Users should try it out for their own applications.

As a coordinated release, the RangeStream feature will become available to users running v1.37 of Kubernetes by enabling the `EtcdRangeStream` feature gate. This early and planned adoption is possible thanks to the merger of etcd and Kubernetes development in 2023.

Keys-only range optimization

etcd v3.7.0 includes a keys-only Range optimization ([#21791: keys-only Range optimization](#)). When processing a keys_only Range request or `etcdctl get --keys-only`, etcd reads solely from its in-memory index. It return keys without loading all serialized values from bbolt as it did previously. The only exception where loading from bbolt is still required is when `keys_only` Range requests must be sorted by value (i.e., when `SortTarget` is set to `VALUE`).

This reduces unnecessary backend reads and memory use for workloads that only need key names, making large keys-only range requests more efficient.

Faster, more reliable etcd leases

v3.7 improves lease expiration and renewal:

- LeaseRevoke requests are now prioritized to ensure timely lease expiration during overload ([#20492: stability enhancement during overload conditions](#)).
- The new `FastLeaseKeepAlive` feature enables faster lease renewal by skipping the wait for the applied index ([#20589: etcdserver: improve linearizable renew lease](#)).

Faster find() operations

etcd 3.7 improves the performance of concurrent watches on keys by making find() operations faster.

([#19768: adt: split interval tree by right endpoint on matched left endpoints](#)).

Other features

Unix socket support

etcd now supports Unix socket endpoints ([#19760: Add Support for Unix Socket endpoints](#)), enabling local communication without a TCP port. Since this is restricted to single-member clusters, it is mainly aimed at development, testing, and edge device use-cases.

Bootstrap from v3store

One of the major changes in etcd v3.7 is that the server now bootstraps entirely from the v3 store, eliminating its dependency on the legacy v2 store during startup.

This milestone is the result of a long-term effort spanning multiple releases, from v3.4 through v3.7. It resolves a long-standing technical debt, significantly simplifies the bootstrap workflow, and lays the foundation for future improvements to etcd.

To maintain backward compatibility, etcd v3.7 continues to generate v2 snapshots. As a result, the `--snapshot-count` flag is also retained in v3.7. This is the last remaining dependency on the legacy v2 store, and both the v2 snapshot generation and the `--snapshot-count` flag are planned to be removed in v3.8.

For more background and implementation details, see GitHub issue #20187: <https://github.com/etcd-io/etcd/issues/20187>.

etcdctl timeouts

All `etcdctl` commands now have a timeout command line argument ([#20708: etcdctl: enable timeout functionality for all commands](#)), so offline utility commands no longer block indefinitely when holding a lock.

Setting the authentication token directly

`clientv3` now allows users to set the JWT directly ([#16803: clientv3: allow setting JWT directly](#), [#20747: clientv3: disable auth retry when token is set](#)),

Retrieve AuthStatus without authenticating

Clients can check their `AuthStatus` without attempting to authenticate first, eliminating some application overhead ([#20802: etcdserver: remove permission check on AuthStatus api](#)).

New watch metrics

v3.7 adds optional watch send-loop metrics ([#21030: Instrument watchstream send loop](#)) for better observability of the watch path:

- `etcd_debugging_server_watch_send_loop_watch_stream_duration_seconds`
- `etcd_debugging_server_watch_send_loop_watch_stream_duration_per_event_seconds`
- `etcd_debugging_server_watch_send_loop_control_stream_duration_seconds`
- `etcd_debugging_server_watch_send_loop_progress_duration_seconds`

There is also a new `etcd_server_request_duration_seconds` metric ([#21038: Add metric etcd_server_request_duration_seconds](#)).

etcdctl command cleanup

`etcdctl` commands were reorganized for clarity ([#20162: etcdctl: organize etcdctl subcommand](#)) and global command line arguments are now hidden to streamline help output ([#20493: etcdctl: hide global flags](#)).

Upgrading

This release contains breaking changes, particularly around the removal of legacy v2 components. Users should review the [upgrade guide](#) before upgrading their nodes. As with all minor releases, perform a rolling upgrade one member at a time and confirm cluster health between steps.

Experimental flags removed

All deprecated experimental flags have been removed ([#19959: Cleanup the deprecated experimental flags](#)). Features in etcd now follow the Kubernetes-style feature-gate lifecycle (Alpha → Beta → GA) introduced in v3.6, rather than the old `--experimental` prefix. If your configuration still relies on `--experimental-*` command line arguments, migrate to using the corresponding feature gates or stable command line arguments before you upgrade to etcd 3.7.

Legacy V2 API packages and code cleanup

To remove the dependencies on v2store, the following components have been removed:

- [v2 discovery](#) ([#20109: Remove v2discovery](#)) packages removed,
- [v2 request](#) support ([#21263: Remove v2 Request and apply v2.go](#))
- [v2 client](#) support ([#20117: Remove client/internal/v2](#)).

These changes may create some breakage for users, particularly those who have not already updated to v3.6.11 or later. Users should report any blockers encountered, or cases that need better upgrade documentation.

Non-blocking client creation

etcd no longer honors the deprecated `grpc.WithBlock` dial option ([#21942: Make the etcd client creation non-blocking](#)). To preserve the previous blocking behavior when needed, follow the guidance in grpc-go's [anti-patterns documentation](#).

Protobuf overhaul

v3.7 migrates and replaces multiple outdated protobuf libraries with fully supported dependencies. This includes replacing `github.com/golang/protobuf` and `github.com/gogo/protobuf` with the fully-supported `google.golang.org/protobuf` ([#14533: Protobuf: cleanup both golang/protobuf and gogo/protobuf](#)), and migrating `grpc-logging` to `grpc-middleware v2` ([#20420: Migrate grpc-logging to grpc-middleware v2](#)).

While these changes are not expected to directly affect users running etcd via official binaries or container images, they may affect users who depend on etcd Go modules, such as the client SDK or packages under `api/` or `pkg/`. These consumers may need to update their code or dependencies due to protobuf and related API changes introduced in this release. More detailed information is available from [the API change tracking issue](#).

Multiarch container images only

For users relying on the official etcd container images, v3.7 will be distributed **only** as multiarch containers. Architecture-tagged images will not be available, so adjust deployments accordingly.

API changes

As with every etcd release, there are a number of API changes. These are designed to be backwards-compatible to the extent possible, but may require adjustment by some users. See our [API Changes In v3.7] page for full information.

bbolt v1.5.0

etcd v3.7 depends on, and includes, v1.5.0 of the bbolt storage engine. v1.5 includes several improvements to functionality and performance, including:

Database file size limits: users may set, and bbolt will enforce, file size limits. When a bolt database exceeds these limits it will refuse to accept writes until the database is compacted or the limit is changed (<https://github.com/etcd-io/bbolt/pull/929>, <https://github.com/etcd-io/bbolt/pull/1210>)

Disable statistics for performance: users may set `NoStatistics` to limit overhead from locks taken by the database statistics viewer (<https://github.com/etcd-io/bbolt/pull/977>)

More efficient hashmap processing: merge spans faster and with less overhead (<https://github.com/etcd-io/bbolt/pull/1179>)

raft v3.7.0

etcd 3.7 depends on, and includes, v3.7.0 of the raft consensus engine. v3.7 includes several improvements, including:

Update the bootstrap process: v3.7 now allows booting from partly initialized snapshots, supporting etcd's initializing directly from v3store (<https://github.com/etcd-io/raft/pull/370>)

Improve the ReadIndex flow to prevent stale reads by injecting a unique identifier into the heartbeat context for read-only operations. See <https://github.com/etcd-io/raft/pull/397> for details.

raft v3.7.0 also includes the same protobuf library updates and refactoring as etcd does (<https://github.com/etcd-io/etcd/issues/14533>)

Dependency Updates

Other dependency updates include a bump to `golang.org/x/crypto` v0.52.0 for CVE resolution ([#21903: \[release-3.7\] Bump golang.org/x/crypto to v0.52.0](#)), an OpenTelemetry contrib update to v0.61.0 ([#20017: Update otelgrpc to v0.61.0](#)), and compilation with Go 1.26.4 ([#21891: \[release-3.7\] Update Go to 1.26.4](#)).

Release cadence

Starting with the v3.7.0 release cycle, etcd has adopted a more predictable release cadence, targeting one major or minor release per year. This provides users and downstream projects with clearer expectations for planning upgrades while ensuring regular delivery of bug fixes, security updates, dependency upgrades, and operational improvements, even when there are few user-facing features.

A code freeze is planned at least one month before each final release to provide sufficient time for stabilization and release validation. For more details, please see the release documentation in [release.md](#).

Contributors

etcd v3.7.0 is the product of more than 100 contributors across the community. Thank you to everyone who wrote code, reviewed PRs, filed and triaged issues, and helped test the alpha, beta, and release candidates.

Leads

The SIG etcd leads for the v3.7 release are: [serathius](#), [ahrtr](#), [fuweid](#), [siyuanfoundation](#), and [jberkus](#). Release Lead [ivanvc](#) leads our release team.

Other contributors

[ah8ad3](#), [ajaysundark](#), [aladesawe](#), [amosehigueuse](#), [ArkaSaha30](#), [ashikjm](#), [AwesomePatrol](#), [dims](#), [gangli113](#), [henrybear327](#), [hwdef](#), [Jille](#), [joshuazh-x](#), [liggitt](#), [miancheng7](#), [mmorel-35](#), [mrueg](#), [purpleidea](#), [qsyqian](#), [redwrasse](#), [rsafonseca](#), [skitt](#), [tcchawla](#), [tjungblu](#), [vivekpatani](#), [wenjiaswe](#)

New contributors

A special welcome to the contributors who made their first etcd contribution in this cycle – including [Jeffrey Ying](#), whose work drove the RangeStream feature. New contributors can have a substantial impact on etcd; if you'd like to get involved, see the [contributor guide](#).

[4rivappa](#), [abdurrehman107](#), [akstron](#), [aliasgher](#), [aojea](#), [apullo777](#), [AR21SM](#), [asttool](#), [BBQing](#), [beforetech](#), [boqishan](#), [caltechustc](#), [carsontham](#), [chuanye-gao](#), [cuiweixie](#), [Deln0r](#), [dmvolod](#), [Dogacel](#), [dongjiang1989](#), [EduardoVega](#), [evertrain](#), [eyupcanakman](#), [gaganhr94](#), [goingforstudying-ctrl](#), [Himanshu-370](#), [HossamSaberX](#), [huajianxiaowanzi](#), [jihogh](#), [jonathan-albrecht-ibm](#), [joshjms](#), [kairosoci](#), [kishen-v](#), [kjgorman](#), [kstrifonoff](#), [letreturn](#), [lorenz](#), [madhav-murali](#), [majiayu000](#), [marcus-hodgson-antithesis](#), [mcrute](#), [mingl1](#), [MohanadKh03](#), [NAM-MAN](#), [nihalmaddala](#), [notandruu](#), [nwnt](#), [pjsharath28](#), [ravisastrik](#), [robin-vidal](#), [rockswe](#), [ronaldngounou](#), [sahilpatel09](#), [SalehBorhani](#), [SebTardif](#), [seshachalam-yv](#), [shashwat010](#), [shivamgcodes](#), [shuan1026](#), [silentred](#), [sneaky-potato](#), [srri](#), [subrajeet-maharana](#), [tchap](#), [tsujiri](#), [upamanyus](#), [varunu28](#), [xiaoxiangirl](#), [xigang](#), [xUser5000](#), [yagikota](#), [yajianggroup](#), [yedou37](#), [Zanda256](#), [zechariahkasina](#), [zhijun42](#), [zhoujiawei](#)

Feedback can be shared through:

- [GitHub issues](#)
- [#sig-etcd slack channel](#) in [Kubernetes Slack](#)
- [etcd-dev mailing list](#)