Выпущен Prometheus 3.15.0

Выпущен Prometheus 3.15.0

Prometheus 3.15.0 — крупное обновление с новыми возможностями для конфигурации, scraping, TSDB, PromQL и service discovery. В релизе также заметно доработали производительность и исправили ряд ошибок в PromQL, хранилище, remote write, native histograms и интерфейсе.

Новые возможности

- Уровень логирования теперь можно менять через параметр `runtime.log_level` при перезагрузке конфигурации. Старый флаг `--log.level` объявлен устаревшим.
- Добавлен флаг `--auto-gomemlimit.refresh-interval`: Prometheus может периодически заново определять лимит памяти контейнера или системы и обновлять `GOMEMLIMIT` во время работы.
- Scraping теперь поддерживает цели, доступные через Unix Domain Sockets.
- Реализована поддержка формата scraping OM 2.0.

Улучшения

- При инициализации TSDB head снижена нагрузка на CPU.
- В Docker service discovery добавлены метки `__meta_docker_container_image` и `__meta_docker_container_image_id`.
- В mixin добавлена панель p95/p99 для задержки отправки пакетов remote write. Эта панель также поддерживает native histograms.
- Флаг `--enable-feature=st-storage` теперь автоматически включает кодирование float-чанков XOR2 и ST-совместимое кодирование histogram-чанков. Отдельно указывать `xor2-encoding` и `histograms-st-encoding` больше не нужно.
- Для remote write и Alertmanager обновлена поддержка `sigv4`: появились поля `session_name` и `tags` для STS AssumeRole, а поле `service_name` теперь задокументировано.
- Scraping научился принимать ответы, сжатые zstd. Возможность включается feature flag `zstd-scrape`.
- Кодирование float-чанков XOR2 в TSDB стабилизировано. Флаг `--enable-feature=xor2-encoding` устарел, вместо него нужно использовать `storage.tsdb.chunk_encoding.floats: xor2`. Перед включением стоит проверить, что сторонние инструменты, читающие TSDB напрямую, например Thanos sidecar, поддерживают XOR2.
- Добавлены метрики `prometheus_tsdb_head_appenders_created_total` и `prometheus_tsdb_head_series_pending_commit_underflow_total`.
- В tracing добавлены дополнительные spans для scraping, API-запросов и вычисления правил.

- На страницах Targets и Service Discovery теперь показывается фактически применяемая конфигурация для каждого scrape pool.
- Для series `summary_count` и `summary_sum` включён синтез времени старта в scrape appender v2.
- При завершении работы scrape pools останавливаются параллельно, поэтому shutdown проходит быстрее.
- В remote storage добавлено диагностическое поле `failed_request_logging`, которое помогает логировать запросы remote write v2 при ошибках отправки.
- Для XOR-чанков в TSDB добавлен быстрый путь декомпрессии, ускоряющий запросы.
- В интерфейсе улучшено форматирование таблиц native histograms и добавлена фоновая полоса, показывающая количество buckets.

Производительность

- В AWS service discovery метки RDS-кластеров теперь строятся один раз на кластер, а не для каждого инстанса.
- Описание RDS-инстансов из разных кластеров выполняется параллельно с учётом лимита `request_concurrency`.
- Каждый ресурс ElastiCache теперь запрашивается один раз за обновление, а не дважды.
- В remote read больше не клонируются labels для sampled reads, если не настроены external labels.
- В remote write повторно используется рабочее состояние OTLP-конвертера между запросами.
- При конвертации classic histograms в native histograms время старта разбирается только тогда, когда эта возможность включена.

Исправления

PromQL

- Range query, у которого конец не был выровнен по step, больше не заставляет вложенные subqueries вычисляться за пределами последнего реального шага родительского запроса. Это снижает лишнее потребление `peakSamples` и уменьшает ненужное чтение из хранилища.
- Сброс start timestamp больше не регистрируется, если start timestamp не изменился между соседними samples.
- Исправлено обогащение `info()` для составных выражений со смешанными ссылками `@` и `offset`, а также для vector-веток без selector.
- `info()` корректно работает, когда входные series используют разные подмножества идентифицирующих labels.
- При включённом отложенном удалении имён метрик `info()` сохраняет ожидаемое удаление metric name.
- `info()` теперь применяет модификатор `@` и `offset` при вычислении info series, поэтому `info(v @ T)` обогащает данные относительно времени `T`.

- Исправлено ложное совпадение `FastRegexMatcher`, когда capturing group стоит вплотную к литералу.
- Устранена паника в range selectors с экспериментальными модификаторами `anchored` или `smoothed`, если выбранная series не содержит samples внутри окна запроса.
- Исправлены пустые результаты для subquery с `@`, когда она используется как matrix-аргумент функции, которая не является step-invariant.
- Сообщение об ошибке many-to-many matching теперь детерминировано: две конфликтующие labels сортируются перед выводом.
- При round-trip через `Expr.String()` сохраняются скобки вокруг duration literals.
- Duration-expression offsets и `@ start()` / `@ end()` перед range selectors теперь отклоняются так же, как уже отклонялись literal offsets и `@ <timestamp>`.
- В annotations для `histogram_quantile` и `histogram_fraction` теперь указывается позиция histogram-аргумента, а не scalar-аргумента.

TSDB и хранение данных

- TSDB больше не теряет samples, если garbage collection удаляет series во время append.
- Series с неподтверждёнными samples сохраняются при compaction выбранных и stale series, включая случаи с пересекающимися appenders.
- После отклонения синтетического zero sample для start timestamp head series больше не удерживается лишний раз.
- Исправлена паника запросов при удалении series после WAL replay.
- Устранён потенциальный deadlock между `mmapSeriesChunks` и `gcSeries`.
- Исправлен интервал перезагрузки blocks по умолчанию для пользовательских options.
- Исправлено переполнение ID in-order и out-of-order chunks: значения теперь корректно оборачиваются и не конфликтуют с битом out-of-order.
- Логи WAL и GC теперь выводят длительности в человекочитаемом виде, а не как целые значения в наносекундах.
- В Agent mode исправлена возможная порча WAL после неудачной записи.
- Agent теперь игнорирует неизвестные типы WAL records, что упрощает откат на предыдущую версию.

Service Discovery

- AWS service discovery больше не падает на serverless MSK clusters, MSK clusters без Open Monitoring, standalone ECS tasks с пользовательскими task groups, ElastiCache ARN без resource ID и ECS tasks без необязательных полей.
- Для AWS Lightsail исправлена паника, если у instance отсутствуют необязательные поля вроде availability zone, blueprint, bundle, name, state или support code.
- Для ElastiCache исправлена паника при отсутствии необязательных полей в ответах API.
- Для EC2 исправлена паника, если API не возвращает необязательные поля instance.

- Некорректное значение `request_concurrency` теперь отклоняется сразу, вместо того чтобы бесконечно подвешивать service discovery.
- При удалении конфигурации во время reload теперь удаляется устаревшая series `prometheus_sd_last_update_timestamp_seconds`.
- В IONOS service discovery исправлены падения при отсутствии server, NIC, volume collections или отдельных свойств server в ответе API.
- В Kubernetes service discovery метка `__meta_kubernetes_service_loadbalancer_ip` заполняется из `status.loadBalancer.ingress`, с fallback на устаревшее поле `spec.loadBalancerIP`.

Scraping, remote write и OTLP

- Metadata больше не влияет на количество shards в Remote Write v2.
- OTLP ingestion больше не отклоняет весь payload, если у одной метрики нет datapoints.
- Scraping больше не добавляет stale marker для series, которая всё ещё отдаётся target, если storage возвращает для неё новую ссылку.
- JSON-форматтер логов теперь корректно форматирует информацию о scrape target.
- Исправлена data race в `Manager.TargetsDroppedCounts`, из-за которой могло неверно считаться количество dropped targets.
- Исправлен nil histogram при смешивании native и classic histograms в одном metric family.

Native histograms и promtool

- `DetectReset` больше не пропускает reset счётчика, когда заполненный bucket за пустым bucket исчезает. Раньше из-за этого `rate()` и `increase()` могли занижать результат.
- `histogram_stddev` и `histogram_stdvar` больше не пропускают buckets.
- Исправлена ошибка `Compact`, из-за которой buckets могли перемещаться на неверные индексы, а integer histograms получали отрицательные значения bucket counts.
- `promtool tsdb dump` больше не теряет native histogram samples.

UI, rules и mixins

- В native histogram chart убрана лишняя отметка на оси X при режиме отображения `linear`.
- Исправлена паника rule manager, созданного без logger, при загрузке rule file с несколькими YAML-документами.
- В mixins исправлены несовпадения labels, из-за которых не срабатывали alerts `PrometheusHAGroupNotIngestingSamples` и `PrometheusHAGroupCrashlooping`.
- HTTP-ответы со сжатием больше не обрезаются, если handler выставил `Content-Length`.