

This is a public document

PUBLICLY SHARED

Grafana Mimir Community Call

- Our meetings are recorded and posted publicly on the Grafana YouTube channel. If you do not want to be recorded or have privacy concerns about other members of your household appearing on camera during the meeting, you are free to disable your mic/camera, join under a pseudonym, or watch the public recordings.
- We will meet on the **last Thursday of every month at 1430 UTC**
- Meeting URL is: <https://meet.google.com/xjz-noqi-dtb>
- Subscribe to our calendar for updates
 - [Web view](#)
 - [Import in your own calendar](#)
- Slack: <https://slack.grafana.com>, #mimir channel.
- Github: <https://github.com/grafana/mimir>
- Documentation: <https://grafana.com/docs/mimir>
- [YouTube playlist with recorded meetings](#)

2025-08-28

Leading: Nick Pillitteri

- Mimir 2.17 has been released
 - <https://github.com/grafana/mimir/releases/tag/mimir-2.17.0>
- Mimir 3.0 work [continues!](#)
 - [GitHub milestone](#)
 - In the Helm chart, default will be ingest storage using a demo Kafka instance (single replica, enough to get started with) with overrides available to run the classic architecture.
 - Question from the community:
 - Will we keep supporting classic architecture?
 - Yes, Mimir 3 will support it.
 - For cases needing more than Prometheus but not a full blown Kafka, what to do?
 - The Helm chart will include a simple Kafka instance for that.
 - Is MQE still experimental?
 - No but we need to update docs. It's the default in queries in 2.17 and default in query-frontends in 3.0

2025-07-31

- [External Sharing] Mimir Community Call - Mimir 3.0
- [External Sharing] Mimir 3 Community FAQs
- [Mimir 3.0 milestone](#) with planned (breaking) changes [and pinned 3.0 release issue](#).
-

2025-06-26

Leading: Dimitar Dimitrov

- Mimir 2.17
 - Distributor HA tracker on Memberlist being promoted to stable
 - with timeouts and write concurrency it's more snappy than before;
 - OTEL-only traces
- Timing of 3.0 release
 - no hard deadline, rough target is end of Q3 - Nov 1, 2025
 - Dimitar Dimitrov will look into publishing the 3.0 targeted changes in a github issue
- Crashed during fresh installs
 - querier, store-gateway, query-scheduler crashloop for a few times before settling
 - this is expected because caches haven't started up yet and their DNS doesn't resolve

2025-05-29

Leading: vladi varankin +1 community member

- Mimir 2.17 will be the last release before Mimir 3
 - If you have any changes you'd like to see in Mimir 3, please create issues with the "mimir-3-proposal" label.
 - Mimir 3 talk from GrafanaCon'25
 - No recording yet
 - The timeline for Mimir 3 isn't set yet, but the team aims for Q3'25
 - The current plan is to keep the "classic" architecture, but to promote the Kafka-based architecture to stable

2025-04-23

Leading: Oleg Zaytsev

- Mimir and GEM 2.16 are out along with version 5.7 of the helm chart

- Nick Pillitteri It has been mentioned in 2.14 release notes but Redis support in Mimir is deprecated and will be removed in 3.0 (we don't have a date for 3.0 yet). Support in Jsonnet and Helm has already been removed or never existed (respectively).
- We're planning a 3.0, if you have any changes you'd like to see there, please create issues with the "mimir-3-proposal" label.
- Denys Zhdanov: What's the motivation for 3.0? Is there anything big or radically different coming or is it like Prometheus 3.0
 - Probably no unexpected changes
 - Ingest storage will be stable, all the docs to migrate, etc.
 - Possibly cleaning up esoteric ways to configure the read path (stand-alone queriers?)
- Denys Zhdanov: DeckHouse Prom++ project
 - Some Russian company took Prometheus and they replaced TSDB with C++ Storage
 - Oleg: Some Grafana, Cloudflare, AWS and Shopify engineers are collaborating on Apache Parquet experiment as a replacement for tsdb.

2025-03-27

Leading: Nick Pillitteri +5 community members

- Mimir 2.16 RC0 available now. Final release early next week
 - Alpine images no longer published (deprecated since 2.12). Instead, distroless images only.
 - Streaming query engine still experimental but adding more PromQL compatibility and bug fixes.
 - Support for native and OOO histograms is enabled by default.

2025-02-27

Leading: Dimitar Dimitrov vladimir varankin Alex Weaver Julien Duchesne Kishor U. +5 community members

- Mimir 2.16 scheduled for beginning of March
- Optimizing query path
 - deploying caches should improve
 - timeouts between query-frontend and query-scheduler
 - github.com/grafana/mimir/issues
- [The next generation of Grafana Mimir: Inside Mimir's redesigned architecture for increased reliability](#)

- Disaster recovery testing. Q: Will the new architecture change the order of ingestion? This might be helpful when catching up from a long outage period. A: order of ingestion isn't changed, but you can skip records
- Anjul: Migrating a client from InfluxDB to Mimir
 - <https://grafana.com/docs/mimir/latest/>
 - Mimir 2.16 will include native influx ingestion
<https://github.com/grafana/mimir/pull/10153>

2025-01-30

Leading: vladimir varankin Armand Grillet Martin Valiente +2 community members

- [Mimir 2.15](#) released
 - Main features were already mentioned last month during the rc-phase.
- Lukas: Alerting experience. Work on consolidating alertmanagers (grafana and alertmanager)
 - Teams look at APIs in Grafana, while Mimir uses a blob of text
 - Armand: We would need to check. There was a work scheduled but we don't have the timeline.
- Lukas: Alert pagination. 1K+ alerting rules.
 - Prometheus feature is implemented. Merged into upstream in November.
 - Armand: seems like it's in progress in Mimir. Will take some time.
- Jhin: Ruler. Moving to remote-ruler mode.
 - Make sense for lots of rules
 - Making dedicated pools of queries for rulers

2024-12-26

Leading: Oleg Zaytsev

No recording because the only community participant was communicating through the chat, so we just took the notes.

- [Mimir 2.15 release in progress, rc.0 is out.](#)
 - The most important change is that Mimir is now PromQL 3.0 compatible
 - This affects regular expression matchers
 - And range selectors are now left-open
 - Mimir's experimental PromQL engine can now be enabled with `-querier.query-engine=mimir`
 - We don't recommend running this in production yet.
- Q: For maintaining alerts as lac, do you recommend using alertmanager CRD vs terraform

- We use <https://github.com/grafana/mimir/blob/main/operations/mimir-rules-action/README.md> internally
- Not aware of terraform, maybe it's a thing?
- Nainar uses BitBucket:
 - Then use mimirtool, you can read how the action is implemented here: <https://github.com/grafana/mimir/blob/main/operations/mimir-rules-action/entrypoint.sh>
 -

2024-11-28

- No community call (Thanksgiving break).

2024-10-29

Leading: Oleg Zaytsev

Attendees: Nikolaos Angelopoulos Martin Valiente + 5 community members

- [Mimir 2.14 released](#)
 - Main features were already mentioned last month prior to the release.
 - We've introduced a new way to switch ingesters to the read-only mode for proper downscaling
- Prometheus 3.0 changes to mention for future Mimir releases:
 - Left open range selectors
 - Regexp matchers matching newlines by default
- Denys Zhdanov: Q: "Ingest storage" isn't documented, so people can't try and provide feedback.
 - Using read/write mode implemented in Helm
 - Oleg Zaytsev probably going away
 - If you want to experiment, try jsonnet
- Quentin Bisson:
 - What are we running on Cloud, kafka?
 - Sort by label function availability
- Peter Yan
 - Running monolithic on VMs, OOO samples crash mimir and they have to reboot the server, they're using 2.14
 - 17M series per instance
 - Check if it's happening at tsdb compaction time
 - Maybe mutex contention fixed ongoing: <https://github.com/prometheus/prometheus/pull/15242>
 - Envoy proxy to mirror reads and writes

- Edwin Tye
 - MQE, is Grafana running it in prod
 - Only in shadow mode.

2024-09-26

Leading: Nick Pillitteri

Attendees:

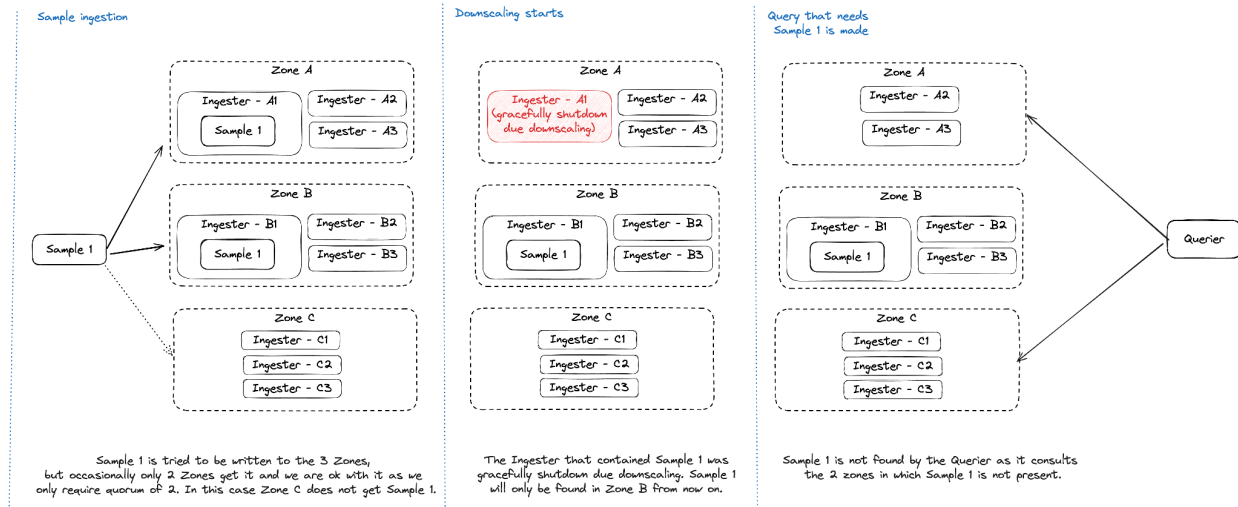
- Mimir 2.14 (not out or RC yet) highlights
 - Query-scheduler improvements to request queuing which prevent a state where all queriers become saturated with requests for a degraded downstream component (e.g., ingesters or store-gateways).
 - Requests in store-gateways won't wait indefinitely for an index-header to lazily load, instead the request will fail after 5 seconds by default. This should help avoid newly started store-gateways from causing the rest of the read path to slow down by allowing the request to retry with a different store-gateway.
 - Second version of ingester downscaling with read-only ingesters (only in Mimir's jsonnet library at the moment)
 - Experimental Mimir Query Engine (MQE) is now available in queriers. We're starting to test this in production and the results so far are very promising (reduced and more consistent CPU and memory usage). There might be issues with it and we'd appreciate people giving it a shot in non-production usage.
- Prometheus 3.0 changes to mention for future Mimir releases:
 - Left open range selectors
 - Regexp matchers matching newlines by default

2024-08-29

Leading: Peter Štibrný

Attendees: Peter Štibrný , Armand Grillet , Andrea Gardiman, an AI 🤖

- Mimir 2.13 release is out.
 - <https://github.com/grafana/mimir/releases/tag/mimir-2.13.0>
- New ingester downscaling
 - The issue:



- Ingest storage updates
 - New architecture of Mimir being based on Kafka
 - Work on faster replays from Kafka
 - Block-builder work
 - Move of strong-consistency to query-frontend.
-

2024-06-27

Leading: Peter Štibraný

Attendees: krajo Krajcsovits , Oleg Zaytsev , Armand Grillet , Roman Hüsler.

- [Mimir 2.13 rc.0](#) highlights
 - Improved CPU performance in processing queries with regular expressions that contain many alternations
 - (for example `foo|bar|baz|...`).
 - Improved OTLP ingestion performance by using native translation, which reduces memory usage in the distributor by up to 30% and CPU usage by up to 8%.
 - Mimirtool can now verify the validity of a Mimir runtime configuration file with the `mimirtool runtime-config verify` command.
 - Active series are now updated along with owned series. This means the number of active series for a tenant is more accurate after scaling out ingesters. As a result, ingesters now more precisely enforce tenants' series limits.
 - Remote read (`/prometheus/api/v1/read`) is becoming a first-class endpoint in Mimir with support in query stats logs and query blocking in the query-frontend.
 - More coming in future releases.
 - Fixed lock contention between writes, reads and TSDB Head GC:
 - Improves latency and fixed apparently stuck head compaction under heavy read

- Store-gateway can be configured to explicitly disable or enable tenants.
 - This is useful in cases where compaction slows down for a tenant to temporarily exclude another tenant while compaction catches up without affecting other tenants.
- Configurable S3 bucket lookup type improves interoperability with S3-compatible providers, such as Tencent and Alibaba. Configure lookup type via `-<prefix>.s3.bucket-lookup-type` or `-common.storage.s3-bucket-lookup-type`.
- Community
 - 3h power outage, when power came back, saw total outage
 - OOO rejected samples
 - Queries didn't go through
 - Apparently nginx was overwhelmed
 -

2024-05-30

Leading: Arve Knudsen

Attendees: Oleg Zaytsev

- Helm
 - Dedicated ruler query path
 - enable with `ruler.remoteEvaluationDedicatedQueryPath: true`
 - deploys dedicated queriers, query-frontends, query-scheduler
 - supports horizontal autoscaling too
 - Credit to [@alex5517](#) for carrying the PR through the finish line and to [@gonzalez](#) for doing the initial effort.
- OTLP
 - OTLP ingestion has been [rewritten](#) in the distributor to convert directly to Mimir write request format; benchmarks show an average speedup of about 5% and memory usage reduction of about 28%.
- Community operational questions:
 - What labels are expected by the mixin?
 - Seeing some unhealthy ingesters in the ring, with a 9 ingesters deployment.

2024-04-25 ([recording](#))

- Contributions
 - What to contribute to?
 - Four main areas of issues:
 - go lang code ([good first issue](#) is usually a good start)
 - [mixin](#) (dashboards & alerts)

- deployment tooling ([helm](#), [jsonnet](#), [deb](#) & [rpm](#) packaging)
- **More importantly: prefer contributing to something that makes a difference to you and how you use Mimir.**
- How to contribute?
 - [See /docs/internal/contributing](#)
 - More generally - feel free to open PRs with proposals at any point. If you're not sure if the work fits in Mimir, open an issue with the problem and your proposal for discussion.
- Helm 6.0
 - Considering switching to use OCI repository as first choice (keeping traditional repo for now)
- Native histograms migration slowly progressing
 - What will this mean for users?
- Kafka ingestion
 - Working on a project to introduce kafka between the distributors and ingesters
 - In that architecture ingesters are solely a read path component
 - Primary goals
 - Separate the read and write paths. An ingester outage doesn't stop ingestion. An outage in kafka doesn't result in a complete outage for queries - it only results in outdated query results
 - Make multi-AZ deployments cheaper by reducing cross-AZ data transfer.
 - Reduce default replication factor to 2 and weaken quorum constraints. Using kafka to store incoming data means we no longer need to replicate it to 3 ingesters. When querying only 1 ingester for each kafka partition is sufficient. This *might* help with reducing cost.
 - Still very experimental and a work in progress. Most of the code is in pkg/storage/ingest and in dskit.
 - There have been plenty of merged PRs if you search for Kafka in github but configuration parameters have been hidden.
- GOMAXPROCS
 - When running in containerized environments the automatic GOMAXPROCS is the number of cores on the host machine. On large machines a high value of GOMAXPROCS can lead to suboptimal performance of the go runtime.
 - At the same time it makes sense to leave some headroom to GOMAXPROCS so that processes can spike in CPU usage when they need to. For components autoscaled on CPU (distributor, querier), spikes would help trigger a scale-out. For ingesters they would allow to keep the read & write paths more stable during query surges.

2024-03-28

- Native histograms fixes:

- <https://github.com/grafana/mimir/pull/7552> (2.12)
- <https://github.com/grafana/mimir/pull/7640> (not in 2.12)
- 2.12 release:
 - Distroless images available
 - Sparse index-headers persisted by store-gateway (much faster lazy loading)
 - Ingester error gRPC handling changes, you need to migrate to 2.11 before updating to 2.12
 - Assorted performance improvements
- Q/A section
 - Thibaud, Giant Swarm:
 - Seeing huge amount of CPU usage in store-gateways – check compactor to make sure it's not falling behind
 - Equivalents to Loki stream sharding? Shuffle sharding on the write path. Store-gateway and query sharding on the read path.
 - Interesting metrics for resource usage? We mostly use the mixin dashboards
 - Tips on caching? Chunks and index-queries are the important ones. Minimize eviction for chunks, scale by CPU usage for index-queries.
 - How can we help? Who to ping about PRs?

2024-02-29 ([recording](#))

- Helm chart got experimental autoscaling as of 5.3.0-weekly.278 and will be released in the stable 5.3.0
 - Thanks to [Jan O. Rundshagen](#) (@beatkind) for contributing it!
 - TL;DR: check out [distributor.kedaAutoscaling](#), [ruler.kedaAutoscaling](#), [query_frontend.kedaAutoscaling](#), and [querier.kedaAutoscaling](#) in the default values.yaml
 - It still has a few rough edges around enabling it, the effort to promote it to a stable feature is tracked here <https://github.com/grafana/mimir/issues/7368>
- Native histograms updates coming in next release:
 - Query memory utilization optimizations inherited from Prometheus and also done by us (@duricanikolic on github).
 - New histogram_avg() PromQL function inherited from Prometheus.
 - We started to work on adding native histogram version of all latency metrics emitted by Mimir.
 - Dashboards, alerting and recording rules to follow, this will enable better measurements and reduce the number of series that Mimir emits.
- Stripe size in the Prometheus TSDB head when running with many thousands of tenants

- there should be diminishing returns when there are more than 4:1 ratio between the concurrent accessors and the number of stripes
- with many tenants the chance of concurrent writes on the same tenant are fairly small, so reducing the stripe size to, e.g. 1024 might help a lot
- Reducing `-blocks-storage.tsdh.head-chunks-write-buffer-size-bytes` from 4MiB to 1MiB also dropped heap with about 30% on a cluster with about 1500 tenants per ingester

2024-01-25

Attendees: Oleg Zaytsev Arve Knudsen Matt Mendick krajo Krajcsovits Dimitar Dimitrov

- [Mimir 2.11](#) has been released
 - Ingestor query request minimization
 - Workers for ingester grpc server and distributor push requests
 - Experimental, will be stable in next release
- Helm chart 5.2.0 was released but included a bug, the correct re-released version is [5.2.1](#).
 - No major changes in 5.2 but we're working on two items:
 - Document migration to enable cluster label verification (eventually it will be a breaking change to enable by default in 6.0)
 - Use Agent flow in meta monitoring (might be a breaking change so it would trigger 6.0)

2023-12-28

- Canceled due to low attendance. Ying Wang krajo Krajcsovits

2023-11-30 ([recording](#))

- What's new in upcoming release 2.11
 - Native histograms:
 - Basic documentation in place for how to instrument, send, ingest, query native histograms. Also a short intro to native histograms.
 - <https://grafana.com/docs/mimir/latest/configure/configure-native-histograms-ingestion/> (updated)
 - <https://grafana.com/docs/mimir/latest/send/native-histograms/>
 - <https://grafana.com/docs/mimir/latest/visualize/native-histograms/>
 - (the docs were backported to 2.10 as well)
 - Reduce resolution instead of rejecting samples with too many buckets.
 - Bugfixes in Prometheus, Mimir as well as in Grafana Agent (flow only) and Grafana (UI). Always use the latest versions.

- Features
 - The Retry-After header is utilized on the writing path for recoverable errors.
 - Grafana Agent version >0.37.3
 - Prometheus version >0.48.0 for Retry-After on 5xx, Retry-After on 429 has been supported for a while
 - Store-gateway sparse index headers are now enabled by default. Reduces loading a block in the store-gateway by up to 90%. Not so helpful when lazy loading is disabled.
 - Ingestor to querier chunks streaming is enabled by default. Should greatly reduce the memory usage of queriers.
 - Added the ability to [block queries by a regex](#). Useful for malicious or careless users.

2023-10-26

We skipped the meeting due to lack of external assistance and short agenda.

- Mimir 2.10.3 released !
 - Mimir 2.10.1 was released updating Go to 1.21.3 and containing several bugfixes.
 - Mimir 2.10.2 was released to address the [CVE-2023-44487](#), however the backport of the gRPC library contained a bug introducing a performance regression: the number of concurrent gRPC streams was eventually reduced to 1.
 - Mimir 2.10.3 fixes the aforementioned bug.
- Mimir 2.9.2 was released with the same security bugfixes, if you're still on that version.
- Features in progress
 - Native histograms: documentation under way. Latest Grafana Agent support.
 - Grafana (UI) improvements under way. Caveat.

2023-09-28 ([recording](#))

Lead: Oleg Zaytsev

Attendees: Marco Pracucci Arve Knudsen Nick Pillitteri Oleg Zaytsev Andrea Gardiman

- Mimir 2.10 released 🎉
 - Switching from 6-weeks to quarterly releases cadence
 - Highlights from [Release notes](#):
 - Reduced ingestor memory utilization (-15% at Grafana Labs)
 - Improve series and label names/values API performance

- Added support to cache cardinality, label names and label values query responses in query-frontend
- *More in the release notes...*
- Experimental features:
 - Support for **ingesting exponential histograms in OpenTelemetry** format
 - **Store-gateway index-header loading improvements**: persist optimized index-header (faster to load) and reload previously loaded index-headers after restart
 - Option to allow **queriers to reduce pressure on ingesters**
 - **Early TSDB Head compaction** in the ingesters to reduce in-memory series when a certain threshold is reached
 - If you want to try this out, you can start rolling this out in only one zone
 - Spread-minimizing token generation algorithm for the ingesters, to **reduce the difference in series pushed to different ingesters**
 - Support for **chunks streaming from store-gateways to queriers** (reduces queries memory utilization)
 - Support for **circuit-breaking the distributor write requests to the ingesters** (reduces pressure on ingesters when failing)
 - Still building operational experience with this at Grafana Labs
 - Support to limit read requests based on CPU/memory utilization. We **don't recommend** using the experimental read path limiting based on resource usage (if you're really interested, use weekly release as there are some bugfixes)
- New changes after 2.10 release (currently in "main" branch):
 - Reduce query-scheduler queue latency on high load
 - Ingestor: prevent query logic from continuing to execute after queries are canceled
 - Read requests rejected by the ingester due to utilization based read path limiting are not retried
 - The -log.buffered option is now enabled by default (the option is deprecated)
- Andrea: MimirContinuousTestFailed alert can be noisy and seems to complain about results that are wrong when they should be correct (like 990 of 1000 correct, 10 wrong)
-

2023-08-31 ([recording](#))

Attendees: Oleg Zaytsev Arve Knudsen krajo Krajcsovits Vernon Miller Sean Hanson

- Update on 2.10 release status.
 - Oleg Zaytsev shepherding this release
 - [Draft release notes](#)

- RC.0 released yesterday
- RC.1 to follow shortly promoting some features to stable and a bugfix.
- Target to publish final release on September 10th.
- We have switched to a quarterly release cycle! Due to bigger features and bugfixes.
- Status:
 - [Krajo] Out of order support for native histograms in Prometheus / Mimir.
 - [Krajo] Looking to introduce experimental auto-scaling support in the Helm chart for some components.
- We plan to replace the Grafana Agent Operator subchart in the mimir-distributed Helm chart with the new Grafana Agent chart which will reduce the need for loading CRDs.
- Documentation about how to use native histograms is upcoming in the next 3 months.

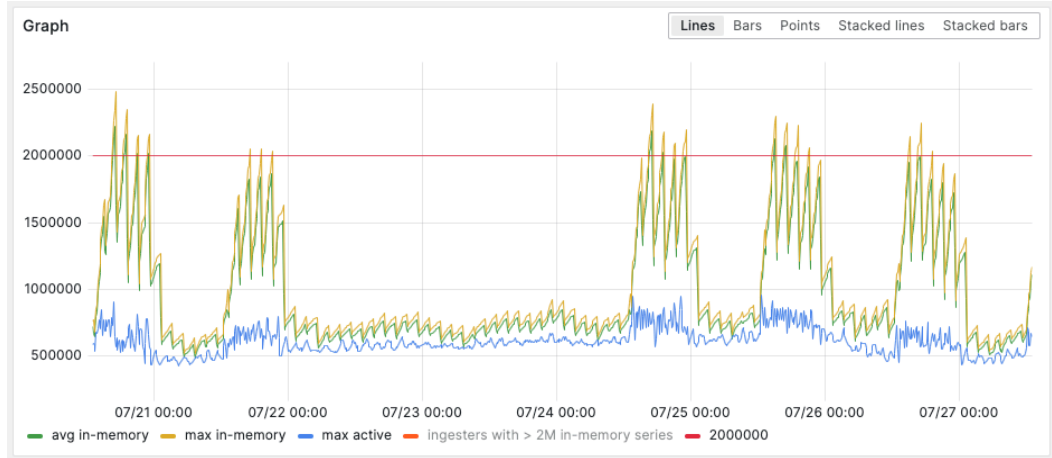
2023-07-27 ([recording](#))

Attendees: Oleg Zaytsev Marco Pracucci Jesús Vázquez Matt Mendick Peter Štibrný krajo Krajcsovits and 5 guests.

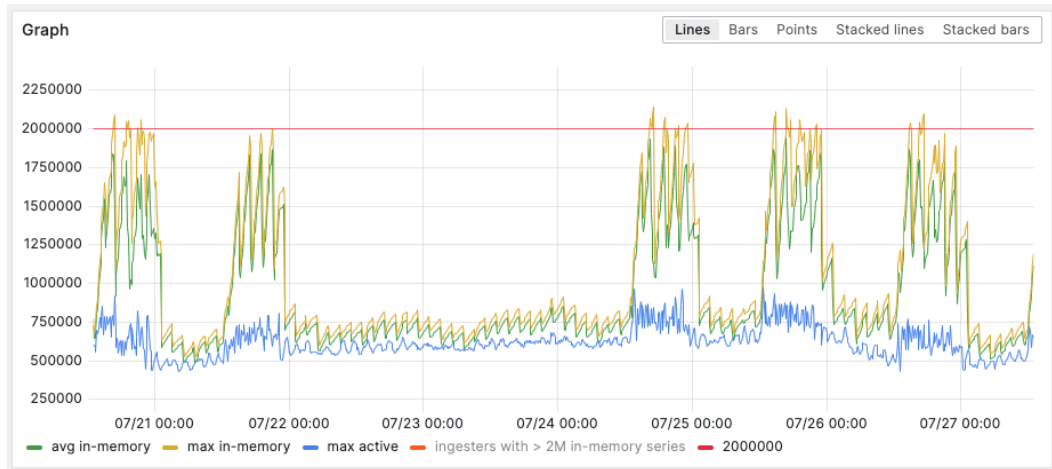
- The mimir-distributed Helm chart release 5.0
 - Due to PodSecurityPolicy vs Helm
 - Make sure you update your helm chart before updating your Kubernetes to 1.25.
- Release 2.10.0 is around the corner (Oleg Zaytsev will shepherd)
 - Prometheus ruler api compatibility: added rule [filtering](#).
 - Experimental support for caching cardinality and label names/values API query results.
 - Simple caching, CDN-like, just cache results for short period (like 1m) for same request.
 - Useful for people rushing onto the same dashboard, Grafana loads template vars through those endpoints.
 - Experimental ingester read (query) limiting based on CPU/memory utilization.
 - We're still tuning the CPU based limiting, seeing some corner cases. Memory-based limiting seems good.
 - We expect this to be delivered as stable in a couple of releases.
 - Experimental support for early TSDB head compaction: use case churn or scale out, shard size change.

Without early compaction:

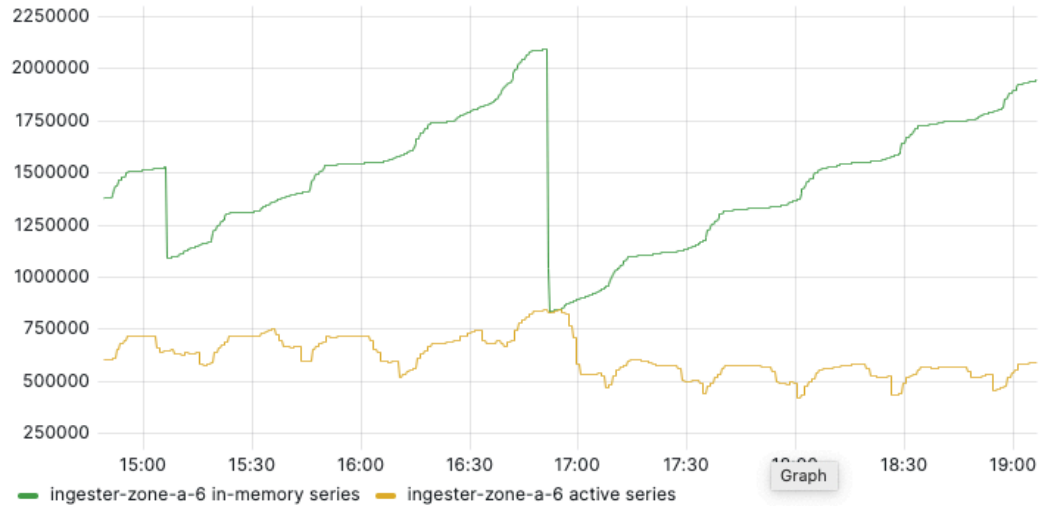
This is a public document



With early compaction (threshold at 2M in-memory series):

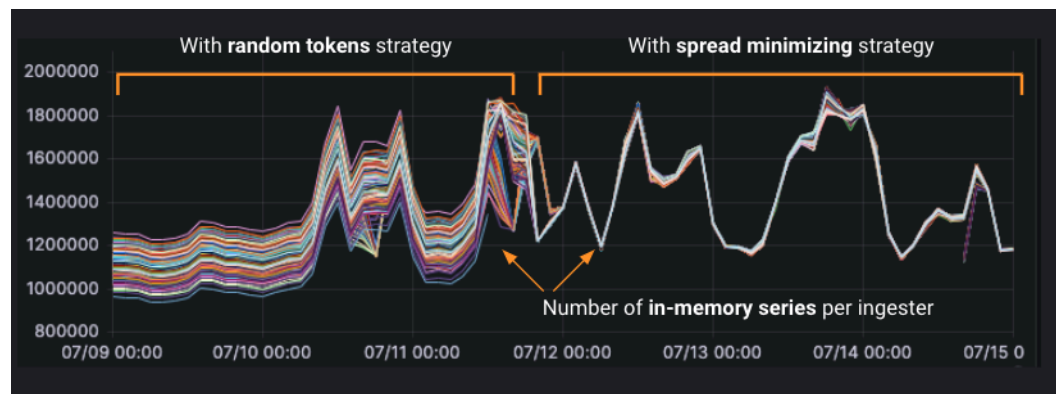


When early compaction triggers, in-memory series in that ingester drop to active series:



- Experimental ingester token spread minimizing strategy.
 - Useful without shuffle sharding. Do not try with shuffle sharding.
-

Learn more: <https://github.com/grafana/mimir/issues/4736>



- Improvements to OpenTelemetry push endpoint.
 - Optimizations.
 - Downscaling of exponential histograms.
 - [Jesús Vázquez] Prometheus is about to merge native OTEL ingestion, maybe some work could be reused in Mimir.
- What we're working on
 - [Krajo] Out of order support for native histograms in Prometheus / Mimir.
 - [Krajo] Looking to introduce experimental auto-scaling support in the Helm chart for some components.

2023-06-29

- We skipped the meeting because of low attendance and no news prepared (we already announced features coming in 2.9 last time, now it's just released)

2023-05-25 ([recording](#))

- What we're working on:
 - [Marco] Reduce querier memory utilization, running PromQL engine while reading chunks from ingesters (so progressively processing data instead of loading all data upfront in memory in the querier). The new experimental feature can be enabled by setting `-querier.prefer-streaming-chunks=true`.
 - [Vernon] Per-Series Extended Retention
 - Still in early development
 - Gives you the ability to select specific series to be retained for a longer time while other series are removed at the expiration of the blocks
 - Reduced storage TCO by allowing you to only keep the series you need and reduce the total storage amount
 - Allows for queries over a longer period of time for those selected series
 - [Dimitar] Reduce store-gateway memory utilization - reducing necessary volume of TSDB index to complete queries; looking to make this the default in 2.9.0
 - TL;DR querying long-term storage should be less resource-intensive
 - to try them out you can set `blocks_storage.bucket_store.series_selection_strategy`
 - possible values
 - `speculative` ← most aggressive optimization
 - `worst-case` ← a “good” tradeoff; recommended value
 - `worst-case-small-posting-lists`
 - `all` ← least aggressive; also same as previous behavior
 - [Marco] Reduce object storage API calls
 - Ruler: introduce event-based rules sync triggering, to resync rules once their config changes via API. This allows to relax the `-ruler.poll-interval`.
 - Compactor: removed unnecessary “object exists” API calls
- Mimir 2.9 will be cut next week
- [Andrea G] A new recommendation on the number of series per block from 25M to 8M; what's the reason, any tradeoffs?
 - [Marco] if you're not seeing problems, no need to carry out changes to accommodate the new recommendation
 - [Marco] Internally at Grafana Labs we've been targeting 8-10M series per block

- [Marco] Usually increasing provides better parallelism during query time
- Keeping number of shards as a power of 2 allows to change query sharding shards in the future

2023-04-27

- krajo Krajcsovits volunteered to lead the call 🎉
- krajo Krajcsovits Update about native histograms support
- 2.8.0 release is around the corner
 - Experimental Redis support
 - Experimental Vault support
- Jeffrey Sullivan would like to discuss [issue 4649](#); namely, the creation of a Mimir Kubernetes Operator
 - Scope? Primary: Govern for example ring size, and other mimir configuration
 - Secondary: deployment, rbac. S3 buckets
 - Grafana: putting it in Mimir repo would slow both of us down, but we can help with knowledge and PR reviews in your repo.
- Andrea Gardiman ingesters and store-gateways load distribution
 - Run with 64 tokens right now, because come from Consul and have ~600 ingesters
 - Blocked from changing number of tokens by <https://github.com/grafana/dskit/pull/79>
 - Thinking of trying with 8K token per ingester to minimize the difference between ingester series load
 - Grafana hasn't tested beyond 1K.
 - Changing the token number will cause reshuffle - resulting in more resource use
 - Post meeting: linked the ingester imbalance and dskit bugfix together.
 - Issue opened for the store-gateway imbalance <https://github.com/grafana/mimir/issues/4863>

2023-03-30

- New Maintainers: Congratulations! 🎉
 - [Charles Korn](#) being a new Mimir maintainer
 - [Wilfriedroset](#), [vaxvms](#) and [bubu11e](#) from OVHCloud being maintainers of DEB/RPM packages
- [Release 2.7](#) 🚀
 - Latest and greatest + 100% bug-free!
 - Lot of store-gateway improvements (streaming, no use of mmap)

- Query sharding can now reuse optimal shard count based on previous runs of the same query
- Ingestion of native histograms! (Querying coming in 2.8)
- New option to avoid merging OOO blocks with usual ones
 - Thanks [Jesús Vázquez](#)
 - Can solve compaction overhead if few OOO series are written over long (months) periods
- New option to add jitter to ingester TSDB compaction intervals, useful to spread the CPU/disk load over time instead of doing it at the same time.
- Next (not released, but worth a highlight): 🧐
 - Optimized regular expression matching in label matchers
 - Thanks [Marco Pracucci](#)
 - New protobuf result format between query-frontend & queries
 - Thanks [Charles Korn](#)
 - Will improve CPU and memory usage to encode and decode results
 - Experimental support for Redis as cache
 - Thanks [Nick Pillitteri](#)
 - Experimental support for Vault to store TLS certs
 - Thanks [Fayzal Ghantiwala](#)
 - Backfill improvements
 - Added optional index and chunks validation (both enabled by default, can be disabled per tenant)
 - Adding concurrency limit for validations
 - Thanks [Vernon Miller](#)
- We're working on adding autoscaling to more components.

2023-02-23 ([recording](#))

- Native histograms progress
 - Write path merged, read path to follow in next couple of weeks. OTEL endpoint ready for exponential histograms
 - Note: native histograms is work in progress in Prometheus as well, feature will be released as experimental
- Working on multiple performance improvements, see [CHANGELOG.md](#) for more details
 - Experimenting with taking into account query cardinality estimation for query-sharding
 - Useful to reduce the overhead of query-sharding on low-cardinality series

2023-01-26

- Mimir 2.6 release
 - RC: <https://github.com/grafana/mimir/releases/tag/mimir-2.6.0-rc.0>
 - Final release late week of January 30 if no issues are found
 - Added experimental streaming store-gateway; should reduce memory peaks handout-of-memory errors; enable with YAML option `blocks_storage: {bucket_store: {streaming_series_batch_size: 5000}}`
 - The store-gateway can now use an alternate code path to read index-headers that does not use memory mapped files. This feature can be enabled setting `-blocks-storage.bucket-store.index-header.stream-reader-enabled=true`.
 - We are moving the Helm documentation into a new place under <https://grafana.com/docs/helm-charts/> It is a work in progress, if you don't find something, please check the earlier version of Mimir docs.
- In the works: native histograms support talks at PromCon 2022 last year: <https://www.youtube.com/watch?v=RDdIDRkuVL8&t=2780s>
- Next:
 - Working on a design for [rollout-operator](#) to automate downscaling of the ingesters.
 - Just released v0.3.0 with a *no-downscale webhook*.
- QA
 - Questions from Theo & Mohamed from Giant Swarm who are considering Mimir
 - Asked about cluster federation feature
 - Questions from Sean from Bloomberg about MetricTank to Mimir migration.
-

2022-12-29

- No community call (Christmas break), see you in January!

2022-11-24

- Mimir 2.5 release
 - RC coming soon (next week)
 - No “big” features or changes, mostly bugfixes and performance optimizations.
 - New metric for detecting lagging compactor
 - The associated Helm 4.0 will bring breaking changes, in particular zone aware replication support - please consult the [changelog](#). Other notable changes:

revised sizing values files (small/large/etc) and less privileges required by mimir pods so they can run without root with restricted policy settings.

- Upcoming improvements in store-gateway
 - Removing mmap usage
 - Streaming series and chunks from store-gateway to querier (do not load all query data upfront in the store-gateway). We expect to reduce the store-gateway memory peaks, at the cost of slightly less performant queries for high cardinality or large time range queries (low cardinality queries should be unaffected).
- Some talks from PromCon 2022 that people interested in Mimir may find interesting:
 - [Anatomy of a Slow Query. and Lessons Learned Building Query Acceleration Techniques](#) (based on work in Mimir)
 - [Out Of Order Support in Prometheus](#) (available in Mimir)
 - [Native Histograms in Prometheus](#), [PromQL for Native Histograms](#) (coming to Mimir soon)
- **No community call in December (Christmas break), see you in January!**

2022-10-27 ([recording](#))

- Mimir 2.4 release
 - To be released 28th of October 2022 (tomorrow)
 - Query-scheduler ring-based service discovery
 - New API endpoint exposes per-tenant limits
 - Community contribution, thank you!
 - Maximum range query length limit
 - Helm chart improvements
 - Topology spread constraints
 - Support for Openshift routing (nginx ingress)
 - ... and more
 - [Anonymous usage statistics tracking](#) is enabled by default
 - See the [documentation](#) to learn about what data is collected or how to disable it (but if you keep it enabled, you help us to learn about how people use Grafana Mimir)
- Open mic / QA
 - Discussion about handling metrics from many different Kubernetes clusters

2022-09-29

- Mimir 2.3.1 released
 - Fixed exponential sharding time of queries with binary operators [#3027](#)
 - Stop panics on OTLP endpoint when a single metric has multiple timeseries. [#3040](#)
- Helm chart upcoming features
 - Published today version 3.1
 - Zone aware replication intro
 - Please read the changelog as applying next version 4.0 could cause data loss if migration isn't done properly.
 - If you're new to Mimir, we suggest you to run with multi-zone enabled.
 - Planned separation of documentation
- Ask us anything!
 - Nick Hippe:
 - Asking for help on a Cortex 1.9 -> Mimir migration
 - Context in the community slack thread: <https://grafana.slack.com/archives/C039863E8P7/p1664311947188919>

2022-08-25 ([recording](#))

- Mimir 2.3 release
 - RC coming soon (today?)
 - Ingest metrics in OpenTelemetry format
 - Increased instant query performance
 - Tenant federation for metadata queries
 - Simpler object storage configuration
 - DEB and RPM packages for Mimir (community contribution)
 - Import historic data to Grafana Mimir
 - New Helm chart minor release
- Usage stats
 - Anonymous collection of usage stats ([what is tracked](#))
 - Disabled by default in upcoming Mimir 2.3.0
 - Planning to enable by default in Mimir 2.4.0, but with single config parameter to opt out
- Open mic / QA

2022-07-28 ([recording](#))

Attendees:

Agenda:

- 🎉 [Mimir 2.2.0 released!](#)
 - Experimental support for out-of-order ingestion
 - Ease-of-use improvements (errors catalog)
 - Bucket index prefix
 - Faster ingester rollouts (2x faster WAL replay)
 - Store-gateway performance optimization
 - [Tons of Helm chart improvements](#)
- We open sourced [Graphite and Datadog ingestion proxies!](#)
- New blog posts
 - [New in Grafana Mimir: Ingest Graphite, Datadog, Influx, and Prometheus metrics into a single storage backend](#)
 - [How we improved Grafana Mimir query performance by up to 10x](#)
- What's next?
 - Experimental OpenTelemetry Metrics (OTLP) ingestion support
 - Added support for tenant federation to metric metadata endpoint
 - Experimental split-by-interval parallelization support for instant queries
 - Fixes and smaller improvements to Helm chart
- Open mic / QA

2022-06-30 ([recording](#))

Attendees: Peter Štibrány , Marco Pracucci , Arve Knudsen , Patrick Oyarzun , Bryan Boreham , Nick Pillitteri , RichiH Hartmann , krajo Krajcsovits , Mauro Stettler , Johanna Ratliff , Steve Simpson , Diego ROJAS, Niko, Michael Burt, Martin Beentjes

- [Mimir 2.2 release candidate](#) is now out
 - Experimental support for out-of-order ingestion
 - Ease-of-use improvements
 - Bucket index prefix
 - Faster ingester rollouts (WAL replay)
- Krajo: Helm Chart 3.0
 - This is the preliminary release notes ([subject to change, unfinished!](#)):
 - Contains breaking changes! We're working on documentation.
 - The Helm chart now supports installing Mimir on OpenShift!
 - The Helm chart can now easily deploy Grafana Agent to scrape metrics and logs from all Mimir pods and ship them to a remote store, making it

easy to monitor the health of your Mimir installation. For more information on how to set this up, see <docs link>.

- We've significantly leveled up the configuration experience for the Helm chart. A few of the most salient changes:
 - We've added an `extraEnvFrom` capability to all Mimir services to enable injecting secrets via environment variables.
 - We've made it possible to globally set environment variables and inject secrets across all pods in the chart using `global.extraEnv` and `global.extraEnvFrom`. (Note that the memcached and minio pods are not included.)
 - We've switched the default storage of the Mimir configuration from a Secret to a ConfigMap, which makes it easier to quickly diff your Mimir configuration between upgrades. We especially like the [Helm diff plugin](#) for this purpose.
 - We've added a `structuredConfig` option which allows overwriting specific key<=>value pairs in the `mimir.config` template, rather than having to maintain the entire `mimir.config` in your own `values.yaml`.
 - We've added the ability to create global pod annotations. This unlocks the ability to trigger a restart of all services in response to a single event, like the update of the secret containing Mimir's storage credentials.
- We've added support for the results cache, a cache used by the query-frontend, that should help improve query performance and reduce calls to object storage. This cache is enabled by default for users of the `small.yaml` and `large.yaml` values files.
- We've set the chart to disable `ingester.ring.unregister-on-shutdown` and `distributor.extend-writes` to make for a smoother upgrade experience. Now rolling restarts of ingesters should be less likely to cause spikes in resource usage.
- We've improved the documentation experience for the Helm chart by adding a 'Getting started with Mimir using the Helm chart'.
- We've added a smoke test for your Mimir cluster to help catch errors immediately after installing or upgrading via Helm. For more information, see <add link to 'helm test' header in getting started with mimi using the helm chart guide>.
- Currently working on a migration guide to help upgrade from 2.1 -> 3.0
- Open mic / Q&A / Feedback from the community?
 - Proxy mentioned during discussion for splitting incoming write requests to multiple tenants: <https://github.com/blind-oracle/cortex-tenant>
 -

2022-05-26 ([recording](#))

Attendees: Oleg Zaytsev Marco Pracucci Bryan Boreham krajo Krajcsovits
Patrick Oyarzun Patryk Prus Johanna Ratliff Giridharan Minivannan

- Mimir 2.1 is coming real soon now! (Out now)
 - Release candidate is out, please give it a try:
<https://github.com/grafana/mimir/releases/tag/mimir-2.1.0-rc.1>
 - Top features ([see release notes](#)):
 - Mimir on ARM: multiarch docker images.
 - Remote ruler mode for improved rule evaluation performance
 - Per-tenant custom trackers for monitoring cardinality
 - Reduce cardinality of Grafana Mimir's /metrics endpoint
 - We plan to release the stable version later **today**
- What's going on
 - Backfill API ([1694](#))
 - Allow users to backfill tsdb blocks into Grafana Mimir from other systems of from on-prem mimir to Grafana Cloud for example
 - Store-gateway thread pool ([1812](#))
 - Experimental solution to mmap page fault issue in store-gateways
 - Testing is still ongoing whether the new implementation is net-positive
 - Errors catalog ([1907](#) [1919](#) [1888](#))
 - Rationale: add a unique ID to each common error returned by API and/or logged, google it and get a runbook explaining the issue and how to resolve it.
 - Helm
 - Moving sources to Mimir repository.
 - This will enable us to make it first class citizen for Mimir in terms of testing and documentation. ([1925](#))
 - Will introduce mimir-distributed-beta chart
 - for new (possibly breaking features). While keeping mimir-distributed stable. This should enable us to add features more rapidly as before.
 - Query results cache support ([1411](#))
 - Multi zone support ([1205](#)) - wip
 - OpenShift support ([1329](#)) - wip
- Open mic / QA
 - [Giridharan] Using Mimir, hoping to bring it to production.
 - Question: Is there a good way in the dashboards to get an overview of all tenants?

- [Marco] In the dashboards, there are a small number of them which are tenant-focused. Most are aggregated across the whole cluster.
- [Marco] Querying data across tenants is available in Mimir
- [Giridharan] I'm interested in knowing whether a tenant is sending too many labels for example
- [Giridharan] Also, I've had trouble getting the dashboards working. Not sure if there are some missing labels and how to get those setup.
- [Marco] It's possible to rebuild the dashboards with custom label requirements
- [Krajo] With the Helm chart, the documentation is not great yet. We can help get it setup with the prometheus operator
- [Giridharan] I'd love to help improve the documentation as well
- [Krajo] We would love for you to try out our getting started guide once it's available
- [Community slack](#) plug: this is a great place to ask questions and we can help troubleshoot as you go about setting it up

2022-04-28 ([recording](#))

Attendees: Marco Pracucci Peter Štibrany Nick Pillitteri Johanna Ratliff Josh Abreu Mesa Tyler Reid Arve Knudsen krajo Krajcsovits Jéssica Lins Matt Mendick RichiH Hartmann samukher@redhat.com Masoud Kalali, Shengning Chen, Andre Ziviani

- [Peter] What's coming in future Mimir releases?
 - In Mimir 2.0, ruler has always executed queries on its own. In Mimir 2.1, ruler will be able to execute queries via query-frontend. Why is this great? Query-frontend implements multiple optimizations to query execution (eg. query-sharding), and this change enables ruler to benefit from optimizations in query-frontend.
 - Support for ingesting out-of-order samples is in the works. You can follow the progress in <https://github.com/grafana/mimir/pull/1753>.
 - Backfilling API for importing and validating existing blocks. PR in progress: <https://github.com/grafana/mimir/pull/1694>
 - Support for streaming chunks for remote-read API. PR in progress: <https://github.com/grafana/mimir/pull/1735>
 - Multiarch Docker images
- [Jesus]: Support for out of order samples - [Design doc](#)
- [Marco] Next release is scheduled for the week of 16th May ([schedule](#)). We need a volunteer to lead the release, among Mimir maintainers.
 - Johanna Ratliff

- [Marco] Open discussion: what's the top priority to get Mimir easier to install and operate?
 - Did you have any issues getting started?
 - Seen from community:
 - How to configure S3/GCS bucket authentication → opportunity to improve documentation ([issue](#))
 - Helm: configure storage in 1 single place
 - Helm: add configuration on how to configure some features in the Helm charts
 - Capacity planning (e.g. which VM type to use in the cloud?)
 - Questions about ruler and alertmanager config → opportunity to improve documentation ([issue](#))
 - Did you have any issues migrating?
 - Did anything hold you back from trying?
 - [Krajo] New and coming Helm features:
 - Support for multizone deployments (ingesters, store-gateways, alertmanagers)
 - Support for OpenShift
 - [Marco] Speaking about global configuration for storage
 - +1: Krajo, Mauro
 - *Open mic*

2022-03-31 ([recording](#))

- [Peter] Mimir launch party!
 - Grafana people talking about exciting features like query sharding, compactor, optimizations, labels caching, regexp optimizations, many more...
- [Marco] Allow to ingest metrics using OpenTelemetry, Influx, Graphite and DataDog
 - OTLP: <https://github.com/grafana/mimir/pull/695>
 - It will be possible to read ingested data using PromQL.
 - Support for other query languages (influx, graphite, datadog) will only be available in the Grafana Enterprise Metrics (GEM).
- [Jesus V] Allow ingesting out of order samples. For now work is in progress in <https://github.com/grafana/mimir-prometheus/tree/out-of-order>. We're working on the ingestion and query path in the tsdb and we will move forward with other bits (like compaction, WAL/mmap-file lifecycle handling etc) in coming weeks. We plan to have an MVP by the end of April.
 - Plan is to eventually upstream this into Prometheus.
- Open mic / QA
 - 10 minutes tutorial to run it locally with docker-compose: <https://grafana.com/tutorials/play-with-grafana-mimir/>
 - Can we get HA tracker (optional) to use memberlist, like hash ring?

- <https://grafana.com/docs/mimir/latest/operators-guide/configuring/configuring-high-availability-deduplication/>
- Bryan: This PR changed the implementation of HA-Tracker: <https://github.com/grafana/mimir/pull/271>
- Marco: we're changing to different replica when we don't get samples from the "current" one, so maybe using memberlist would work just fine in this case as well.
- Oleg: created an issue: <https://github.com/grafana/mimir/issues/1597>
- Mimir helm charts
 - <https://github.com/grafana/helm-charts/tree/main/charts/mimir-distributed>
 - [Migration guide from Cortex to Mimir using Helm](#)
- Migrating from Cortex to Mimir
 - [Migration guide](#)
 - [Mimirtool config conversion tool](#)
 - [Video tutorial using jsonnet](#)