

# Changelog

All notable changes to this project will be documented in this file.

The format is based on [Keep a Changelog](#).

This project adheres to [Semantic Versioning](#).

## Unreleased

### Added

### Changed

### Deprecated

### Removed

### Fixed

### Security

## 0.20.0 - 2021-04-23

### Added

- The OTLP exporter now has two new convenience functions, `NewExportPipeline` and `InstallNewPipeline`, setup and install the exporter in tracing and metrics pipelines. (#1373)
- Adds semantic conventions for exceptions. (#1492)
- Added Jaeger Environment variables: `OTEL_EXPORTER_JAEGER_AGENT_HOST`, `OTEL_EXPORTER_JAEGER_AGENT_PORT` These environment variables can be used to override Jaeger agent hostname and port (#1752)
- Option `ExportTimeout` was added to batch span processor. (#1755)
- `trace.TraceFlags` is now a defined type over `byte` and `WithSampled(bool) TraceFlags` and `IsSampled() bool` methods have been added to it. (#1770)
- The `Event` and `Link` struct types from the `go.opentelemetry.io/otel` package now include a `DroppedAttributeCount` field to record the number of attributes that were not recorded due to configured limits being reached. (#1771)
- The Jaeger exporter now reports dropped attributes for a Span event in the exported log. (#1771)
- Adds test to check BatchSpanProcessor ignores `OnEnd` and `ForceFlush` post `Shutdown`. (#1772)
- Extract resource attributes from the `OTEL_RESOURCE_ATTRIBUTES` environment variable and merge them with the `resource.Default` resource as well as resources provided to the `TracerProvider` and metric `Controller`. (#1785)
- Added `WithOSType` resource configuration option to set OS (Operating System) type resource attribute (`os.type`). (#1788)
- Added `WithProcess*` resource configuration options to set Process resource attributes. (#1788)
  - `process.pid`
  - `process.executable.name`
  - `process.executable.path`
  - `process.command_args`
  - `process.owner`

- `process.runtime.name`
- `process.runtime.version`
- `process.runtime.description`
- Adds `k8s.node.name` and `k8s.node.uid` attribute keys to the `semconv` package. (#1789)
- Added support for configuring OTLP/HTTP and OTLP/gRPC Endpoints, TLS Certificates, Headers, Compression and Timeout via Environment Variables. (#1758, #1769 and #1811)
  - `OTEL_EXPORTER_OTLP_ENDPOINT`
  - `OTEL_EXPORTER_OTLP_TRACES_ENDPOINT`
  - `OTEL_EXPORTER_OTLP_METRICS_ENDPOINT`
  - `OTEL_EXPORTER_OTLP_HEADERS`
  - `OTEL_EXPORTER_OTLP_TRACES_HEADERS`
  - `OTEL_EXPORTER_OTLP_METRICS_HEADERS`
  - `OTEL_EXPORTER_OTLP_COMPRESSION`
  - `OTEL_EXPORTER_OTLP_TRACES_COMPRESSION`
  - `OTEL_EXPORTER_OTLP_METRICS_COMPRESSION`
  - `OTEL_EXPORTER_OTLP_TIMEOUT`
  - `OTEL_EXPORTER_OTLP_TRACES_TIMEOUT`
  - `OTEL_EXPORTER_OTLP_METRICS_TIMEOUT`
  - `OTEL_EXPORTER_OTLP_CERTIFICATE`
  - `OTEL_EXPORTER_OTLP_TRACES_CERTIFICATE`
  - `OTEL_EXPORTER_OTLP_METRICS_CERTIFICATE`
- Adds `otlpgrpc.WithTimeout` option for configuring timeout to the otlp/gRPC exporter. (#1821)

## Fixed

- The `Span.IsRecording` implementation from `go.opentelemetry.io/otel/sdk/trace` always returns false when not being sampled. (#1750)
- The Jaeger exporter now correctly sets tags for the Span status code and message. This means it uses the correct tag keys ( `"otel.status_code"` , `"otel.status_description"` ) and does not set the status message as a tag unless it is set on the span. (#1761)
- The Jaeger exporter now correctly records Span event's names using the `"event"` key for a tag. Additionally, this tag is overridden, as specified in the OTel specification, if the event contains an attribute with that key. (#1768)
- Zipkin Exporter: Ensure mapping between OTel and Zipkin span data complies with the specification. (#1688)
- Fixed typo for default service name in Jaeger Exporter. (#1797)
- Fix flaky OTLP for the reconnection of the client connection. (#1527, #1814)

## Changed

- Span `RecordError` now records an `exception` event to comply with the semantic convention specification. (#1492)
- Jaeger exporter was updated to use thrift v0.14.1. (#1712)
- Migrate from using internally built and maintained version of the OTLP to the one hosted at `go.opentelemetry.io/proto/otlp` . (#1713)
- Migrate from using `github.com/gogo/protobuf` to `google.golang.org/protobuf` to match `go.opentelemetry.io/proto/otlp` . (#1713)
- The storage of a local or remote Span in a `context.Context` using its `SpanContext` is unified to store just the current Span. The Span's `SpanContext` can now self-identify as being remote or not. This means that

- `"go.opentelemetry.io/otel/trace".ContextWithRemoteSpanContext` will now overwrite any existing current Span, not just existing remote Spans, and make it the current Span in a `context.Context`. (#1731)
- Improve OTLP/gRPC exporter connection errors. (#1737)
- Information about a parent span context in a `"go.opentelemetry.io/otel/export/trace".SpanSnapshot` is unified in a new `Parent` field. The existing `ParentSpanID` and `HasRemoteParent` fields are removed in favor of this. (#1748)
- The `ParentContext` field of the `"go.opentelemetry.io/otel/sdk/trace".SamplingParameters` is updated to hold a `context.Context` containing the parent span. This changes it to make `SamplingParameters` conform with the OpenTelemetry specification. (#1749)
- Updated Jaeger Environment Variables: `JAEGER_ENDPOINT`, `JAEGER_USER`, `JAEGER_PASSWORD` to `OTEL_EXPORTER_JAEGER_ENDPOINT`, `OTEL_EXPORTER_JAEGER_USER`, `OTEL_EXPORTER_JAEGER_PASSWORD` in compliance with OTel spec (#1752)
- Modify `BatchSpanProcessor.ForceFlush` to abort after timeout/cancellation. (#1757)
- The `DroppedAttributeCount` field of the `Span` in the `go.opentelemetry.io/otel` package now only represents the number of attributes dropped for the span itself. It no longer is a conglomerate of itself, events, and link attributes that have been dropped. (#1771)
- Make `ExportSpans` in Jaeger Exporter honor context deadline. (#1773)
- Modify Zipkin Exporter default service name, use default resource's serviceName instead of empty. (#1777)
- The `go.opentelemetry.io/otel/sdk/export/trace` package is merged into the `go.opentelemetry.io/otel/sdk/trace` package. (#1778)
- The `prometheus.InstallNewPipeline` example is moved from comment to example test (#1796)
- The convenience functions for the stdout exporter have been updated to return the `TracerProvider` implementation and enable the shutdown of the exporter. (#1800)
- Replace the flush function returned from the Jaeger exporter's convenience creation functions ( `InstallNewPipeline` and `NewExportPipeline` ) with the `TracerProvider` implementation they create. This enables the caller to shutdown and flush using the related `TracerProvider` methods. (#1822)
- Updated the Jaeger exporter to have a default endpoint, `http://localhost:14250`, for the collector. (#1824)
- Changed the function `WithCollectorEndpoint` in the Jaeger exporter to no longer accept an endpoint as an argument. The endpoint can be passed with the `CollectorEndpointOption` using the `WithEndpoint` function or by setting the `OTEL_EXPORTER_JAEGER_ENDPOINT` environment variable value appropriately. (#1824)
- The Jaeger exporter no longer batches exported spans itself, instead it relies on the SDK's `BatchSpanProcessor` for this functionality. (#1830)
- The Jaeger exporter creation functions ( `NewRawExporter`, `NewExportPipeline`, and `InstallNewPipeline` ) no longer accept the removed `Option` type as a variadic argument. (#1830)

## Removed

- Removed Jaeger Environment variables: `JAEGER_SERVICE_NAME`, `JAEGER_DISABLED`, `JAEGER_TAGS` These environment variables will no longer be used to override values of the Jaeger exporter (#1752)
- No longer set the links for a `Span` in `go.opentelemetry.io/otel/sdk/trace` that is configured to be a new root. This is unspecified behavior that the OpenTelemetry community plans to standardize in the future. To prevent backwards incompatible changes when it is specified, these links are removed. (#1726)
- Setting error status while recording error with `Span` from `oteltest` package. (#1729)

- The concept of a remote and local Span stored in a context is unified to just the current Span. Because of this `"go.opentelemetry.io/otel/trace".RemoteSpanContextFromContext` is removed as it is no longer needed. Instead, `"go.opentelemetry.io/otel/trace".SpanContextFromContext` can be used to return the current Span. If needed, that Span's `SpanContext.IsRemote()` can then be used to determine if it is remote or not. (#1731)
- The `HasRemoteParent` field of the `"go.opentelemetry.io/otel/sdk/trace".SamplingParameters` is removed. This field is redundant to the information returned from the `Remote` method of the `SpanContext` held in the `ParentContext` field. (#1749)
- The `trace.FlagsDebug` and `trace.FlagsDeferred` constants have been removed and will be localized to the B3 propagator. (#1770)
- Remove `Process` configuration, `WithProcessFromEnv` and `ProcessFromEnv`, and type from the Jaeger exporter package. The information that could be configured in the `Process` struct should be configured in a `Resource` instead. (#1776, #1804)
- Remove the `WithDisabled` option from the Jaeger exporter. To disable the exporter unregister it from the `TracerProvider` or use a no-operation `TracerProvider`. (#1806)
- Removed the functions `CollectorEndpointFromEnv` and `WithCollectorEndpointOptionFromEnv` from the Jaeger exporter. These functions for retrieving specific environment variable values are redundant of other internal functions and are not intended for end user use. (#1824)
- Removed the Jaeger exporter `WithSDKOptions` `Option`. This option was used to set SDK options for the exporter creation convenience functions. These functions are provided as a way to easily setup or install the exporter with what are deemed reasonable SDK settings for common use cases. If the SDK needs to be configured differently, the `NewRawExporter` function and direct setup of the SDK with the desired settings should be used. (#1825)
- The `WithBufferMaxCount` and `WithBatchMaxCount` `Option`s from the Jaeger exporter are removed. The exporter no longer batches exports, instead relying on the SDK's `BatchSpanProcessor` for this functionality. (#1830)
- The Jaeger exporter `Option` type is removed. The type is no longer used by the exporter to configure anything. All of the previous configuration these options provided were duplicates of SDK configuration. They have all been removed in favor of using the SDK configuration and focuses the exporter configuration to be only about the endpoints it will send telemetry to. (#1830)

## 0.19.0 - 2021-03-18

### Added

- Added `Marshaler` config option to `otlphttp` to enable otlp over json or protobufs. (#1586)
- A `ForceFlush` method to the `"go.opentelemetry.io/otel/sdk/trace".TracerProvider` to flush all registered `SpanProcessor`s. (#1608)
- Added `WithSampler` and `WithSpanLimits` to tracer provider. (#1633, #1702)
- `"go.opentelemetry.io/otel/trace".SpanContext` now has a `remote` property, and `IsRemote()` predicate, that is true when the `SpanContext` has been extracted from remote context data. (#1701)
- A `Valid` method to the `"go.opentelemetry.io/otel/attribute".KeyValue` type. (#1703)

### Changed

- `trace.SpanContext` is now immutable and has no exported fields. (#1573)

- `trace.NewSpanContext()` can be used in conjunction with the `trace.SpanContextConfig` struct to initialize a new `SpanContext` where all values are known.
- Update the `ForceFlush` method signature to the `"go.opentelemetry.io/otel/sdk/trace".SpanProcessor` to accept a `context.Context` and return an error. (#1608)
- Update the `Shutdown` method to the `"go.opentelemetry.io/otel/sdk/trace".TracerProvider` return an error on shutdown failure. (#1608)
- The `SimpleSpanProcessor` will now shut down the enclosed `SpanExporter` and gracefully ignore subsequent calls to `OnEnd` after `Shutdown` is called. (#1612)
- `"go.opentelemetry.io/sdk/metric/controller.basic".WithPusher` is replaced with `WithExporter` to provide consistent naming across project. (#1656)
- Added non-empty string check for trace `Attribute` keys. (#1659)
- Add `description` to `SpanStatus` only when `StatusCode` is set to error. (#1662)
- Jaeger exporter falls back to `resource.Default` 's `service.name` if the exported Span does not have one. (#1673)
- Jaeger exporter populates Jaeger's Span Process from Resource. (#1673)
- Renamed the `LabelSet` method of `"go.opentelemetry.io/otel/sdk/resource".Resource` to `Set`. (#1692)
- Changed `WithSDK` to `WithSDKOptions` to accept variadic arguments of `TracerProviderOption` type in `go.opentelemetry.io/otel/exporters/trace/jaeger` package. (#1693)
- Changed `WithSDK` to `WithSDKOptions` to accept variadic arguments of `TracerProviderOption` type in `go.opentelemetry.io/otel/exporters/trace/zipkin` package. (#1693)

## Removed

- Removed `serviceName` parameter from Zipkin exporter and uses resource instead. (#1549)
- Removed `WithConfig` from tracer provider to avoid overriding configuration. (#1633)
- Removed the exported `SimpleSpanProcessor` and `BatchSpanProcessor` structs. These are now returned as a `SpanProcessor` interface from their respective constructors. (#1638)
- Removed `WithRecord()` from `trace.SpanOption` when creating a span. (#1660)
- Removed setting status to `Error` while recording an error as a span event in `RecordError`. (#1663)
- Removed `jaeger.WithProcess` configuration option. (#1673)
- Removed `ApplyConfig` method from `"go.opentelemetry.io/otel/sdk/trace".TracerProvider` and the now unneeded `Config` struct. (#1693)

## Fixed

- Jaeger Exporter: Ensure mapping between OTEL and Jaeger span data complies with the specification. (#1626)
- `SamplingResult.TraceState` is correctly propagated to a newly created span's `SpanContext`. (#1655)
- The `otel-collector` example now correctly flushes metric events prior to shutting down the exporter. (#1678)
- Do not set span status message in `SpanStatusFromHTTPStatusCode` if it can be inferred from `http.status_code`. (#1681)
- Synchronization issues in global trace delegate implementation. (#1686)

- Reduced excess memory usage by global `TracerProvider` . (#1687)

## 0.18.0 - 2021-03-03

### Added

- Added `resource.Default()` for use with meter and tracer providers. (#1507)
- `AttributePerEventCountLimit` and `AttributePerLinkCountLimit` for `SpanLimits` . (#1535)
- Added `Keys()` method to `propagation.TextMapCarrier` and `propagation.HeaderCarrier` to adapt `http.Header` to this interface. (#1544)
- Added `code` attributes to `go.opentelemetry.io/otel/semconv` package. (#1558)
- Compatibility testing suite in the CI system for the following systems. (#1567)

OS	Go Version	Architecture
Ubuntu	1.15	amd64
Ubuntu	1.14	amd64
Ubuntu	1.15	386
Ubuntu	1.14	386
MacOS	1.15	amd64
MacOS	1.14	amd64
Windows	1.15	amd64
Windows	1.14	amd64
Windows	1.15	386
Windows	1.14	386

### Changed

- Replaced interface `oteltest.SpanRecorder` with its existing implementation `StandardSpanRecorder` . (#1542)
- Default span limit values to 128. (#1535)
- Rename `MaxEventsPerSpan` , `MaxAttributesPerSpan` and `MaxLinksPerSpan` to `EventCountLimit` , `AttributeCountLimit` and `LinkCountLimit` , and move these fields into `SpanLimits` . (#1535)
- Renamed the `otel/label` package to `otel/attribute` . (#1541)
- Vendor the Jaeger exporter's dependency on Apache Thrift. (#1551)
- Parallelize the CI linting and testing. (#1567)
- Stagger timestamps in exact aggregator tests. (#1569)
- Changed all examples to use `WithBatchTimeout(5 * time.Second)` rather than `WithBatchTimeout(5)` . (#1621)
- Prevent end-users from implementing some interfaces (#1575)

```
"otel/exporters/otlp/otlphttp".Option
"otel/exporters/stdout".Option
"otel/oteltest".Option
```

```
"otel/trace".TracerOption
"otel/trace".SpanOption
"otel/trace".EventOption
"otel/trace".LifecycleOption
"otel/trace".InstrumentationOption
"otel/sdk/resource".Option
"otel/sdk/trace".ParentBasedSamplerOption
"otel/sdk/trace".ReadOnlySpan
"otel/sdk/trace".ReadWriteSpan
```

### Removed

- Removed attempt to resample spans upon changing the span name with `span.SetName()`. (#1545)
- The `test-benchmark` is no longer a dependency of the `precommit` make target. (#1567)
- Removed the `test-386` make target. This was replaced with a full compatibility testing suite (i.e. multi OS/arch) in the CI system. (#1567)

### Fixed

- The sequential timing check of timestamps in the stdout exporter are now setup explicitly to be sequential (#1571). (#1572)
- Windows build of Jaeger tests now compiles with OS specific functions (#1576). (#1577)
- The sequential timing check of timestamps of `go.opentelemetry.io/otel/sdk/metric/aggregator/lastvalue` are now setup explicitly to be sequential (#1578). (#1579)
- Validate tracestate header keys with vendors according to the W3C TraceContext specification (#1475). (#1581)
- The OTLP exporter includes related labels for translations of a GaugeArray (#1563). (#1570)

## [0.17.0](#) - 2021-02-12

### Changed

- Rename project default branch from `master` to `main`. (#1505)
- Reverse order in which `Resource` attributes are merged, per change in spec. (#1501)
- Add tooling to maintain "replace" directives in `go.mod` files automatically. (#1528)
- Create new modules: `otel/metric`, `otel/trace`, `otel/oteltest`, `otel/sdk/export/metric`, `otel/sdk/metric` (#1528)
- Move metric-related public global APIs from `otel` to `otel/metric/global`. (#1528)

### Fixed

- Fixed `otlpgrpc` reconnection issue.
- The example code in the README.md of `go.opentelemetry.io/otel/exporters/otlp` is moved to a compiled example test and used the new `WithAddress` instead of `WithEndpoint`. (#1513)
- The `otel-collector` example now uses the default OTLP receiver port of the collector.

## [0.16.0](#) - 2021-01-13

### Added

- Add the `ReadOnlySpan` and `ReadWriteSpan` interfaces to provide better control for accessing span data. (#1360)
- `NewGRPCDriver` function returns a `ProtocolDriver` that maintains a single gRPC connection to the collector. (#1369)

- Added documentation about the project's versioning policy. (#1388)
- Added `NewSplitDriver` for OTLP exporter that allows sending traces and metrics to different endpoints. (#1418)
- Added codeql workflow to GitHub Actions (#1428)
- Added Gosec workflow to GitHub Actions (#1429)
- Add new HTTP driver for OTLP exporter in `exporters/otlp/otlphttp`. Currently it only supports the binary protobuf payloads. (#1420)
- Add an OpenCensus exporter bridge. (#1444)

## Changed

- Rename `internal/testing` to `internal/internaltest`. (#1449)
- Rename `export.SpanData` to `export.SpanSnapshot` and use it only for exporting spans. (#1360)
- Store the parent's full `SpanContext` rather than just its span ID in the `span` struct. (#1360)
- Improve span duration accuracy. (#1360)
- Migrated CI/CD from CircleCI to GitHub Actions (#1382)
- Remove duplicate checkout from GitHub Actions workflow (#1407)
- Metric `array` aggregator renamed `exact` to match its `aggregation.Kind` (#1412)
- Metric `exact` aggregator includes per-point timestamps (#1412)
- Metric stdout exporter uses MinMaxSumCount aggregator for ValueRecorder instruments (#1412)
- `NewExporter` from `exporters/otlp` now takes a `ProtocolDriver` as a parameter. (#1369)
- Many OTLP Exporter options became gRPC ProtocolDriver options. (#1369)
- Unify endpoint API that related to OTel exporter. (#1401)
- Optimize metric histogram aggregator to re-use its slice of buckets. (#1435)
- Metric aggregator `Count()` and histogram `Bucket.Counts` are consistently `uint64`. (#1430)
- Histogram aggregator accepts functional options, uses default boundaries if none given. (#1434)
- `SamplingResult` now passed a `Tracestate` from the parent `SpanContext` (#1432)
- Moved gRPC driver for OTLP exporter to `exporters/otlp/otlpgrpc`. (#1420)
- The `TraceContext` propagator now correctly propagates `TraceState` through the `SpanContext`. (#1447)
- Metric Push and Pull Controller components are combined into a single "basic" Controller:
  - `WithExporter()` and `Start()` to configure Push behavior
  - `Start()` is optional; use `Collect()` and `ForEach()` for Pull behavior
  - `Start()` and `Stop()` accept Context. (#1378)
- The `Event` type is moved from the `otel/sdk/export/trace` package to the `otel/trace` API package. (#1452)

## Removed

- Remove `errUninitializedSpan` as its only usage is now obsolete. (#1360)
- Remove Metric export functionality related to quantiles and summary data points: this is not specified (#1412)
- Remove DDSketch metric aggregator; our intention is to re-introduce this as an option of the histogram aggregator after [new OTLP histogram data types](#) are released (#1412)

## Fixed

- `BatchSpanProcessor.Shutdown()` will now shutdown underlying `export.SpanExporter`. (#1443)

## 0.15.0 - 2020-12-10



## Added

- The `WithIDGenerator` `TracerProviderOption` is added to the `go.opentelemetry.io/otel/trace` package to configure an `IDGenerator` for the `TracerProvider`. (#1363)

## Changed

- The Zipkin exporter now uses the Span status code to determine. (#1328)
- `NewExporter` and `Start` functions in `go.opentelemetry.io/otel/exporters/otlp` now receive `context.Context` as a first parameter. (#1357)
- Move the OpenCensus example into `example` directory. (#1359)
- Moved the SDK's `internal.IDGenerator` interface in to the `sdk/trace` package to enable support for externally-defined ID generators. (#1363)
- Bump `github.com/google/go-cmp` from 0.5.3 to 0.5.4 (#1374)
- Bump `github.com/golangci/golangci-lint` in `/internal/tools` (#1375)

## Fixed

- Metric SDK `SumObserver` and `UpDownSumObserver` instruments correctness fixes. (#1381)

## [0.14.0](#) - 2020-11-19

## Added

- An `EventOption` and the related `NewEventConfig` function are added to the `go.opentelemetry.io/otel` package to configure Span events. (#1254)
- A `TextMapPropagator` and associated `TextMapCarrier` are added to the `go.opentelemetry.io/otel/oteltest` package to test `TextMap` type propagators and their use. (#1259)
- `SpanContextFromContext` returns `SpanContext` from context. (#1255)
- `TraceState` has been added to `SpanContext`. (#1340)
- `DeploymentEnvironmentKey` added to `go.opentelemetry.io/otel/semconv` package. (#1323)
- Add an OpenCensus to OpenTelemetry tracing bridge. (#1305)
- Add a parent context argument to `SpanProcessor.OnStart` to follow the specification. (#1333)
- Add missing tests for `sdk/trace/attributes_map.go`. (#1337)

## Changed

- Move the `go.opentelemetry.io/otel/api/trace` package into `go.opentelemetry.io/otel/trace` with the following changes. (#1229) (#1307)
  - `ID` has been renamed to `TraceID`.
  - `IDFromHex` has been renamed to `TraceIDFromHex`.
  - `EmptySpanContext` is removed.
- Move the `go.opentelemetry.io/otel/api/trace/tracetest` package into `go.opentelemetry.io/otel/oteltest`. (#1229)
- OTLP Exporter updates:
  - supports OTLP v0.6.0 (#1230, #1354)
  - supports configurable aggregation temporality (default: Cumulative, optional: Stateless). (#1296)
- The Sampler is now called on local child spans. (#1233)

- The `Kind` type from the `go.opentelemetry.io/otel/api/metric` package was renamed to `InstrumentKind` to more specifically describe what it is and avoid semantic ambiguity. (#1240)
- The `MetricKind` method of the `Descriptor` type in the `go.opentelemetry.io/otel/api/metric` package was renamed to `Descriptor.InstrumentKind`. This matches the returned type and fixes misuse of the term metric. (#1240)
- Move test harness from the `go.opentelemetry.io/otel/api/apitest` package into `go.opentelemetry.io/otel/oteltest`. (#1241)
- Move the `go.opentelemetry.io/otel/api/metric/metrictest` package into `go.opentelemetry.io/oteltest` as part of #964. (#1252)
- Move the `go.opentelemetry.io/otel/api/metric` package into `go.opentelemetry.io/otel/metric` as part of #1303. (#1321)
- Move the `go.opentelemetry.io/otel/api/metric/registry` package into `go.opentelemetry.io/otel/metric/registry` as a part of #1303. (#1316)
- Move the `Number` type (together with related functions) from `go.opentelemetry.io/otel/api/metric` package into `go.opentelemetry.io/otel/metric/number` as a part of #1303. (#1316)
- The function signature of the `Span.AddEvent` method in `go.opentelemetry.io/otel` is updated to no longer take an unused context and instead take a required name and a variable number of `EventOption`s. (#1254)
- The function signature of the `Span.RecordError` method in `go.opentelemetry.io/otel` is updated to no longer take an unused context and instead take a required error value and a variable number of `EventOption`s. (#1254)
- Move the `go.opentelemetry.io/otel/api/global` package to `go.opentelemetry.io/otel`. (#1262) (#1330)
- Move the `Version` function from `go.opentelemetry.io/otel/sdk` to `go.opentelemetry.io/otel`. (#1330)
- Rename correlation context header from `"otcorrelations"` to `"baggage"` to match the OpenTelemetry specification. (#1267)
- Fix `Code.UnmarshalJSON` to work with valid JSON only. (#1276)
- The `resource.New()` method changes signature to support builtin attributes and functional options, including `telemetry.sdk.*` and `host.name` semantic conventions; the former method is renamed `resource.NewWithAttributes`. (#1235)
- The Prometheus exporter now exports non-monotonic counters (i.e. `UpDownCounter`s) as gauges. (#1210)
- Correct the `Span.End` method documentation in the `otel` API to state updates are not allowed on a span after it has ended. (#1310)
- Updated span collection limits for attribute, event and link counts to 1000 (#1318)
- Renamed `semconv.HTTPURLKey` to `semconv.HTTPURLKey`. (#1338)

## Removed

- The `ErrInvalidHexID`, `ErrInvalidTraceIDLength`, `ErrInvalidSpanIDLength`, `ErrInvalidSpanIDLength`, or `ErrNilSpanID` from the `go.opentelemetry.io/otel` package are unexported now. (#1243)
- The `AddEventWithTimestamp` method on the `Span` interface in `go.opentelemetry.io/otel` is removed due to its redundancy. It is replaced by using the `AddEvent` method with a `WithTimestamp` option. (#1254)

- The `MockSpan` and `MockTracer` types are removed from `go.opentelemetry.io/otel/oteltest`. `Tracer` and `Span` from the same module should be used in their place instead. (#1306)
- `WorkerCount` option is removed from `go.opentelemetry.io/otel/exporters/otlp`. (#1350)
- Remove the following labels types: `INT32`, `UINT32`, `UINT64` and `FLOAT32`. (#1314)

## Fixed

- Rename `MergeItererator` to `MergeIterator` in the `go.opentelemetry.io/otel/label` package. (#1244)
- The `go.opentelemetry.io/otel/api/global` packages `global TextMapPropagator` now delegates functionality to a globally set delegate for all previously returned propagators. (#1258)
- Fix condition in `label.Any`. (#1299)
- Fix global `TracerProvider` to pass options to its configured provider. (#1329)
- Fix missing handler for `ExactKind` aggregator in OTLP metrics transformer (#1309)

## [0.13.0](#) - 2020-10-08

### Added

- OTLP Metric exporter supports Histogram aggregation. (#1209)
- The `Code` struct from the `go.opentelemetry.io/otel/codes` package now supports JSON marshaling and unmarshaling as well as implements the `Stringer` interface. (#1214)
- A Baggage API to implement the OpenTelemetry specification. (#1217)
- Add Shutdown method to `sdk/trace/provider`, shutdown processors in the order they were registered. (#1227)

### Changed

- Set default propagator to no-op propagator. (#1184)
- The `HTTPSupplier`, `HTTPExtractor`, `HTTPInjector`, and `HTTPPropagator` from the `go.opentelemetry.io/otel/api/propagation` package were replaced with unified `TextMapCarrier` and `TextMapPropagator` in the `go.opentelemetry.io/otel/propagation` package. (#1212) (#1325)
- The `New` function from the `go.opentelemetry.io/otel/api/propagation` package was replaced with `NewCompositeTextMapPropagator` in the `go.opentelemetry.io/otel` package. (#1212)
- The status codes of the `go.opentelemetry.io/otel/codes` package have been updated to match the latest OpenTelemetry specification. They now are `Unset`, `Error`, and `Ok`. They no longer track the gRPC codes. (#1214)
- The `StatusCode` field of the `SpanData` struct in the `go.opentelemetry.io/otel/sdk/export/trace` package now uses the codes package from this package instead of the gRPC project. (#1214)
- Move the `go.opentelemetry.io/otel/api/baggage` package into `go.opentelemetry.io/otel/baggage`. (#1217) (#1325)
- A `Shutdown` method of `SpanProcessor` and all its implementations receives a context and returns an error. (#1264)

### Fixed

- Copies of data from arrays and slices passed to `go.opentelemetry.io/otel/label.ArrayValue()` are now used in the returned `Value` instead of using the mutable data itself. (#1226)

## Removed

- The `ExtractHTTP` and `InjectHTTP` functions from the `go.opentelemetry.io/otel/api/propagation` package were removed. (#1212)
- The `Propagators` interface from the `go.opentelemetry.io/otel/api/propagation` package was removed to conform to the OpenTelemetry specification. The explicit `TextMapPropagator` type can be used in its place as this is the `Propagator` type the specification defines. (#1212)
- The `SetAttribute` method of the `Span` from the `go.opentelemetry.io/otel/api/trace` package was removed given its redundancy with the `SetAttributes` method. (#1216)
- The internal implementation of Baggage storage is removed in favor of using the new Baggage API functionality. (#1217)
- Remove duplicate hostname key `HostHostNameKey` in Resource semantic conventions. (#1219)
- Nested array/slice support has been removed. (#1226)

## [0.12.0](#) - 2020-09-24

### Added

- A `SpanConfigure` function in `go.opentelemetry.io/otel/api/trace` to create a new `SpanConfig` from `SpanOptions`. (#1108)
- In the `go.opentelemetry.io/otel/api/trace` package, `NewTracerConfig` was added to construct new `TracerConfig`s. This addition was made to conform with our project option conventions. (#1155)
- Instrumentation library information was added to the Zipkin exporter. (#1119)
- The `SpanProcessor` interface now has a `ForceFlush()` method. (#1166)
- More semantic conventions for k8s as resource attributes. (#1167)

### Changed

- Add reconnecting udp connection type to Jaeger exporter. This change adds a new optional implementation of the `udpConn` interface used to detect changes to an agent's host dns record. It then adopts the new destination address to ensure the exporter doesn't get stuck. This change was ported from `jaegertracing/jaeger-client-go#520`. (#1063)
- Replace `StartOption` and `EndOption` in `go.opentelemetry.io/otel/api/trace` with `SpanOption`. This change is matched by replacing the `StartConfig` and `EndConfig` with a unified `SpanConfig`. (#1108)
- Replace the `LinkedTo` span option in `go.opentelemetry.io/otel/api/trace` with `WithLinks`. This is be more consistent with our other option patterns, i.e. passing the item to be configured directly instead of its component parts, and provides a cleaner function signature. (#1108)
- The `go.opentelemetry.io/otel/api/trace` `TracerOption` was changed to an interface to conform to project option conventions. (#1109)
- Move the `B3` and `TraceContext` from within the `go.opentelemetry.io/otel/api/trace` package to their own `go.opentelemetry.io/otel/propagators` package. This removal of the propagators is reflective of the OpenTelemetry specification for these propagators as well as cleans up the `go.opentelemetry.io/otel/api/trace` API. (#1118)
- Rename Jaeger tags used for instrumentation library information to reflect changes in OpenTelemetry specification. (#1119)
- Rename `ProbabilitySampler` to `TraceIDRatioBased` and change semantics to ignore parent span sampling status. (#1115)
- Move `tools` package under `internal`. (#1141)

- Move `go.opentelemetry.io/otel/api/correlation` package to `go.opentelemetry.io/otel/api/baggage` . (#1142) The `correlation.Context` propagator has been renamed `baggage.Baggage` . Other exported functions and types are unchanged.
- Rename `ParentOrElse` sampler to `ParentBased` and allow setting samplers depending on parent span. (#1153)
- In the `go.opentelemetry.io/otel/api/trace` package, `SpanConfigure` was renamed to `NewSpanConfig` . (#1155)
- Change `dependabot.yml` to add a `Skip Changelog` label to dependabot-sourced PRs. (#1161)
- The [configuration style guide](#) has been updated to recommend the use of `newConfig()` instead of `configure()` . (#1163)
- The `otlp.Config` type has been unexported and changed to `otlp.config` , along with its initializer. (#1163)
- Ensure exported interface types include parameter names and update the Style Guide to reflect this styling rule. (#1172)
- Don't consider unset environment variable for resource detection to be an error. (#1170)
- Rename `go.opentelemetry.io/otel/api/metric.ConfigureInstrument` to `NewInstrumentConfig` and `go.opentelemetry.io/otel/api/metric.ConfigureMeter` to `NewMeterConfig` .
- `ValueObserver` instruments use `LastValue` aggregator by default. (#1165)
- OTLP Metric exporter supports `LastValue` aggregation. (#1165)
- Move the `go.opentelemetry.io/otel/api/unit` package to `go.opentelemetry.io/otel/unit` . (#1185)
- Rename `Provider` to `MeterProvider` in the `go.opentelemetry.io/otel/api/metric` package. (#1190)
- Rename `NoopProvider` to `NoopMeterProvider` in the `go.opentelemetry.io/otel/api/metric` package. (#1190)
- Rename `NewProvider` to `NewMeterProvider` in the `go.opentelemetry.io/otel/api/metric/metricstest` package. (#1190)
- Rename `Provider` to `MeterProvider` in the `go.opentelemetry.io/otel/api/metric/registry` package. (#1190)
- Rename `NewProvider` to `NewMeterProvider` in the `go.opentelemetry.io/otel/api/metric/registryc` package. (#1190)
- Rename `Provider` to `TracerProvider` in the `go.opentelemetry.io/otel/api/trace` package. (#1190)
- Rename `NoopProvider` to `NoopTracerProvider` in the `go.opentelemetry.io/otel/api/trace` package. (#1190)
- Rename `Provider` to `TracerProvider` in the `go.opentelemetry.io/otel/api/trace/tracetest` package. (#1190)
- Rename `NewProvider` to `NewTracerProvider` in the `go.opentelemetry.io/otel/api/trace/tracetest` package. (#1190)
- Rename `WrapperProvider` to `WrapperTracerProvider` in the `go.opentelemetry.io/otel/bridge/opentracing` package. (#1190)
- Rename `NewWrapperProvider` to `NewWrapperTracerProvider` in the `go.opentelemetry.io/otel/bridge/opentracing` package. (#1190)
- Rename `Provider` method of the pull controller to `MeterProvider` in the `go.opentelemetry.io/otel/sdk/metric/controller/pull` package. (#1190)

- Rename `Provider` method of the push controller to `MeterProvider` in the `go.opentelemetry.io/otel/sdk/metric/controller/push` package. (#1190)
- Rename `ProviderOptions` to `TracerProviderConfig` in the `go.opentelemetry.io/otel/sdk/trace` package. (#1190)
- Rename `ProviderOption` to `TracerProviderOption` in the `go.opentelemetry.io/otel/sdk/trace` package. (#1190)
- Rename `Provider` to `TracerProvider` in the `go.opentelemetry.io/otel/sdk/trace` package. (#1190)
- Rename `NewProvider` to `NewTracerProvider` in the `go.opentelemetry.io/otel/sdk/trace` package. (#1190)
- Renamed `SamplingDecision` values to comply with OpenTelemetry specification change. (#1192)
- Renamed Zipkin attribute names from `ot.status_code` & `ot.status_description` to `otel.status_code` & `otel.status_description`. (#1201)
- The default SDK now invokes registered `SpanProcessor`s in the order they were registered with the `TracerProvider`. (#1195)
- Add test of spans being processed by the `SpanProcessor`s in the order they were registered. (#1203)

## Removed

- Remove the B3 propagator from `go.opentelemetry.io/otel/propagators`. It is now located in the `go.opentelemetry.io/contrib/propagators/` module. (#1191)
- Remove the semantic convention for HTTP status text, `HTTPStatusTextKey` from package `go.opentelemetry.io/otel/semconv`. (#1194)

## Fixed

- Zipkin example no longer mentions `ParentSampler`, corrected to `ParentBased`. (#1171)
- Fix missing shutdown processor in otel-collector example. (#1186)
- Fix missing shutdown processor in basic and namedtracer examples. (#1197)

## 0.11.0 - 2020-08-24

### Added

- Support for exporting array-valued attributes via OTLP. (#992)
- `Noop` and `InMemory` `SpanBatcher` implementations to help with testing integrations. (#994)
- Support for filtering metric label sets. (#1047)
- A dimensionality-reducing metric Processor. (#1057)
- Integration tests for more OTel Collector Attribute types. (#1062)
- A new `WithSpanProcessor` `ProviderOption` is added to the `go.opentelemetry.io/otel/sdk/trace` package to create a `Provider` and automatically register the `SpanProcessor`. (#1078)

### Changed

- Rename `sdk/metric/processor/test` to `sdk/metric/processor/processortest`. (#1049)
- Rename `sdk/metric/controller/test` to `sdk/metric/controller/controllertest`. (#1049)
- Rename `api/testharness` to `api/apitest`. (#1049)
- Rename `api/trace/testtrace` to `api/trace/tracetest`. (#1049)
- Change Metric Processor to merge multiple observations. (#1024)
- The `go.opentelemetry.io/otel/bridge/opentracing` bridge package has been made into its own module. This removes the package dependencies of this bridge from the rest of the OpenTelemetry based

project. (#1038)

- Renamed `go.opentelemetry.io/otel/api/standard` package to `go.opentelemetry.io/otel/semconv` to avoid the ambiguous and generic name `standard` and better describe the package as containing OpenTelemetry semantic conventions. (#1016)
- The environment variable used for resource detection has been changed from `OTEL_RESOURCE_LABELS` to `OTEL_RESOURCE_ATTRIBUTES`. (#1042)
- Replace `WithSyncer` with `WithBatcher` in examples. (#1044)
- Replace the `google.golang.org/grpc/codes` dependency in the API with an equivalent `go.opentelemetry.io/otel/codes` package. (#1046)
- Merge the `go.opentelemetry.io/otel/api/label` and `go.opentelemetry.io/otel/api/kv` into the new `go.opentelemetry.io/otel/label` package. (#1060)
- Unify Callback Function Naming. Rename `*Callback` with `*Func`. (#1061)
- CI builds validate against last two versions of Go, dropping 1.13 and adding 1.15. (#1064)
- The `go.opentelemetry.io/otel/sdk/export/trace` interfaces `SpanSyncer` and `SpanBatcher` have been replaced with a specification compliant `Exporter` interface. This interface still supports the export of `SpanData`, but only as a slice. Implementation are also required now to return any error from `ExportSpans` if one occurs as well as implement a `Shutdown` method for exporter clean-up. (#1078)
- The `go.opentelemetry.io/otel/sdk/trace` `NewBatchSpanProcessor` function no longer returns an error. If a `nil` exporter is passed as an argument to this function, instead of it returning an error, it now returns a `BatchSpanProcessor` that handles the export of `SpanData` by not taking any action. (#1078)
- The `go.opentelemetry.io/otel/sdk/trace` `NewProvider` function to create a `Provider` no longer returns an error, instead only a `*Provider`. This change is related to `NewBatchSpanProcessor` not returning an error which was the only error this function would return. (#1078)

## Removed

- Duplicate, unused API sampler interface. (#999) Use the [Sampler interface](#) provided by the SDK instead.
- The `grpctrace` instrumentation was moved to the `go.opentelemetry.io/contrib` repository and out of this repository. This move includes moving the `grpc` example to the `go.opentelemetry.io/contrib` as well. (#1027)
- The `WithSpan` method of the `Tracer` interface. The functionality this method provided was limited compared to what a user can provide themselves. It was removed with the understanding that if there is sufficient user need it can be added back based on actual user usage. (#1043)
- The `RegisterSpanProcessor` and `UnregisterSpanProcessor` functions. These were holdovers from an approach prior to the `TracerProvider` design. They were not used anymore. (#1077)
- The `oterror` package. (#1026)
- The `othttp` and `httptrace` instrumentations were moved to `go.opentelemetry.io/contrib`. (#1032)

## Fixed

- The `semconv.HTTPServerMetricAttributesFromHTTPRequest()` function no longer generates the high-cardinality `http.request.content.length` label. (#1031)
- Correct instrumentation version tag in Jaeger exporter. (#1037)
- The SDK span will now set an error event if the `End` method is called during a panic (i.e. it was deferred). (#1043)

- Move internally generated protobuf code from the `go.opentelemetry.io/otel` to the OTLP exporter to reduce dependency overhead. (#1050)
- The `otel-collector` example referenced outdated collector processors. (#1006)

## 0.10.0 - 2020-07-29

This release migrates the default OpenTelemetry SDK into its own Go module, decoupling the SDK from the API and reducing dependencies for instrumentation packages.

### Added

- The Zipkin exporter now has `NewExportPipeline` and `InstallNewPipeline` constructor functions to match the common pattern. These function build a new exporter with default SDK options and register the exporter with the `global` package respectively. (#944)
- Add propagator option for gRPC instrumentation. (#986)
- The `testtrace` package now tracks the `trace.SpanKind` for each span. (#987)

### Changed

- Replace the `RegisterGlobal Option` in the Jaeger exporter with an `InstallNewPipeline` constructor function. This matches the other exporter constructor patterns and will register a new exporter after building it with default configuration. (#944)
- The `trace ( go.opentelemetry.io/otel/exporters/trace/stdout )` and `metric ( go.opentelemetry.io/otel/exporters/metric/stdout ) stdout` exporters are now merged into a single exporter at `go.opentelemetry.io/otel/exporters/stdout`. This new exporter was made into its own Go module to follow the pattern of all exporters and decouple it from the `go.opentelemetry.io/otel` module. (#956, #963)
- Move the `go.opentelemetry.io/otel/exporters/test` test package to `go.opentelemetry.io/otel/sdk/export/metric/metrictest`. (#962)
- The `go.opentelemetry.io/otel/api/kv/value` package was merged into the parent `go.opentelemetry.io/otel/api/kv` package. (#968)
  - `value.Bool` was replaced with `kv.BoolValue`.
  - `value.Int64` was replaced with `kv.Int64Value`.
  - `value.Uint64` was replaced with `kv.Uint64Value`.
  - `value.Float64` was replaced with `kv.Float64Value`.
  - `value.Int32` was replaced with `kv.Int32Value`.
  - `value.Uint32` was replaced with `kv.Uint32Value`.
  - `value.Float32` was replaced with `kv.Float32Value`.
  - `value.String` was replaced with `kv.StringValue`.
  - `value.Int` was replaced with `kv.IntValue`.
  - `value.Uint` was replaced with `kv.UintValue`.
  - `value.Array` was replaced with `kv.ArrayValue`.
- Rename `Infer` to `Any` in the `go.opentelemetry.io/otel/api/kv` package. (#972)
- Change `othttp` to use the `httpsnoop` package to wrap the `ResponseWriter` so that optional interfaces ( `http.Hijacker` , `http.Flusher` , etc.) that are implemented by the original `ResponseWriter` are also implemented by the wrapped `ResponseWriter`. (#979)
- Rename `go.opentelemetry.io/otel/sdk/metric/aggregator/test` package to `go.opentelemetry.io/otel/sdk/metric/aggregator/aggregatortest`. (#980)
- Make the SDK into its own Go module called `go.opentelemetry.io/otel/sdk`. (#985)



- Changed the default trace `Sampler` from `AlwaysOn` to `ParentOrElse (AlwaysOn)` . (#989)

#### Removed

- The `IndexedAttribute` function from the `go.opentelemetry.io/otel/api/label` package was removed in favor of `IndexedLabel` which it was synonymous with. (#970)

#### Fixed

- Bump `github.com/golangci/golangci-lint` from 1.28.3 to 1.29.0 in `/tools`. (#953)
- Bump `github.com/google/go-cmp` from 0.5.0 to 0.5.1. (#957)
- Use `global.Handle` for span export errors in the OTLP exporter. (#946)
- Correct Go language formatting in the README documentation. (#961)
- Remove default SDK dependencies from the `go.opentelemetry.io/otel/api` package. (#977)
- Remove default SDK dependencies from the `go.opentelemetry.io/otel/instrumentation` package. (#983)
- Move documented examples for `go.opentelemetry.io/otel/instrumentation/grpctrace` interceptors into Go example tests. (#984)

### 0.9.0 - 2020-07-20

#### Added

- A new Resource Detector interface is included to allow resources to be automatically detected and included. (#939)
- A Detector to automatically detect resources from an environment variable. (#939)
- Github action to generate protobuf Go bindings locally in `internal/opentelemetry-proto-gen` . (#938)
- OTLP .proto files from `open-telemetry/opentelemetry-proto` imported as a git submodule under `internal/opentelemetry-proto` . References to `github.com/open-telemetry/opentelemetry-proto` changed to `go.opentelemetry.io/otel/internal/opentelemetry-proto-gen` . (#942)

#### Changed

- Non-nil value `struct s` for key-value pairs will be marshalled using JSON rather than `Sprintf` . (#948)

#### Removed

- Removed dependency on `github.com/open-telemetry/opentelemetry-collector` . (#943)

### 0.8.0 - 2020-07-09

#### Added

- The `B3Encoding` type to represent the B3 encoding(s) the B3 propagator can inject. A value for HTTP supported encodings (Multiple Header: `MultipleHeader` , Single Header: `SingleHeader` ) are included. (#882)
- The `FlagsDeferred` trace flag to indicate if the trace sampling decision has been deferred. (#882)
- The `FlagsDebug` trace flag to indicate if the trace is a debug trace. (#882)
- Add `peer.service` semantic attribute. (#898)
- Add database-specific semantic attributes. (#899)
- Add semantic convention for `faas.coldstart` and `container.id` . (#909)
- Add http content size semantic conventions. (#905)

- Include `http.request_content_length` in HTTP request basic attributes. (#905)
- Add semantic conventions for operating system process resource attribute keys. (#919)
- The Jaeger exporter now has a `WithBatchMaxCount` option to specify the maximum number of spans sent in a batch. (#931)

## Changed

- Update `CONTRIBUTING.md` to ask for updates to `CHANGELOG.md` with each pull request. (#879)
- Use lowercase header names for B3 Multiple Headers. (#881)
- The B3 propagator `SingleHeader` field has been replaced with `InjectEncoding`. This new field can be set to combinations of the `B3Encoding` bitmasks and will inject trace information in these encodings. If no encoding is set, the propagator will default to `MultipleHeader` encoding. (#882)
- The B3 propagator now extracts from either HTTP encoding of B3 (Single Header or Multiple Header) based on what is contained in the header. Preference is given to Single Header encoding with Multiple Header being the fallback if Single Header is not found or is invalid. This behavior change is made to dynamically support all correctly encoded traces received instead of having to guess the expected encoding prior to receiving. (#882)
- Extend semantic conventions for RPC. (#900)
- To match constant naming conventions in the `api/standard` package, the `FaaS*` key names are appended with a suffix of `Key`. (#920)
  - `"api/standard".FaaSName` -> `FaaSNameKey`
  - `"api/standard".FaaSID` -> `FaaSIDKey`
  - `"api/standard".FaaSVersion` -> `FaaSVersionKey`
  - `"api/standard".FaaSInstance` -> `FaaSInstanceKey`

## Removed

- The `FlagsUnused` trace flag is removed. The purpose of this flag was to act as the inverse of `FlagsSampled`, the inverse of `FlagsSampled` is used instead. (#882)
- The B3 header constants ( `B3SingleHeader`, `B3DebugFlagHeader`, `B3TraceIDHeader`, `B3SpanIDHeader`, `B3SampledHeader`, `B3ParentSpanIDHeader` ) are removed. If B3 header keys are needed [the authoritative OpenZipkin package constants](#) should be used instead. (#882)

## Fixed

- The B3 Single Header name is now correctly `b3` instead of the previous `X-B3`. (#881)
- The B3 propagator now correctly supports sampling only values ( `b3: 0`, `b3: 1`, or `b3: d` ) for a Single B3 Header. (#882)
- The B3 propagator now propagates the debug flag. This removes the behavior of changing the debug flag into a set sampling bit. Instead, this now follow the B3 specification and omits the `X-B3-Sampling` header. (#882)
- The B3 propagator now tracks "unset" sampling state (meaning "defer the decision") and does not set the `X-B3-Sampling` header when injecting. (#882)
- Bump [github.com/itchyny/gojq](https://github.com/itchyny/gojq) from 0.10.3 to 0.10.4 in /tools. (#883)
- Bump [github.com/opentracing/opentracing-go](https://github.com/opentracing/opentracing-go) from v1.1.1-0.20190913142402-a7454ce5950e to v1.2.0. (#885)
- The tracing time conversion for OTLP spans is now correctly set to `UnixNano`. (#896)
- Ensure span status is not set to `Unknown` when no HTTP status code is provided as it is assumed to be `200 OK`. (#908)
- Ensure `httptrace.clientTracer` closes `http.headers` span. (#912)
- Prometheus exporter will not apply stale updates or forget inactive metrics. (#903)

- Add test for `api.standard HTTPClientAttributesFromHTTPRequest` . (#905)
- Bump `github.com/golangci/golangci-lint` from 1.27.0 to 1.28.1 in `/tools`. (#901, #913)
- Update `otel-collector` example to use the v0.5.0 collector. (#915)
- The `grpctrace` instrumentation uses a span name conforming to the OpenTelemetry semantic conventions (does not contain a leading slash ( `/` )). (#922)
- The `grpctrace` instrumentation includes an `rpc.method` attribute now set to the gRPC method name. (#900, #922)
- The `grpctrace` instrumentation `rpc.service` attribute now contains the package name if one exists. This is in accordance with OpenTelemetry semantic conventions. (#922)
- Correlation Context extractor will no longer insert an empty map into the returned context when no valid values are extracted. (#923)
- Bump `google.golang.org/api` from 0.28.0 to 0.29.0 in `/exporters/trace/jaeger`. (#925)
- Bump `github.com/itchyny/gojq` from 0.10.4 to 0.11.0 in `/tools`. (#926)
- Bump `github.com/golangci/golangci-lint` from 1.28.1 to 1.28.2 in `/tools`. (#930)

## 0.7.0 - 2020-06-26

This release implements the v0.5.0 version of the OpenTelemetry specification.

### Added

- The `othttp` instrumentation now includes default metrics. (#861)
- This CHANGELOG file to track all changes in the project going forward.
- Support for array type attributes. (#798)
- Apply transitive dependabot `go.mod` dependency updates as part of a new automatic Github workflow. (#844)
- Timestamps are now passed to exporters for each export. (#835)
- Add new `Accumulation` type to metric SDK to transport telemetry from `Accumulator` s to `Processor` s. This replaces the prior `Record` `struct` use for this purpose. (#835)
- New dependabot integration to automate package upgrades. (#814)
- `Meter` and `Tracer` implementations accept instrumentation version as an optional argument. This instrumentation version is passed on to exporters. (#811) (#805) (#802)
- The OTLP exporter includes the instrumentation version in telemetry it exports. (#811)
- Environment variables for Jaeger exporter are supported. (#796)
- New `aggregation.Kind` in the export metric API. (#808)
- New example that uses OTLP and the collector. (#790)
- Handle errors in the span `SetName` during span initialization. (#791)
- Default service config to enable retries for retry-able failed requests in the OTLP exporter and an option to override this default. (#777)
- New `go.opentelemetry.io/otel/api/oterror` package to uniformly support error handling and definitions for the project. (#778)
- New `global` default implementation of the `go.opentelemetry.io/otel/api/oterror.Handler` interface to be used to handle errors prior to an user defined `Handler` . There is also functionality for the user to register their `Handler` as well as a convenience function `Handle` to handle an error with this `global Handler` (#778)
- Options to specify propagators for `httptrace` and `grpctrace` instrumentation. (#784)
- The required `application/json` header for the Zipkin exporter is included in all exports. (#774)
- Integrate HTTP semantics helpers from the contrib repository into the `api/standard` package. #769

### Changed

- Rename `Integrator` to `Processor` in the metric SDK. (#863)
- Rename `AggregationSelector` to `AggregatorSelector`. (#859)
- Rename `SynchronizedCopy` to `SynchronizedMove`. (#858)
- Rename `simple integrator` to `basic integrator`. (#857)
- Merge otel collector examples. (#841)
- Change the metric SDK to support cumulative, delta, and pass-through exporters directly. With these changes, cumulative and delta specific exporters are able to request the correct kind of aggregation from the SDK. (#840)
- The `Aggregator.Checkpoint` API is renamed to `SynchronizedCopy` and adds an argument, a different `Aggregator` into which the copy is stored. (#812)
- The `export.Aggregator` contract is that `Update()` and `SynchronizedCopy()` are synchronized with each other. All the aggregation interfaces (`Sum`, `LastValue`, ...) are not meant to be synchronized, as the caller is expected to synchronize aggregators at a higher level after the `Accumulator`. Some of the `Aggregators` used unnecessary locking and that has been cleaned up. (#812)
- Use of `metric.Number` was replaced by `int64` now that we use `sync.Mutex` in the `MinMaxSumCount` and `Histogram` `Aggregators`. (#812)
- Replace `AlwaysParentSample` with `ParentSample(fallback)` to match the OpenTelemetry v0.5.0 specification. (#810)
- Rename `sdk/export/metric/aggregator` to `sdk/export/metric/aggregation`. #808
- Send configured headers with every request in the OTLP exporter, instead of just on connection creation. (#806)
- Update error handling for any one off error handlers, replacing, instead, with the `global.Handle` function. (#791)
- Rename `plugin` directory to `instrumentation` to match the OpenTelemetry specification. (#779)
- Makes the argument order to `Histogram` and `DDSketch` `New()` consistent. (#781)

## Removed

- `Uint64NumberKind` and related functions from the API. (#864)
- Context arguments from `Aggregator.Checkpoint` and `Integrator.Process` as they were unused. (#803)
- `SpanID` is no longer included in parameters for sampling decision to match the OpenTelemetry specification. (#775)

## Fixed

- Upgrade OTLP exporter to opentelemetry-proto matching the opentelemetry-collector v0.4.0 release. (#866)
- Allow changes to `go.sum` and `go.mod` when running dependabot tidy-up. (#871)
- Bump `github.com/stretchr/testify` from 1.4.0 to 1.6.1. (#824)
- Bump `github.com/prometheus/client_golang` from 1.7.0 to 1.7.1 in `/exporters/metric/prometheus`. (#867)
- Bump `google.golang.org/grpc` from 1.29.1 to 1.30.0 in `/exporters/trace/jaeger`. (#853)
- Bump `google.golang.org/grpc` from 1.29.1 to 1.30.0 in `/exporters/trace/zipkin`. (#854)
- Bumps `github.com/golang/protobuf` from 1.3.2 to 1.4.2 (#848)
- Bump `github.com/stretchr/testify` from 1.4.0 to 1.6.1 in `/exporters/otlp` (#817)
- Bump `github.com/golangci/golangci-lint` from 1.25.1 to 1.27.0 in `/tools` (#828)
- Bump `github.com/prometheus/client_golang` from 1.5.0 to 1.7.0 in `/exporters/metric/prometheus` (#838)
- Bump `github.com/stretchr/testify` from 1.4.0 to 1.6.1 in `/exporters/trace/jaeger` (#829)
- Bump `github.com/benbjohnson/clock` from 1.0.0 to 1.0.3 (#815)
- Bump `github.com/stretchr/testify` from 1.4.0 to 1.6.1 in `/exporters/trace/zipkin` (#823)
- Bump `github.com/itchyny/gojq` from 0.10.1 to 0.10.3 in `/tools` (#830)
- Bump `github.com/stretchr/testify` from 1.4.0 to 1.6.1 in `/exporters/metric/prometheus` (#822)

- Bump google.golang.org/grpc from 1.27.1 to 1.29.1 in /exporters/trace/zipkin (#820)
- Bump google.golang.org/grpc from 1.27.1 to 1.29.1 in /exporters/trace/jaeger (#831)
- Bump github.com/google/go-cmp from 0.4.0 to 0.5.0 (#836)
- Bump github.com/google/go-cmp from 0.4.0 to 0.5.0 in /exporters/trace/jaeger (#837)
- Bump github.com/google/go-cmp from 0.4.0 to 0.5.0 in /exporters/otlp (#839)
- Bump google.golang.org/api from 0.20.0 to 0.28.0 in /exporters/trace/jaeger (#843)
- Set span status from HTTP status code in the othttp instrumentation. (#832)
- Fixed typo in push controller comment. (#834)
- The `Aggregator` testing has been updated and cleaned. (#812)
- `metric.Number(0)` expressions are replaced by `0` where possible. (#812)
- Fixed `global handler_test.go` test failure. #804
- Fixed `BatchSpanProcessor.Shutdown` to wait until all spans are processed. (#766)
- Fixed OTLP example's accidental early close of exporter. (#807)
- Ensure zipkin exporter reads and closes response body. (#788)
- Update instrumentation to use `api/standard` keys instead of custom keys. (#782)
- Clean up tools and RELEASING documentation. (#762)

## 0.6.0 - 2020-05-21

### Added

- Support for `Resource`s in the prometheus exporter. (#757)
- New pull controller. (#751)
- New `UpDownSumObserver` instrument. (#750)
- OpenTelemetry collector demo. (#711)
- New `SumObserver` instrument. (#747)
- New `UpDownCounter` instrument. (#745)
- New timeout `Option` and configuration function `WithTimeout` to the push controller. (#742)
- New `api/standards` package to implement semantic conventions and standard key-value generation. (#731)

### Changed

- Rename `Register*` functions in the metric API to `New*` for all `Observer` instruments. (#761)
- Use `[]float64` for histogram boundaries, not `[]metric.Number`. (#758)
- Change OTLP example to use exporter as a trace `Syncer` instead of as an unneeded `Batcher`. (#756)
- Replace `WithResourceAttributes()` with `WithResource()` in the trace SDK. (#754)
- The prometheus exporter now uses the new pull controller. (#751)
- Rename `ScheduleDelayMillis` to `BatchTimeout` in the trace `BatchSpanProcessor`. (#752)
- Support use of synchronous instruments in asynchronous callbacks (#725)
- Move `Resource` from the `Export` method parameter into the metric export `Record`. (#739)
- Rename `Observer` instrument to `ValueObserver`. (#734)
- The push controller now has a method `(Provider())` to return a `metric.Provider` instead of the old `Meter` method that acted as a `metric.Provider`. (#738)
- Replace `Measure` instrument by `ValueRecorder` instrument. (#732)
- Rename correlation context header from `"Correlation-Context"` to `"otcorrelations"` to match the OpenTelemetry specification. 727)

### Fixed

- Ensure gRPC `ClientStream` override methods do not panic in `grpctrace` package. (#755)

- Disable parts of `BatchSpanProcessor` test until a fix is found. (#743)
- Fix `string` case in `kv Infer` function. (#746)
- Fix panic in `grpctrace` client interceptors. (#740)
- Refactor the `api/metrics` push controller and add `CheckpointSet` synchronization. (#737)
- Rewrite span batch process queue batching logic. (#719)
- Remove the push controller named Meter map. (#738)
- Fix Histogram aggregator initial state (fix #735). (#736)
- Ensure golang alpine image is running `golang-1.14` for examples. (#733)
- Added test for `grpctrace UnaryInterceptorClient`. (#695)
- Rearrange `api/metric` code layout. (#724)

## **0.5.0 - 2020-05-13**

### **Added**

- Batch `Observer` callback support. (#717)
- Alias `api` types to root package of project. (#696)
- Create basic `othttp.Transport` for simple client instrumentation. (#678)
- `SetAttribute(string, interface{})` to the trace API. (#674)
- Jaeger exporter option that allows user to specify custom http client. (#671)
- `Stringer` and `Infer` methods to `key` s. (#662)

### **Changed**

- Rename `NewKey` in the `kv` package to just `Key`. (#721)
- Move `core` and `key` to `kv` package. (#720)
- Make the metric API `Meter` a `struct` so the abstract `MeterImpl` can be passed and simplify implementation. (#709)
- Rename SDK `Batcher` to `Integrator` to match draft OpenTelemetry SDK specification. (#710)
- Rename SDK `Ungrouped` integrator to `simple.Integrator` to match draft OpenTelemetry SDK specification. (#710)
- Rename SDK `SDK` `struct` to `Accumulator` to match draft OpenTelemetry SDK specification. (#710)
- Move `Number` from `core` to `api/metric` package. (#706)
- Move `SpanContext` from `core` to `trace` package. (#692)
- Change traceparent header from `Traceparent` to `traceparent` to implement the W3C specification. (#681)

### **Fixed**

- Update tooling to run generators in all submodules. (#705)
- gRPC interceptor regexp to match methods without a service name. (#683)
- Use a `const` for padding 64-bit B3 trace IDs. (#701)
- Update `mockZipkin` listen address from `:0` to `127.0.0.1:0`. (#700)
- Left-pad 64-bit B3 trace IDs with zero. (#698)
- Propagate at least the first W3C tracestate header. (#694)
- Remove internal `StateLocker` implementation. (#688)
- Increase instance size CI system uses. (#690)
- Add a `key` benchmark and use reflection in `key.Infer()`. (#679)
- Fix internal `global` test by using `global.Meter` with `RecordBatch()`. (#680)
- Reimplement histogram using mutex instead of `StateLocker`. (#669)
- Switch `MinMaxSumCount` to a mutex lock implementation instead of `StateLocker`. (#667)

- Update documentation to not include any references to `WithKeys` . (#672)
- Correct misspelling. (#668)
- Fix clobbering of the span context if extraction fails. (#656)
- Bump `golangci-lint` and work around the corrupting bug. (#666) (#670)

## 0.4.3 - 2020-04-24

### Added

- `Dockerfile` and `docker-compose.yml` to run example code. (#635)
- New `grpctrace` package that provides gRPC client and server interceptors for both unary and stream connections. (#621)
- New `api/label` package, providing common label set implementation. (#651)
- Support for JSON marshaling of `Resources` . (#654)
- `TraceID` and `SpanID` implementations for `Stringer` interface. (#642)
- `RemoteAddrKey` in the othttp plugin to include the HTTP client address in top-level spans. (#627)
- `WithSpanFormatter` option to the othttp plugin. (#617)
- Updated README to include section for compatible libraries and include reference to the contrib repository. (#612)
- The prometheus exporter now supports exporting histograms. (#601)
- A `String` method to the `Resource` to return a hashable identifier for a now unique resource. (#613)
- An `Iter` method to the `Resource` to return an array `AttributeIterator` . (#613)
- An `Equal` method to the `Resource` test the equivalence of resources. (#613)
- An iterable structure ( `AttributeIterator` ) for `Resource` attributes.

### Changed

- zipkin export's `NewExporter` now requires a `serviceName` argument to ensure this needed values is provided. (#644)
- Pass `Resources` through the metrics export pipeline. (#659)

### Removed

- `WithKeys` option from the metric API. (#639)

### Fixed

- Use the `label.Set.Equivalent` value instead of an encoding in the batcher. (#658)
- Correct typo `trace.Exporter` to `trace.SpanSyncer` in comments. (#653)
- Use type names for return values in jaeger exporter. (#648)
- Increase the visibility of the `api/key` package by updating comments and fixing usages locally. (#650)
- `Checkpoint` only after `Update` ; Keep records in the `sync.Map` longer. (#647)
- Do not cache `reflect.ValueOf()` in metric Labels. (#649)
- Batch metrics exported from the OTLP exporter based on `Resource` and labels. (#626)
- Add error wrapping to the prometheus exporter. (#631)
- Update the OTLP exporter batching of traces to use a unique `string` representation of an associated `Resource` as the batching key. (#623)
- Update OTLP `SpanData` transform to only include the `ParentSpanID` if one exists. (#614)
- Update `Resource` internal representation to uniquely and reliably identify resources. (#613)
- Check return value from `CheckpointSet.ForEach` in prometheus exporter. (#622)
- Ensure spans created by httptrace client tracer reflect operation structure. (#618)

- Create a new recorder rather than reuse when multiple observations in same epoch for asynchronous instruments. (#610)
- The default port the OTLP exporter uses to connect to the OpenTelemetry collector is updated to match the one the collector listens on by default. (#611)

## [0.4.2](#) - 2020-03-31

### Fixed

- Fix `pre_release.sh` to update version in `sdk/opentelemetry.go`. (#607)
- Fix time conversion from internal to OTLP in OTLP exporter. (#606)

## [0.4.1](#) - 2020-03-31

### Fixed

- Update `tag.sh` to create signed tags. (#604)

## [0.4.0](#) - 2020-03-30

### Added

- New API package `api/metric/registry` that exposes a `MeterImpl` wrapper for use by SDKs to generate unique instruments. (#580)
- Script to verify examples after a new release. (#579)

### Removed

- The dogstatsd exporter due to lack of support. This additionally removes support for statsd. (#591)
- `LabelSet` from the metric API. This is replaced by a `[]core.KeyValue` slice. (#595)
- `Labels` from the metric API's `Meter` interface. (#595)

### Changed

- The metric `export.Labels` became an interface which the SDK implements and the `export` package provides a simple, immutable implementation of this interface intended for testing purposes. (#574)
- Renamed `internal/metric.Meter` to `MeterImpl`. (#580)
- Renamed `api/global/internal.obsImpl` to `asyncImpl`. (#580)

### Fixed

- Corrected missing return in mock span. (#582)
- Update License header for all source files to match CNCF guidelines and include a test to ensure it is present. (#586) (#596)
- Update to v0.3.0 of the OTLP in the OTLP exporter. (#588)
- Update pre-release script to be compatible between GNU and BSD based systems. (#592)
- Add a `RecordBatch` benchmark. (#594)
- Moved span transforms of the OTLP exporter to the internal package. (#593)
- Build both go-1.13 and go-1.14 in circleci to test for all supported versions of Go. (#569)
- Removed unneeded allocation on empty labels in OTLP exporter. (#597)
- Update `BatchedSpanProcessor` to process the queue until no data but respect max batch size. (#599)
- Update project documentation godoc.org links to pkg.go.dev. (#602)

## [0.3.0](#) - 2020-03-21



This is a first official beta release, which provides almost fully complete metrics, tracing, and context propagation functionality. There is still a possibility of breaking changes.

## Added

- Add `Observer` metric instrument. (#474)
- Add global `Propagators` functionality to enable deferred initialization for propagators registered before the first Meter SDK is installed. (#494)
- Simplified export setup pipeline for the jaeger exporter to match other exporters. (#459)
- The zipkin trace exporter. (#495)
- The OTLP exporter to export metric and trace telemetry to the OpenTelemetry collector. (#497) (#544) (#545)
- The `StatusMessage` field was added to the trace `Span`. (#524)
- Context propagation in OpenTracing bridge in terms of OpenTelemetry context propagation. (#525)
- The `Resource` type was added to the SDK. (#528)
- The global API now supports a `Tracer` and `Meter` function as shortcuts to getting a global `*Provider` and calling these methods directly. (#538)
- The metric API now defines a generic `MeterImpl` interface to support general purpose `Meter` construction. Additionally, `SyncImpl` and `AsyncImpl` are added to support general purpose instrument construction. (#560)
- A metric `Kind` is added to represent the `MeasureKind`, `ObserverKind`, and `CounterKind`. (#560)
- Scripts to better automate the release process. (#576)

## Changed

- Default to to use `AlwaysSampler` instead of `ProbabilitySampler` to match OpenTelemetry specification. (#506)
- Renamed `AlwaysSampleSampler` to `AlwaysOnSampler` in the trace API. (#511)
- Renamed `NeverSampleSampler` to `AlwaysOffSampler` in the trace API. (#511)
- The `Status` field of the `Span` was changed to `StatusCode` to disambiguate with the added `StatusMessage`. (#524)
- Updated the trace `Sampler` interface conform to the OpenTelemetry specification. (#531)
- Rename metric API `Options` to `Config`. (#541)
- Rename metric `Counter` aggregator to be `Sum`. (#541)
- Unify metric options into `Option` from instrument specific options. (#541)
- The trace API's `TraceProvider` now support `Resource`s. (#545)
- Correct error in zipkin module name. (#548)
- The jaeger trace exporter now supports `Resource`s. (#551)
- Metric SDK now supports `Resource`s. The `WithResource` option was added to configure a `Resource` on creation and the `Resource` method was added to the metric `Descriptor` to return the associated `Resource`. (#552)
- Replace `ErrNoLastValue` and `ErrEmptyDataSet` by `ErrNoData` in the metric SDK. (#557)
- The stdout trace exporter now supports `Resource`s. (#558)
- The metric `Descriptor` is now included at the API instead of the SDK. (#560)
- Replace `Ordered` with an iterator in `export.Labels`. (#567)

## Removed

- The vendor specific Stackdriver. It is now hosted on 3rd party vendor infrastructure. (#452)
- The `Unregister` method for metric observers as it is not in the OpenTelemetry specification. (#560)

- `GetDescriptor` from the metric SDK. (#575)
- The `Gauge` instrument from the metric API. (#537)

## Fixed

- Make histogram aggregator checkpoint consistent. (#438)
- Update README with import instructions and how to build and test. (#505)
- The default label encoding was updated to be unique. (#508)
- Use `NewRoot` in the othttp plugin for public endpoints. (#513)
- Fix data race in `BatchedSpanProcessor`. (#518)
- Skip test-386 for Mac OS 10.15.x (Catalina and upwards). #521
- Use a variable-size array to represent ordered labels in maps. (#523)
- Update the OTLP protobuf and update changed import path. (#532)
- Use `StateLocker` implementation in `MinMaxSumCount`. (#546)
- Eliminate goroutine leak in histogram stress test. (#547)
- Update OTLP exporter with latest protobuf. (#550)
- Add filters to the othttp plugin. (#556)
- Provide an implementation of the `Header*` filters that do not depend on Go 1.14. (#565)
- Encode labels once during checkpoint. The checkpoint function is executed in a single thread so we can do the encoding lazily before passing the encoded version of labels to the exporter. This is a cheap and quick way to avoid encoding the labels on every collection interval. (#572)
- Run coverage over all packages in `COVERAGE_MOD_DIR`. (#573)

## 0.2.3 - 2020-03-04

### Added

- `RecordError` method on `Span` in the trace API to Simplify adding error events to spans. (#473)
- Configurable push frequency for exporters setup pipeline. (#504)

### Changed

- Rename the `exporter` directory to `exporters`. The `go.opentelemetry.io/otel/exporter/trace/jaeger` package was mistakenly released with a `v1.0.0` tag instead of `v0.1.0`. This resulted in all subsequent releases not becoming the default latest. A consequence of this was that all `go get`s pulled in the incompatible `v0.1.0` release of that package when pulling in more recent packages from other otel packages. Renaming the `exporter` directory to `exporters` fixes this issue by renaming the package and therefore clearing any existing dependency tags. Consequentially, this action also renames *all* exporter packages. (#502)

### Removed

- The `CorrelationContextHeader` constant in the `correlation` package is no longer exported. (#503)

## 0.2.2 - 2020-02-27

### Added

- `HTTPSupplier` interface in the propagation API to specify methods to retrieve and store a single value for a key to be associated with a carrier. (#467)
- `HTTPExtractor` interface in the propagation API to extract information from an `HTTPSupplier` into a context. (#467)

- `HTTPInjector` interface in the propagation API to inject information into an `HTTPSupplier`. (#467)
- `Config` and configuring `Option` to the propagator API. (#467)
- `Propagators` interface in the propagation API to contain the set of injectors and extractors for all supported carrier formats. (#467)
- `HTTPPropagator` interface in the propagation API to inject and extract from an `HTTPSupplier`. (#467)
- `WithInjectors` and `WithExtractors` functions to the propagator API to configure injectors and extractors to use. (#467)
- `ExtractHTTP` and `InjectHTTP` functions to apply configured HTTP extractors and injectors to a passed context. (#467)
- Histogram aggregator. (#433)
- `DefaultPropagator` function and have it return `trace.TraceContext` as the default context propagator. (#456)
- `AlwaysParentSample` sampler to the trace API. (#455)
- `WithNewRoot` option function to the trace API to specify the created span should be considered a root span. (#451)

## Changed

- Renamed `WithMap` to `ContextWithMap` in the correlation package. (#481)
- Renamed `FromContext` to `MapFromContext` in the correlation package. (#481)
- Move correlation context propagation to correlation package. (#479)
- Do not default to putting remote span context into links. (#480)
- Propagators extrac
- `Tracer.WithSpan` updated to accept `StartOptions`. (#472)
- Renamed `MetricKind` to `Kind` to not stutter in the type usage. (#432)
- Renamed the `export` package to `metric` to match directory structure. (#432)
- Rename the `api/distributedcontext` package to `api/correlation`. (#444)
- Rename the `api/propagators` package to `api/propagation`. (#444)
- Move the propagators from the `propagators` package into the `trace` API package. (#444)
- Update `Float64Gauge`, `Int64Gauge`, `Float64Counter`, `Int64Counter`, `Float64Measure`, and `Int64Measure` metric methods to use value receivers instead of pointers. (#462)
- Moved all dependencies of tools package to a tools directory. (#466)

## Removed

- Binary propagators. (#467)
- NOOP propagator. (#467)

## Fixed

- Upgraded `github.com/golangci/golangci-lint` from `v1.21.0` to `v1.23.6` in `tools/`. (#492)
- Fix a possible nil-dereference crash (#478)
- Correct comments for `InstallNewPipeline` in the stdout exporter. (#483)
- Correct comments for `InstallNewPipeline` in the dogstatsd exporter. (#484)
- Correct comments for `InstallNewPipeline` in the prometheus exporter. (#482)
- Initialize `onError` based on `Config` in prometheus exporter. (#486)
- Correct module name in prometheus exporter README. (#475)
- Removed tracer name prefix from span names. (#430)
- Fix `aggregator_test.go` import package comment. (#431)

- Improved detail in stdout exporter. (#436)
- Fix a dependency issue (generate target should depend on stringer, not lint target) in Makefile. (#442)
- Reorders the Makefile targets within `precommit` target so we generate files and build the code before doing linting, so we can get much nicer errors about syntax errors from the compiler. (#442)
- Reword function documentation in gRPC plugin. (#446)
- Send the `span.kind` tag to Jaeger from the jaeger exporter. (#441)
- Fix `metadataSupplier` in the jaeger exporter to overwrite the header if existing instead of appending to it. (#441)
- Upgraded to Go 1.13 in CI. (#465)
- Correct opentelemetry.io URL in trace SDK documentation. (#464)
- Refactored reference counting logic in SDK determination of stale records. (#468)
- Add call to `runtime.Gosched` in instrument `acquireHandle` logic to not block the collector. (#469)

### [0.2.1.1](#) - 2020-01-13

#### Fixed

- Use stateful batcher on Prometheus exporter fixing regression introduced in #395. (#428)

### [0.2.1](#) - 2020-01-08

#### Added

- Global meter forwarding implementation. This enables deferred initialization for metric instruments registered before the first Meter SDK is installed. (#392)
- Global trace forwarding implementation. This enables deferred initialization for tracers registered before the first Trace SDK is installed. (#406)
- Standardize export pipeline creation in all exporters. (#395)
- A testing, organization, and comments for 64-bit field alignment. (#418)
- Script to tag all modules in the project. (#414)

#### Changed

- Renamed `propagation` package to `propagators`. (#362)
- Renamed `B3Propagator` `propagator` to `B3`. (#362)
- Renamed `TextFormatPropagator` `propagator` to `TextFormat`. (#362)
- Renamed `BinaryPropagator` `propagator` to `Binary`. (#362)
- Renamed `BinaryFormatPropagator` `propagator` to `BinaryFormat`. (#362)
- Renamed `NoopTextFormatPropagator` `propagator` to `NoopTextFormat`. (#362)
- Renamed `TraceContextPropagator` `propagator` to `TraceContext`. (#362)
- Renamed `SpanOption` to `StartOption` in the trace API. (#369)
- Renamed `StartOptions` to `StartConfig` in the trace API. (#369)
- Renamed `EndOptions` to `EndConfig` in the trace API. (#369)
- `Number` now has a pointer receiver for its methods. (#375)
- Renamed `CurrentSpan` to `SpanFromContext` in the trace API. (#379)
- Renamed `SetCurrentSpan` to `ContextWithSpan` in the trace API. (#379)
- Renamed `Message` in Event to `Name` in the trace API. (#389)
- Prometheus exporter no longer aggregates metrics, instead it only exports them. (#385)
- Renamed `HandleImpl` to `BoundInstrumentImpl` in the metric API. (#400)
- Renamed `Float64CounterHandle` to `Float64CounterBoundInstrument` in the metric API. (#400)
- Renamed `Int64CounterHandle` to `Int64CounterBoundInstrument` in the metric API. (#400)

- Renamed `Float64GaugeHandle` to `Float64GaugeBoundInstrument` in the metric API. (#400)
- Renamed `Int64GaugeHandle` to `Int64GaugeBoundInstrument` in the metric API. (#400)
- Renamed `Float64MeasureHandle` to `Float64MeasureBoundInstrument` in the metric API. (#400)
- Renamed `Int64MeasureHandle` to `Int64MeasureBoundInstrument` in the metric API. (#400)
- Renamed `Release` method for bound instruments in the metric API to `Unbind`. (#400)
- Renamed `AcquireHandle` method for bound instruments in the metric API to `Bind`. (#400)
- Renamed the `File` option in the stdout exporter to `Writer`. (#404)
- Renamed all `Options` to `Config` for all metric exports where this wasn't already the case.

## Fixed

- Aggregator import path corrected. (#421)
- Correct links in README. (#368)
- The README was updated to match latest code changes in its examples. (#374)
- Don't capitalize error statements. (#375)
- Fix ignored errors. (#375)
- Fix ambiguous variable naming. (#375)
- Removed unnecessary type casting. (#375)
- Use named parameters. (#375)
- Updated release schedule. (#378)
- Correct http-stackdriver example module name. (#394)
- Removed the `http.request` span in `httptrace` package. (#397)
- Add comments in the metrics SDK (#399)
- Initialize checkpoint when creating ddsketch aggregator to prevent panic when merging into a empty one. (#402) (#403)
- Add documentation of compatible exporters in the README. (#405)
- Typo fix. (#408)
- Simplify span check logic in SDK tracer implementation. (#419)

## 0.2.0 - 2019-12-03

### Added

- Unary gRPC tracing example. (#351)
- Prometheus exporter. (#334)
- Dogstatsd metrics exporter. (#326)

### Changed

- Rename `MaxSumCount` aggregation to `MinMaxSumCount` and add the `Min` interface for this aggregation. (#352)
- Rename `GetMeter` to `Meter`. (#357)
- Rename `HTTPTraceContextPropagator` to `TraceContextPropagator`. (#355)
- Rename `HTTPB3Propagator` to `B3Propagator`. (#355)
- Rename `HTTPTraceContextPropagator` to `TraceContextPropagator`. (#355)
- Move `/global` package to `/api/global`. (#356)
- Rename `GetTracer` to `Tracer`. (#347)

### Removed

- `SetAttribute` from the `Span` interface in the trace API. (#361)
- `AddLink` from the `Span` interface in the trace API. (#349)

- `Link` from the `Span` interface in the trace API. (#349)

### Fixed

- Exclude example directories from coverage report. (#365)
- Lint make target now implements automatic fixes with `golangci-lint` before a second run to report the remaining issues. (#360)
- Drop `GO111MODULE` environment variable in Makefile as Go 1.13 is the project specified minimum version and this environment variable is not needed for that version of Go. (#359)
- Run the race checker for all test. (#354)
- Redundant commands in the Makefile are removed. (#354)
- Split the `generate` and `lint` targets of the Makefile. (#354)
- Renames `circle-ci` target to more generic `ci` in Makefile. (#354)
- Add example Prometheus binary to gitignore. (#358)
- Support negative numbers with the `MaxSumCount`. (#335)
- Resolve race conditions in `push_test.go` identified in #339. (#340)
- Use `/usr/bin/env bash` as a shebang in scripts rather than `/bin/bash`. (#336)
- Trace benchmark now tests both `AlwaysSample` and `NeverSample`. Previously it was testing `AlwaysSample` twice. (#325)
- Trace benchmark now uses a `[]byte` for `TraceID` to fix failing test. (#325)
- Added a trace benchmark to test variadic functions in `setAttribute` vs `setAttributes`. (#325)
- The `defaultkeys` batcher was only using the encoded label set as its map key while building a checkpoint. This allowed distinct label sets through, but any metrics sharing a label set could be overwritten or merged incorrectly. This was corrected. (#333)

## 0.1.2 - 2019-11-18

### Fixed

- Optimized the `simplelru` map for attributes to reduce the number of allocations. (#328)
- Removed unnecessary unslicing of parameters that are already a slice. (#324)

## 0.1.1 - 2019-11-18

This release contains a Metrics SDK with stdout exporter and supports basic aggregations such as counter, gauges, array, maxsumcount, and ddskeetch.

### Added

- Metrics stdout export pipeline. (#265)
- Array aggregation for raw measure metrics. (#282)
- The `core.Value` now have a `MarshalJSON` method. (#281)

### Removed

- `WithService`, `WithResources`, and `WithComponent` methods of tracers. (#314)
- Prefix slash in `Tracer.Start()` for the Jaeger example. (#292)

### Changed

- Allocation in `LabelSet` construction to reduce GC overhead. (#318)
- `trace.WithAttributes` to append values instead of replacing (#315)
- Use a formula for tolerance in sampling tests. (#298)
- Move export types into trace and metric-specific sub-directories. (#289)

- `SpanKind` back to being based on an `int` type. (#288)

### Fixed

- URL to OpenTelemetry website in README. (#323)
- Name of othttp default tracer. (#321)
- `ExportSpans` for the stackdriver exporter now handles `nil` context. (#294)
- CI modules cache to correctly restore/save from/to the cache. (#316)
- Fix metric SDK race condition between `LoadOrStore` and the assignment `rec.recorder = i.meter.exporter.AggregatorFor(rec)`. (#293)
- README now reflects the new code structure introduced with these changes. (#291)
- Make the basic example work. (#279)

## **0.1.0 - 2019-11-04**

This is the first release of open-telemetry go library. It contains api and sdk for trace and meter.

### Added

- Initial OpenTelemetry trace and metric API prototypes.
- Initial OpenTelemetry trace, metric, and export SDK packages.
- A wireframe bridge to support compatibility with OpenTracing.
- Example code for a basic, http-stackdriver, http, jaeger, and named tracer setup.
- Exporters for Jaeger, Stackdriver, and stdout.
- Propagators for binary, B3, and trace-context protocols.
- Project information and guidelines in the form of a README and CONTRIBUTING.
- Tools to build the project and a Makefile to automate the process.
- Apache-2.0 license.
- CircleCI build CI manifest files.
- CODEOWNERS file to track owners of this project.