

Programming Gin Monitoring Service

Web framework monitoring

Jungmin Choi / Hanwha Vision



Introduction

- DevOps Engineer
- Programming for stable service
- Interested in Monitoring
- Beginner Gopher



cjm104174@gmail.com



Index

- What's Gin
- Need for Gin Monitoring
- How to Build Gin Monitoring Service
- Demo
- Q&A





What's Gin



What's Gin

what's gin



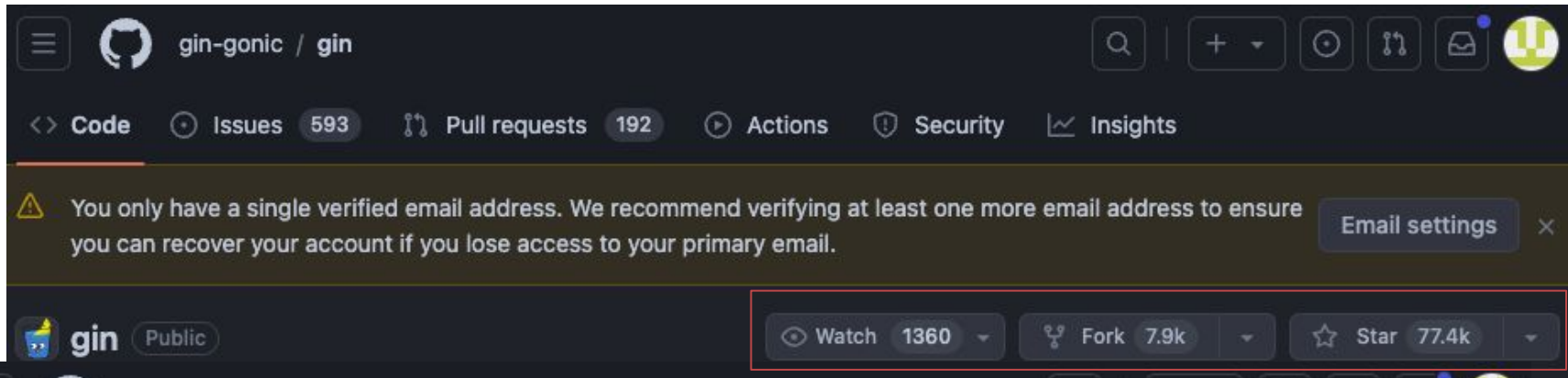
Golang WebFramework

- Popular
- Performance

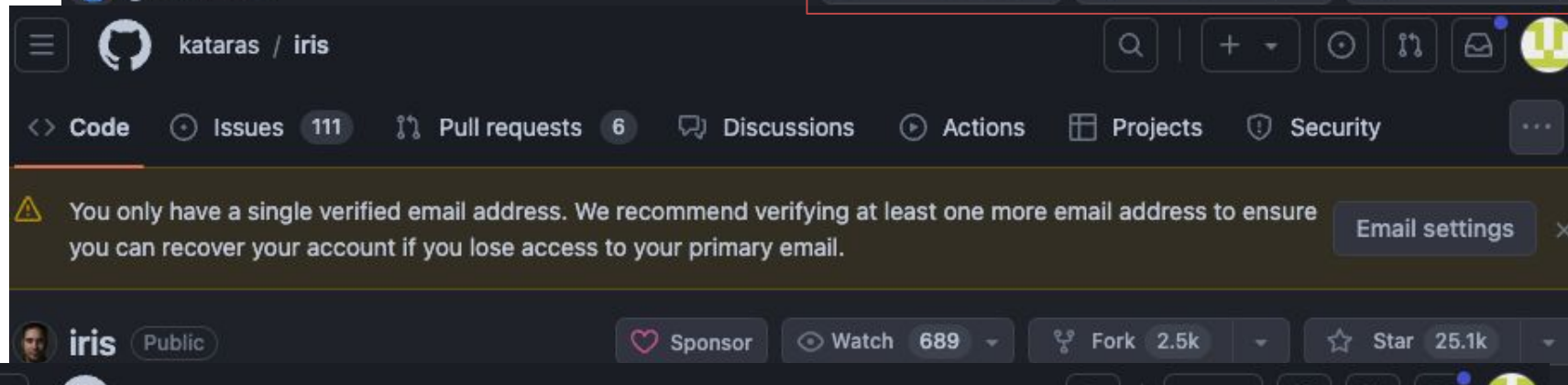


Most Popular Webframework (Golang)

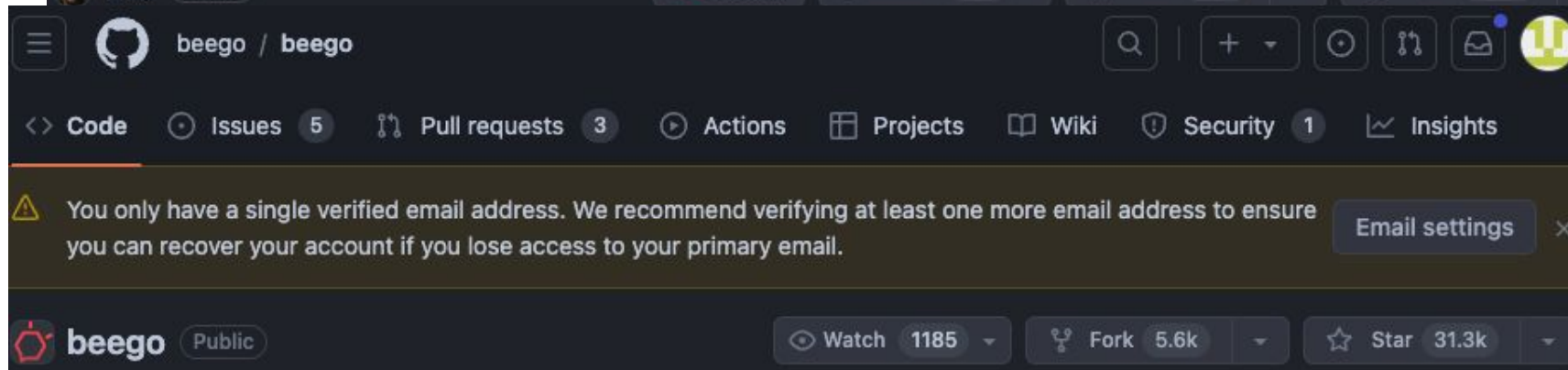
Gin



Iris



Beego





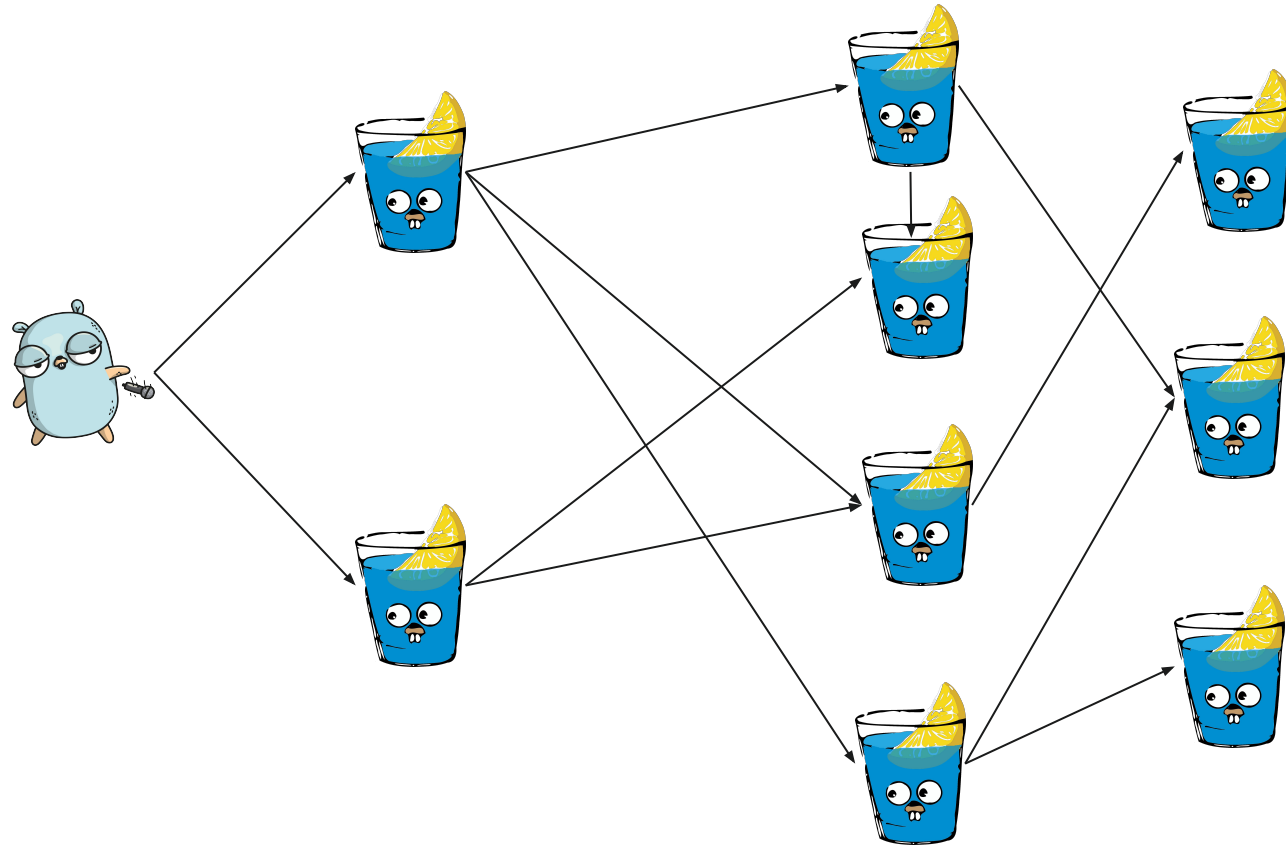
Need for Gin Monitoring



Web Analytics

```
{
  "GET/metrics": {
    "avg_duration": 0.0007532495,
    "count": 2,
    "status_codes": {
      "200": 2
    }
  },
  "GET/ping": {
    "avg_duration": 0.00016386133333333333,
    "count": 3,
    "status_codes": {
      "200": 3
    }
  },
  "GETunknown": {
    "avg_duration": 0.0000719375,
    "count": 2,
    "status_codes": {
      "404": 2
    }
  }
}
```





Complex communication structures



How to Build Gin Monitoring Service



Gin Monitoring

- Metric Monitoring
- Distributed Trace Monitoring
- Log Monitoring



How to Build Gin Monitoring Service

Gin Middleware




Client Request —————> Gin Middleware —————> Server Response

- Logging
- Auth
- Monitoring
- etc...



How to Build Gin Monitoring Service

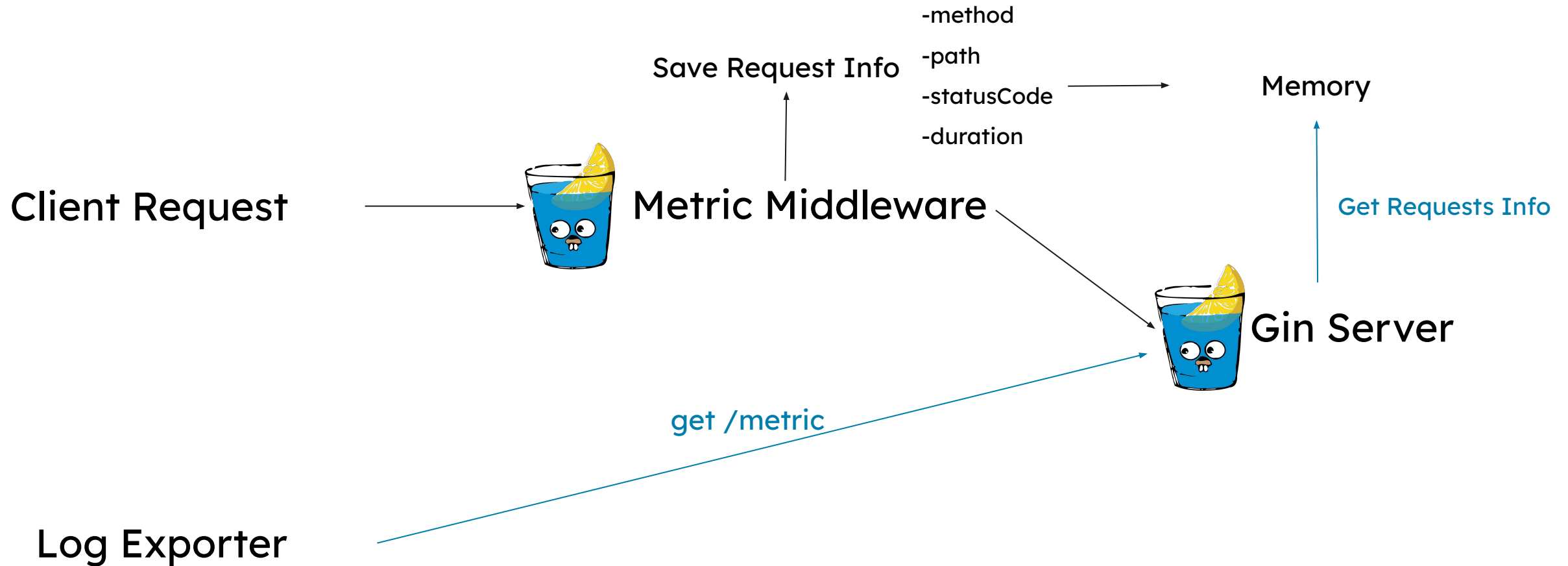
Metric Monitoring

- 
- Request Duration
 - Request Count
 - Status Code



How to Build Gin Monitoring Service

Metric Monitoring



```
package metrics

import (
    "sync"
    "time"
    "github.com/gin-gonic/gin"
)

func MetricsMiddleware() gin.HandlerFunc {
    return func(c *gin.Context) {
        start := time.Now()
        c.Next()
        duration := time.Since(start)
        status := c.Writer.Status()
        endpoint := c.FullPath()
        if endpoint == "" {
            endpoint = "/unknown"
        }
        recordMetrics(c.Request.Method, endpoint, duration, status)
    }
}
```



```
var (  
    requestCount      = make(map[string]int)  
    requestDuration    = make(map[string]time.Duration)  
    requestStatusCodes = make(map[string]map[int]int)  
    mu                 sync.Mutex  
)  
  
func recordMetrics(method, endpoint string, duration time.Duration, status int) {  
    mu.Lock()  
    defer mu.Unlock()  
  
    requestCount[method+endpoint]++  
    requestDuration[method+endpoint] += duration  
  
    if _, exists := requestStatusCodes[method+endpoint]; !exists {  
        requestStatusCodes[method+endpoint] = make(map[int]int)  
    }  
  
    requestStatusCodes[method+endpoint][status]++  
}
```




```
func MetricsHandler(c *gin.Context) {  
    mu.Lock()  
    defer mu.Unlock()  
  
    metricsData := make(map[string]interface{})  
    for endpoint, count := range requestCount {  
        avgDuration := requestDuration[endpoint].Seconds() / float64(count)  
        metricsData[endpoint] = map[string]interface{}{  
            "count":          count,  
            "avg_duration":   avgDuration,  
            "status_codes": requestStatusCodes[endpoint],  
        }  
    }  
    c.JSON(200, metricsData)  
}
```



```
package main

import (
    "github.com/MAYFCHOI/gin-monitor/metrics"
    "github.com/gin-gonic/gin"
)

func main() {
    r := gin.New()
    r.Use(metrics.MetricsMiddleware())
    r.GET("/metrics", metrics.MetricsHandler)
    r.Run()
}
```

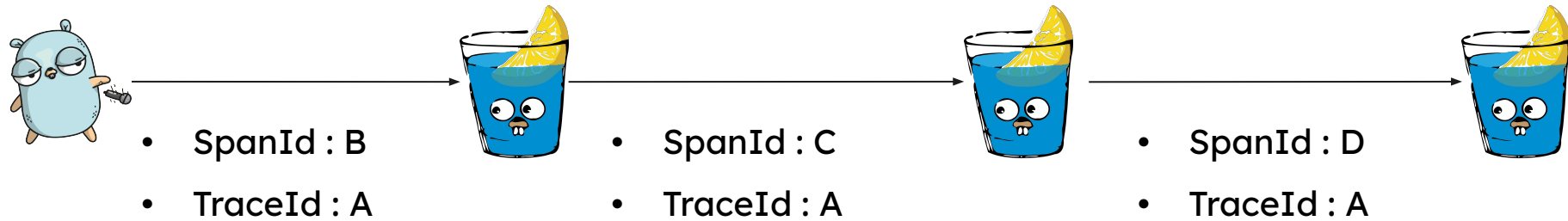


```
{
  "metrics": {
    "GET/metrics": {
      "avg_duration": 1.17757009345794,
      "count": 107,
      "status_codes": {
        "200": 107
      }
    },
    "GET/process": {
      "avg_duration": 2.625,
      "count": 40,
      "status_codes": {
        "200": 40
      }
    },
    "GET/unknown": {
      "avg_duration": 0,
      "count": 1,
      "status_codes": {
        "404": 1
      }
    }
  },
  "service_name": "test-server1"
}
```

How to Build Gin Monitoring Service

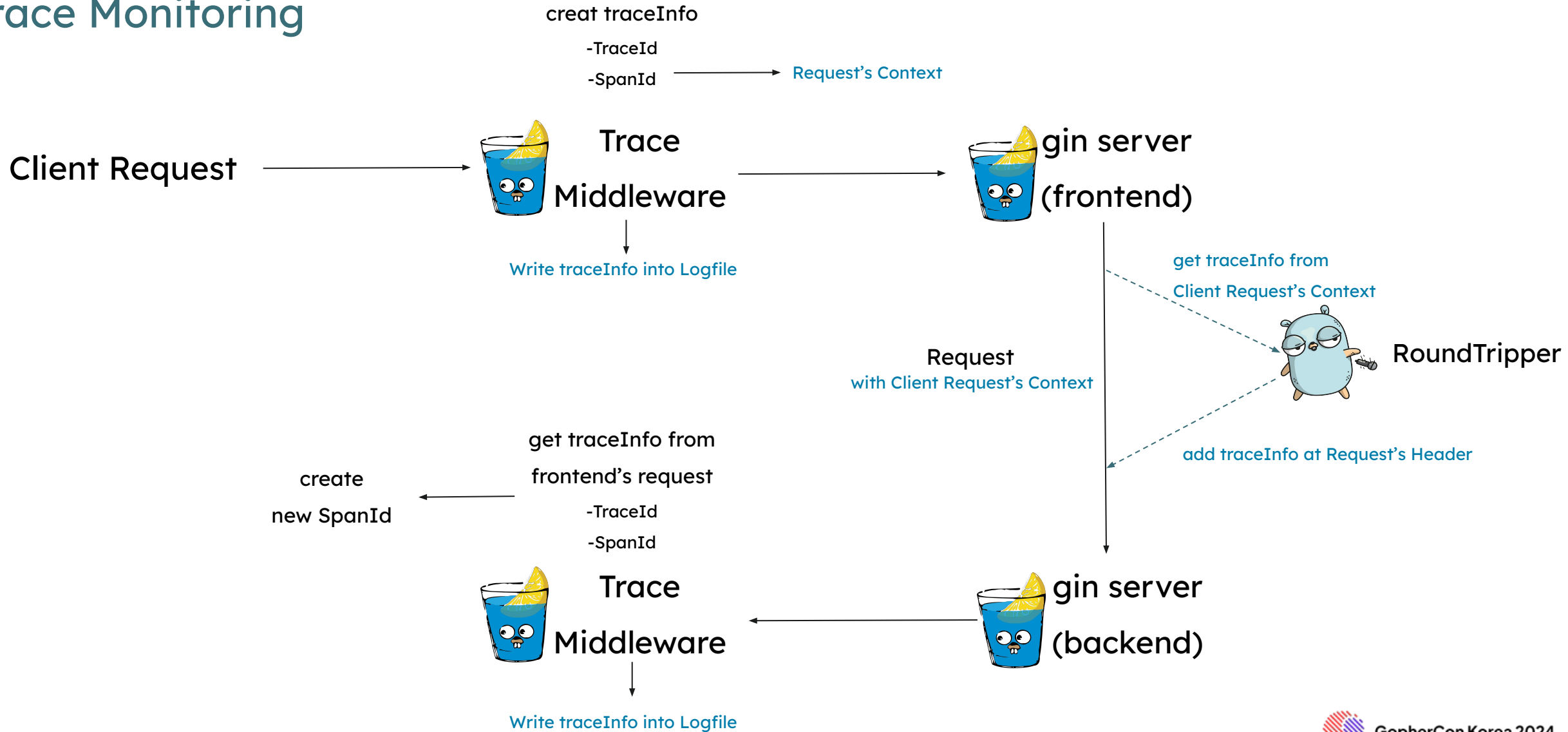
Distributed Trace Monitoring

- SpanId
- TraceId



How to Build Gin Monitoring Service

Trace Monitoring



```

type traceContextKey struct{}

type Span struct {
    TraceID    string
    SpanID     string
    ParentSpan *Span
}

func NewSpan(traceID string, parent *Span) *Span {
    return &Span{
        TraceID:    traceID,
        SpanID:     uuid.New().String(),
        ParentSpan: parent,
    }
}

func FromContext(ctx context.Context) *Span {
    span, _ := ctx.Value(traceContextKey{}).(*Span)
    return span
}

func NewContext(ctx context.Context, span *Span) context.Context {
    return context.WithValue(ctx, traceContextKey{}, span)
}

```



```

func TracingMiddleware() gin.HandlerFunc {
    return func(c *gin.Context) {
        traceID := c.Request.Header.Get("X-Trace-ID")
        spanID := c.Request.Header.Get("X-Span-ID")

        var span *Span
        var parentSpan *Span

        if traceID != "" && spanID != "" {
            parentSpan = &Span{TraceID: traceID, SpanID: spanID}
            span = NewSpan(traceID, parentSpan)
        } else {
            span = NewSpan(uuid.New().String(), nil)
            c.Request.Header.Set("X-Trace-ID", span.TraceID)
            c.Request.Header.Set("X-Span-ID", span.SpanID)
        }

        ctx := NewContext(c.Request.Context(), span)
        c.Request = c.Request.WithContext(ctx)

        start := time.Now()
        c.Next()
        duration := time.Since(start)

        log.Printf("TraceID: %s, SpanID: %s, ParentSpanID: %s, Method: %s, Path: %s, Duration: %s, Status: %d",
            span.TraceID, span.SpanID, parentSpanID(span), c.Request.Method, c.Request.URL.Path, duration.String(), c.Writer.Status())

        c.Writer.Header().Set("X-Trace-ID", span.TraceID)
        c.Writer.Header().Set("X-Span-ID", span.SpanID)
    }
}

```



```
type TracingTransport struct {
    Transport http.RoundTripper
}

func NewTracingTransport(transport http.RoundTripper) *TracingTransport {
    return &TracingTransport{
        Transport: transport,
    }
}

func (t *TracingTransport) RoundTrip(req *http.Request) (*http.Response, error) {
    ctx := req.Context()
    span := FromContext(ctx)
    if span != nil {
        req.Header.Set("X-Trace-ID", span.TraceID)
        req.Header.Set("X-Span-ID", span.SpanID)
    }
    return t.Transport.RoundTrip(req)
}
```




```
package main

import (
    "net/http"
    "github.com/MAYFCHOI/gin-monitoring/tracing"
    "github.com/gin-gonic/gin"
)

func main() {
    r := gin.New()
    r.Use(tracing.TracingMiddleware())
    client := &http.Client{
        Transport: tracing.NewTracingTransport(http.DefaultTransport),
    }
    r.GET("/process", processRequest(client))
    r.Run()
}
```



```
func processRequest(client *http.Client) gin.HandlerFunc {
    return func(c *gin.Context) {
        req, err := http.NewRequestWithContext(c.Request.Context(), "GET", "http://backend:8086/api",
nil)

        if err != nil {
            c.JSON(http.StatusInternalServerError, gin.H{"error": "Internal error"})
            return
        }
        resp, err := client.Do(req)
        if err != nil {
            c.JSON(http.StatusInternalServerError, gin.H{"error": "Backend service call failed"})
            return
        }
        defer resp.Body.Close()
        c.JSON(http.StatusOK, gin.H{"message": "Request processed successfully"})
    }
}
```



test-server.log M X

logs > test-server.log

```
1 2024/10/07 05:24:19 Service: test-server1, TraceID: 8678e9fe-fec5-4a2f-9db1-9906b06c4dc8, SpanID: d87e8bb6-9fb5-4422-98e4-56413d4a4faf, ParentSpanID: , Method: GET, Path: /process, Duration: 4.7664ms,
2 2024/10/07 05:24:25 Service: test-server1, TraceID: 2b6cb680-28e6-438e-bca6-5e1fde98daaf, SpanID: 5b12f79d-07b9-48d3-966a-21e61c713ee6, ParentSpanID: , Method: GET, Path: /process, Duration: 3.9809ms,
3 2024/10/07 05:24:26 Service: test-server1, TraceID: 9bec813e-024b-48c0-8f62-2be8e29b03a1, SpanID: f5b3768e-b178-4e7b-b0b4-88ff9cf99a60, ParentSpanID: , Method: GET, Path: /process, Duration: 4.1867ms,
4 2024/10/07 05:24:26 Service: test-server1, TraceID: 3fac5134-fcc1-42b7-a00c-ec86e4ebd19c, SpanID: 466c765f-812c-4dd6-b191-750eb1cd7a37, ParentSpanID: , Method: GET, Path: /process, Duration: 1.8531ms,
5 2024/10/07 05:24:26 Service: test-server1, TraceID: 8d7dcf15-f21f-493b-addf-a6db3710d931, SpanID: 47dae10d-bb14-4ebb-9374-882927896827, ParentSpanID: , Method: GET, Path: /process, Duration: 1.7948ms,
6 2024/10/07 05:24:27 Service: test-server1, TraceID: 631c2bbd-e3be-45eb-85af-eb99a524fae2, SpanID: 3cb7f240-e75a-4bfd-a437-c404e2145c12, ParentSpanID: , Method: GET, Path: /process, Duration: 2.1156ms,
7 2024/10/07 05:24:27 Service: test-server1, TraceID: 9f09a6ff-2d14-40de-9a4e-9287bc677325, SpanID: d8439b72-4864-4f36-98f8-bca3fff41997, ParentSpanID: , Method: GET, Path: /process, Duration: 1.8019ms,
8 2024/10/07 05:24:27 Service: test-server1, TraceID: 76350adf-72ee-4382-9b34-211dca1f88ac, SpanID: 0506b161-b095-41d1-a40c-44846ae213bd, ParentSpanID: , Method: GET, Path: /process, Duration: 1.8794ms,
9 2024/10/07 05:24:27 Service: test-server1, TraceID: 9d0d1219-2dbc-4113-9171-670d07ca1e3d, SpanID: 003b6503-d4f7-49f7-876b-344107dd347a, ParentSpanID: , Method: GET, Path: /process, Duration: 1.8976ms,
10 2024/10/07 05:24:27 Service: test-server1, TraceID: 82245e4c-2203-409e-84c1-fa817ca82975, SpanID: 6eb073f5-ea27-4927-9626-9396c70f997d, ParentSpanID: , Method: GET, Path: /process, Duration: 1.7549ms,
```

test-server2.log M X

logs > test-server2.log

```
1 2024/10/07 05:24:19 Service: test-server2, TraceID: 8678e9fe-fec5-4a2f-9db1-9906b06c4dc8, SpanID: 04594a05-d063-4aab-a156-e67da2e4dbfb, ParentSpanID: d87e8bb6-9fb5-4422-98e4-56413d4a4faf, Method: GET,
2 2024/10/07 05:24:25 Service: test-server2, TraceID: 2b6cb680-28e6-438e-bca6-5e1fde98daaf, SpanID: a97b8749-8b76-43f6-8154-57203f76191d, ParentSpanID: 5b12f79d-07b9-48d3-966a-21e61c713ee6, Method: GET,
3 2024/10/07 05:24:26 Service: test-server2, TraceID: 9bec813e-024b-48c0-8f62-2be8e29b03a1, SpanID: d17b9d64-3ed2-4a03-a11a-eeb813915d67, ParentSpanID: f5b3768e-b178-4e7b-b0b4-88ff9cf99a60, Method: GET,
4 2024/10/07 05:24:26 Service: test-server2, TraceID: 3fac5134-fcc1-42b7-a00c-ec86e4ebd19c, SpanID: 5912e0aa-74b6-4131-9942-b8c3b9e5d2fc, ParentSpanID: 466c765f-812c-4dd6-b191-750eb1cd7a37, Method: GET,
5 2024/10/07 05:24:26 Service: test-server2, TraceID: 8d7dcf15-f21f-493b-addf-a6db3710d931, SpanID: 6d5e9aa5-9e64-4ac2-a9cb-4c3592707d64, ParentSpanID: 47dae10d-bb14-4ebb-9374-882927896827, Method: GET,
6 2024/10/07 05:24:27 Service: test-server2, TraceID: 631c2bbd-e3be-45eb-85af-eb99a524fae2, SpanID: e15d6856-9aeb-4c17-b344-f8dc8cc39ba2, ParentSpanID: 3cb7f240-e75a-4bfd-a437-c404e2145c12, Method: GET,
7 2024/10/07 05:24:27 Service: test-server2, TraceID: 9f09a6ff-2d14-40de-9a4e-9287bc677325, SpanID: 129d4925-5b64-4951-bd83-a75952cf6aa2, ParentSpanID: d8439b72-4864-4f36-98f8-bca3fff41997, Method: GET,
8 2024/10/07 05:24:27 Service: test-server2, TraceID: 76350adf-72ee-4382-9b34-211dca1f88ac, SpanID: 54658801-3227-4df8-8071-0cd0cd326d5e, ParentSpanID: 0506b161-b095-41d1-a40c-44846ae213bd, Method: GET,
9 2024/10/07 05:24:27 Service: test-server2, TraceID: 9d0d1219-2dbc-4113-9171-670d07ca1e3d, SpanID: 18b70cc5-2f3d-41b1-9cba-48f0c5258cf3, ParentSpanID: 003b6503-d4f7-49f7-876b-344107dd347a, Method: GET,
10 2024/10/07 05:24:27 Service: test-server2, TraceID: 82245e4c-2203-409e-84c1-fa817ca82975, SpanID: bc6aae44-4086-48ab-b490-aeba47ad06d2, ParentSpanID: 6eb073f5-ea27-4927-9626-9396c70f997d, Method: GET,
11 2024/10/07 05:24:28 Service: test-server2, TraceID: 009e0add-04e3-42e0-830b-0f4f751ff741, SpanID: d7720327-3bb6-47ed-ba65-350aef0b00d2, ParentSpanID: d2f05640-b20f-47b0-8b0b-4dc6b3e7ad40, Method: GET,
```

test-server3.log M X

logs > test-server3.log

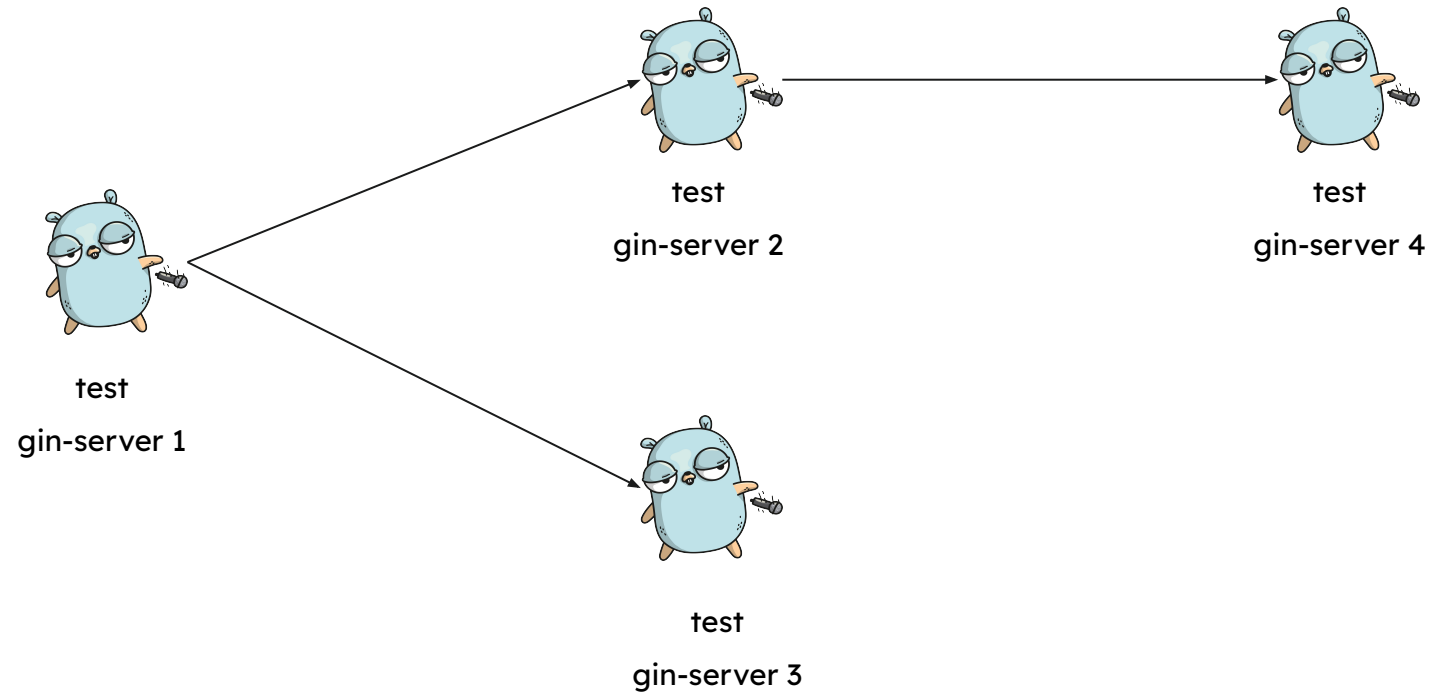
```
1 2024/10/07 05:24:19 Service: test-server3, TraceID: 8678e9fe-fec5-4a2f-9db1-9906b06c4dc8, SpanID: 26b635c1-037a-435e-a1eb-20d5e1f2ea7e, ParentSpanID: d87e8bb6-9fb5-4422-98e4-56413d4a4faf, Method: GET,
2 2024/10/07 05:24:25 Service: test-server3, TraceID: 2b6cb680-28e6-438e-bca6-5e1fde98daaf, SpanID: 7e79539e-3414-4c82-985f-2c8cad228167, ParentSpanID: 5b12f79d-07b9-48d3-966a-21e61c713ee6, Method: GET,
3 2024/10/07 05:24:26 Service: test-server3, TraceID: 9bec813e-024b-48c0-8f62-2be8e29b03a1, SpanID: d3c2437a-8274-44fc-9733-3d3141da452c, ParentSpanID: f5b3768e-b178-4e7b-b0b4-88ff9cf99a60, Method: GET,
4 2024/10/07 05:24:26 Service: test-server3, TraceID: 3fac5134-fcc1-42b7-a00c-ec86e4ebd19c, SpanID: b1ae6c18-0533-4552-9e72-16b7d6719b09, ParentSpanID: 466c765f-812c-4dd6-b191-750eb1cd7a37, Method: GET,
5 2024/10/07 05:24:26 Service: test-server3, TraceID: 8d7dcf15-f21f-493b-addf-a6db3710d931, SpanID: e148bcf9-461d-4044-8dc9-244a6972d23d, ParentSpanID: 47dae10d-bb14-4ebb-9374-882927896827, Method: GET,
6 2024/10/07 05:24:27 Service: test-server3, TraceID: 631c2bbd-e3be-45eb-85af-eb99a524fae2, SpanID: 32e22711-e681-468e-ae0f-5a62a517615f, ParentSpanID: 3cb7f240-e75a-4bfd-a437-c404e2145c12, Method: GET,
7 2024/10/07 05:24:27 Service: test-server3, TraceID: 9f09a6ff-2d14-40de-9a4e-9287bc677325, SpanID: 318990b7-199d-4796-95d3-897d2f620db9, ParentSpanID: d8439b72-4864-4f36-98f8-bca3fff41997, Method: GET,
8 2024/10/07 05:24:27 Service: test-server3, TraceID: 76350adf-72ee-4382-9b34-211dca1f88ac, SpanID: 6e26deda-6010-4788-9e85-371d0413dab3, ParentSpanID: 0506b161-b095-41d1-a40c-44846ae213bd, Method: GET,
9 2024/10/07 05:24:27 Service: test-server3, TraceID: 9d0d1219-2dbc-4113-9171-670d07ca1e3d, SpanID: 274dabbf-dcbf-46b1-806d-bc6289927e25, ParentSpanID: 003b6503-d4f7-49f7-876b-344107dd347a, Method: GET,
```



Demo



DEMO



DEMO Program


```

import (
    "log"
    "net/http"
    "github.com/MAYFCHOI/gin-monitoring/metrics"
    "github.com/MAYFCHOI/gin-monitoring/tracing"
    "github.com/gin-gonic/gin"
)

func main() {
    r := gin.Default()
    r.Use(metrics.MetricsMiddleware(
        metrics.MetricInit{
            ServiceName: "test-server2"}))

    r.Use(tracing.TracingMiddleware(
        tracing.TraceInit{
            ServiceName: "test-server2",
            Logpath:     "../logs/test-server2.log"}))

    client := &http.Client{
        Transport: tracing.NewTracingTransport(http.DefaultTransport),
    }

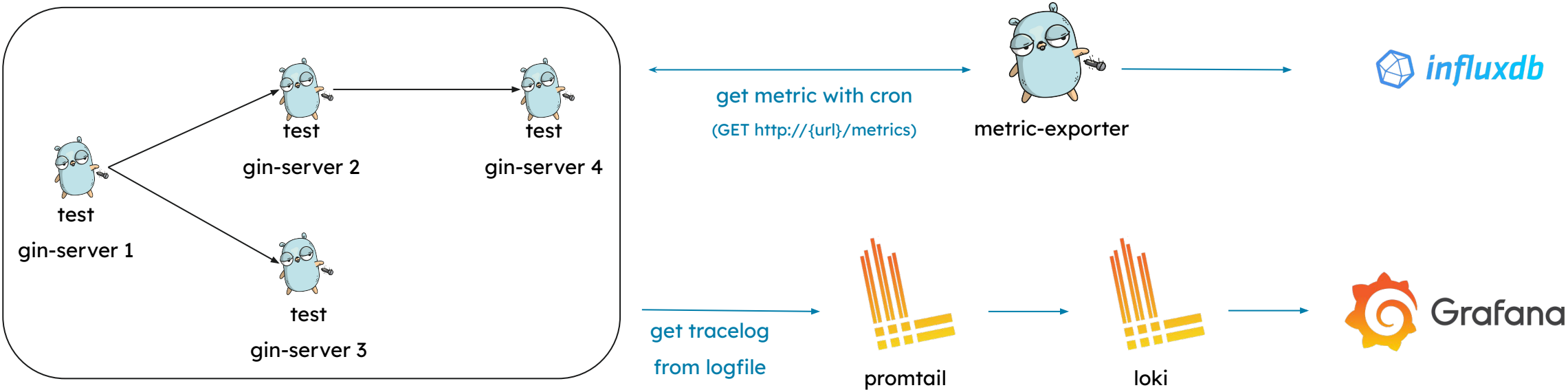
    r.GET("/ping", func(c *gin.Context) {
        c.JSON(200, gin.H{"message": "pong"})
    })
    r.GET("/metrics", metrics.MetricsHandler)
    r.GET("/process", processRequest(client))

    if err := r.Run(":8081"); err != nil {
        log.Fatalf("Failed to run server: %v", err)
    }
}

```



DEMO



DEMO Monitoring Architecture

```
package cron

import (
    "github.com/robfig/cron/v3"
)

func SetupCron() (*cron.Cron, error) {
    c := cron.New()
    _, err := c.AddFunc("*/* * * * *", fetchMetrics)
    if err != nil {
        return nil, err
    }

    return c, nil
}
```




```

func fetchMetric(server config.ServerInfo) (*model.MetricsResponse, error) {
    url := fmt.Sprintf("http://%s:%s%s", server.Url, server.Port, server.Path)
    client := &http.Client{Timeout: 10 * time.Second}
    resp, err := client.Get(url)
    if err != nil {
        return nil, fmt.Errorf("failed to fetch metrics from %s: %v", url, err)
    }
    defer resp.Body.Close()

    body, err := io.ReadAll(resp.Body)
    if err != nil {
        return nil, fmt.Errorf("failed to read response body: %v", err)
    }

    var metrics model.MetricsResponse
    if err := json.Unmarshal(body, &metrics); err != nil {
        return nil, fmt.Errorf("failed to unmarshal response: %v", err)
    }

    return &metrics, nil
}

```



```

package influx

import (
    "context"
    "metric-exporter/config"
    "metric-exporter/model"
    "time"

    influxdb2 "github.com/influxdata/influxdb-client-go/v2"
)

func InfluxInit() {
    config.DB = influxdb2.NewClient(config.Env.Database.Url, config.Env.Database.Token)
}

func GinMetricToInflux(metricsResponse model.MetricsResponse) {
    now := time.Now()
    writeAPI := config.DB.WriteAPIBlocking(config.Env.Database.Org, config.Env.Database.Bucket)

    for path, metric := range metricsResponse.Metrics {
        duration := metric.AvgDuration * float64(metric.Count)
        p := influxdb2.NewPointWithMeasurement(metricsResponse.ServiceName).
            AddTag("path", path).
            AddField("count", metric.Count).
            AddField("duration", duration)
        for k, v := range metric.StatusCodes {
            p.AddField(k, v)
        }
        p.SetTime(now)
        writeAPI.WritePoint(context.Background(), p)
    }
}

```



DEMO

Gin Metric

ADD CELL

ADD NOTE

Show Variables

Enable Annotations

...

?

🔄

SET AUTO REFRESH

📍 Local

This dashboard doesn't have any cells with defined variables. [Learn How](#)



InfluxDB

(metric)

DEMO

Outline

loki

Queries

Logs

Kick start your query

Label browser

Explain query

Label filters

filename = /var/log/trace/test-server.log

Line contains

Text to find

+ Operations

{filename="/var/log/trace/test-server.log"} |= ''

> Options Type: Range Line limit: 1000

+ Add query

Query history

Query inspector

Logs volume

Logs

Time

Unique labels

Wrap lines

Prettify JSON

Deduplication

None

Exact

Numbers

Signature

Common labels: job=traceLog Line limit: 1000 (20 displayed) Total bytes processed: 4.24 kB

> 2024/10/07 05:24:29 Service: test-server1, TraceID: b4cf0942-73b4-4b2f-bf5c-56cd8491e216, SpanID: 991a0a4b-a668-4769-a3ff-8a65ba3edb4e, ParentSpanID: , Method: GET, Path: /process, Duration: 1.7854ms, Status: 200

> 2024/10/07 05:24:29 Service: test-server1, TraceID: 51c91bd5-795c-4978-b527-ee6ed8451388, SpanID: a9d88812-1bd4-4fe6-9db2-17217f69295d, ParentSpanID: , Method: GET, Path: /process, Duration: 1.8118ms, Status: 200

> 2024/10/07 05:24:29 Service: test-server1, TraceID: a876b8eb-d7ae-4995-838c-3662e00a9651, SpanID: 8d45100a-f594-40b0-a9cd-087f87122e8d, ParentSpanID: , Method: GET, Path: /process, Duration: 1.6913ms, Status: 200

> 2024/10/07 05:24:29 Service: test-server1, TraceID: 068afeff-ae61-47ed-acbb-32cbdfda0011, SpanID: ee08cf4a-e589-457f-85f5-60db8d1931b2, ParentSpanID: , Method: GET, Path: /process, Duration: 1.8419ms, Status: 200

> 2024/10/07 05:24:29 Service: test-server1, TraceID: 3c6e655e-88cf-4e28-9aca-0d35951e343d, SpanID: 7d07418e-2b30-4953-82ba-8d4d826299c6, ParentSpanID: , Method: GET, Path: /process, Duration: 1.6983ms, Status: 200

> 2024/10/07 05:24:28 Service: test-server1, TraceID: 9331ad2b-5f7b-49a5-a853-6e3c8a6fb492, SpanID: 3f9b04fb-d367-46e4-822f-a4109d96e519, ParentSpanID: , Method: GET, Path: /process, Duration: 1.6768ms, Status: 200

> 2024/10/07 05:24:28 Service: test-server1, TraceID: ae192401-6892-480b-be65-8cc7689bdfef, SpanID: 88de2837-77f9-4423-b66d-df1481fa1ccf, ParentSpanID: , Method: GET, Path: /process, Duration: 1.5807ms, Status: 200

> 2024/10/07 05:24:28 Service: test-server1, TraceID: e8707b36-bdf9-4b02-bc64-29a8fd515820, SpanID: bbb03c1b-14c2-4b82-ac64-d16340810ba7, ParentSpanID: , Method: GET, Path: /process, Duration: 1.7967ms, Status: 200

> 2024/10/07 05:24:28 Service: test-server1, TraceID: 26587ce8-e74d-4da7-80b2-6d27b9b143f9, SpanID: 6eacd286-7e6f-4a6d-9c8e-beaf99cd4d92, ParentSpanID: , Method: GET, Path: /process, Duration: 1.8159ms, Status: 200

> 2024/10/07 05:24:28 Service: test-server1, TraceID: 88ae0ddd-84e3-42e9-838b-0f4751ff7a11, SpanID: d2f95649-ba0f-42b8-ab8b-4dc6bae2ed40, ParentSpanID: , Method: GET, Path: /process, Duration: 2.020399ms, Status: 200

> 2024/10/07 05:24:27 Service: test-server1, TraceID: 82245e4c-2283-409e-84c1-fa817ca82975, SpanID: 6eb073f5-ea27-4927-9626-9396c70f997d, ParentSpanID: , Method: GET, Path: /process, Duration: 1.7549ms, Status: 200

> 2024/10/07 05:24:27 Service: test-server1, TraceID: 9d0d1219-2dbc-4113-9171-670d07ca1e3d, SpanID: 003b6503-d4f7-49f7-876b-344107dd347a, ParentSpanID: , Method: GET, Path: /process, Duration: 1.8976ms, Status: 200

> 2024/10/07 05:24:27 Service: test-server1, TraceID: 76350adf-72ee-4382-9b34-211dca1f88ac, SpanID: 0506b161-b095-41d1-a48c-44846ae213bd, ParentSpanID: , Method: GET, Path: /process, Duration: 1.8794ms, Status: 200

> 2024/10/07 05:24:27 Service: test-server1, TraceID: 9f09a6ff-d214-48de-9a4e-9287bc677325, SpanID: d8439b72-4864-4f36-98f8-bca3ffff1497, ParentSpanID: , Method: GET, Path: /process, Duration: 1.8019ms, Status: 200

> 2024/10/07 05:24:27 Service: test-server1, TraceID: 631c2bbd-e3be-45eb-85af-eb99a524fae2, SpanID: 3cb7f240-e75a-4bfd-a437-c404e2145c12, ParentSpanID: , Method: GET, Path: /process, Duration: 2.1156ms, Status: 200

> 2024/10/07 05:24:26 Service: test-server1, TraceID: 8d7dcf15-f21f-493b-addf-a6db3710d931, SpanID: 47dae10d-bb14-4ebb-9374-882927896827, ParentSpanID: , Method: GET, Path: /process, Duration: 1.7948ms, Status: 200

> 2024/10/07 05:24:26 Service: test-server1, TraceID: 3fac5134-fcc1-42b7-a00c-ec86e4ebd19c, SpanID: 466c765f-812c-4dd6-b191-750eb1cd7a37, ParentSpanID: , Method: GET, Path: /process, Duration: 1.8531ms, Status: 200

> 2024/10/07 05:24:26 Service: test-server1, TraceID: 9bec813e-024b-48c0-8f62-2be8e29b03a1, SpanID: f5b3768e-b178-4e7b-b0b4-88ff9cf99a60, ParentSpanID: , Method: GET, Path: /process, Duration: 4.1867ms, Status: 200

> 2024/10/07 05:24:25 Service: test-server1, TraceID: 2b6cb680-28a6-438e-bca6-5e1fde98daaf, SpanID: 5b12f79d-07b9-48d3-966a-21e61c713ee6, ParentSpanID: , Method: GET, Path: /process, Duration: 3.9809ms, Status: 200

> 2024/10/07 05:24:19 Service: test-server1, TraceID: 8678e9fe-fec5-4a2f-9db1-9906b06c4dc8, SpanID: d87e8bb6-9fb5-4422-98e4-56413d4a4faf, ParentSpanID: , Method: GET, Path: /process, Duration: 4.7664ms, Status: 200

Grafana

(tracelog)

GopherCon Korea 2024

test-server1.tracelog

```
> 2024/10/07 05:24:29 Service: test-server1, TraceID: b4cf0942-73b4-4b2f-bf5c-56dc8491e216, SpanID: 991a0a4b-a668-4769-a3ff-8a65ba3edb4e, ParentSpanID: , Method: GET, Path: /process, Duration: 1.7854ms, Status: 200
> 2024/10/07 05:24:29 Service: test-server1, TraceID: 51c91bd5-795c-4970-b527-ee6ed8451388, SpanID: a9d88812-1bd4-4fe6-9db2-17217f69295d, ParentSpanID: , Method: GET, Path: /process, Duration: 1.8118ms, Status: 200
> 2024/10/07 05:24:29 Service: test-server1, TraceID: a876b8eb-d7ae-4995-838c-3662e00a9651, SpanID: 8d45100a-f594-40b0-a9cd-087f87122e8d, ParentSpanID: , Method: GET, Path: /process, Duration: 1.6913ms, Status: 200
> 2024/10/07 05:24:29 Service: test-server1, TraceID: 068afeff-ae61-47ed-acbb-32cbdfda0011, SpanID: ee08cf4a-e589-457f-85f5-60db8d1931b2, ParentSpanID: , Method: GET, Path: /process, Duration: 1.8419ms, Status: 200
> 2024/10/07 05:24:29 Service: test-server1, TraceID: 3c6e655e-88cf-4e28-9aca-0d35951e343d, SpanID: 7d87418e-2b30-4953-82ba-8d4d826299c6, ParentSpanID: , Method: GET, Path: /process, Duration: 1.6983ms, Status: 200
> 2024/10/07 05:24:28 Service: test-server1, TraceID: 9331ad2b-5f7b-49a5-a853-6e3c8a6fb492, SpanID: 3f9b04fb-d367-46e4-822f-a4109d96e519, ParentSpanID: , Method: GET, Path: /process, Duration: 1.6768ms, Status: 200
> 2024/10/07 05:24:28 Service: test-server1, TraceID: ae192401-6892-400b-be65-8cc7689bdbef, SpanID: 88de2837-77f9-4423-b66d-df1481fa1ccf, ParentSpanID: , Method: GET, Path: /process, Duration: 1.5807ms, Status: 200
> 2024/10/07 05:24:28 Service: test-server1, TraceID: e8707b36-bdf9-4b02-bc64-29a8fd515820, SpanID: bbb03c1b-14c2-4b82-ac64-d16340810ba7, ParentSpanID: , Method: GET, Path: /process, Duration: 1.7967ms, Status: 200
> 2024/10/07 05:24:28 Service: test-server1, TraceID: 26587ce8-e74d-4da7-80b2-6d27b9b143f9, SpanID: 6eadc286-7e6f-4a6d-9c8e-beaf99cd4d92, ParentSpanID: , Method: GET, Path: /process, Duration: 1.8159ms, Status: 200
> 2024/10/07 05:24:28 Service: test-server1, TraceID: 88ae0ddd-84e3-42e9-838b-0f4751ff7a11, SpanID: d2f95649-ba0f-42b8-ab8b-4dc6bae2ed40, ParentSpanID: , Method: GET, Path: /process, Duration: 2.020399ms, Status: 200
> 2024/10/07 05:24:27 Service: test-server1, TraceID: 82245e4c-2203-409e-84c1-fa817ca82975, SpanID: 6eb073f5-ea27-4927-9626-9396c70f997d, ParentSpanID: , Method: GET, Path: /process, Duration: 1.7549ms, Status: 200
> 2024/10/07 05:24:27 Service: test-server1, TraceID: 9d0d1219-2dbc-4113-9171-670d07ca1e3d, SpanID: 003b6503-d4f7-49f7-876b-344107dd347a, ParentSpanID: , Method: GET, Path: /process, Duration: 1.8976ms, Status: 200
> 2024/10/07 05:24:27 Service: test-server1, TraceID: 76850cdf-703a-4000-0b34-011ddc1f003a, SpanID: 0f0b5161-5005-41d1-4d0c-4d046e0a30b3, ParentSpanID: , Method: GET, Path: /process, Duration: 1.8704ms, Status: 200
```

test-server2.tracelog

```
> 2024/10/07 05:24:29 Service: test-server2, TraceID: b4cf0942-73b4-4b2f-bf5c-56dc8491e216, SpanID: 3dbf526f-e498-4fa6-b4f9-bf64d92433b1, ParentSpanID: 991a0a4b-a668-4769-a3ff-8a65ba3edb4e, Method: GET, Path: /process, Duration: 639.8µs, Status: 200
> 2024/10/07 05:24:29 Service: test-server2, TraceID: 51c91bd5-795c-4970-b527-ee6ed8451388, SpanID: a10b7665-86d8-4aa3-b483-87c4c305e68c, ParentSpanID: a9d88812-1bd4-4fe6-9db2-17217f69295d, Method: GET, Path: /process, Duration: 525.7µs, Status: 200
> 2024/10/07 05:24:29 Service: test-server2, TraceID: a876b8eb-d7ae-4995-838c-3662e00a9651, SpanID: 107d905a-cae6-4d1a-aa70-80bf49dd1957, ParentSpanID: 8d45100a-f594-40b0-a9cd-087f87122e8d, Method: GET, Path: /process, Duration: 571.3µs, Status: 200
> 2024/10/07 05:24:29 Service: test-server2, TraceID: 068afeff-ae61-47ed-acbb-32cbdfda0011, SpanID: 8f1b0f50-c8d2-4a05-ac42-26668025934b, ParentSpanID: ee08cf4a-e589-457f-85f5-60db8d1931b2, Method: GET, Path: /process, Duration: 684µs, Status: 200
> 2024/10/07 05:24:29 Service: test-server2, TraceID: 3c6e655e-88cf-4e28-9aca-0d35951e343d, SpanID: 1e7f8616-2005-423d-9536-db67ca1d2393, ParentSpanID: 7d87418e-2b30-4953-82ba-8d4d826299c6, Method: GET, Path: /process, Duration: 549.7µs, Status: 200
> 2024/10/07 05:24:28 Service: test-server2, TraceID: 9331ad2b-5f7b-49a5-a853-6e3c8a6fb492, SpanID: 92a8556d-1ffa-4bee-8915-103aaa931fda, ParentSpanID: 3f9b04fb-d367-46e4-822f-a4109d96e519, Method: GET, Path: /process, Duration: 555.8µs, Status: 200
> 2024/10/07 05:24:28 Service: test-server2, TraceID: ae192401-6892-400b-be65-8cc7689bdbef, SpanID: fdc81d74-a1bc-4c75-a173-907db51876c8, ParentSpanID: 88de2837-77f9-4423-b66d-df1481fa1ccf, Method: GET, Path: /process, Duration: 503.2µs, Status: 200
> 2024/10/07 05:24:28 Service: test-server2, TraceID: e8707b36-bdf9-4b02-bc64-29a8fd515820, SpanID: 7a04b50f-69ed-437b-ab4c-1a275f06af65, ParentSpanID: bbb03c1b-14c2-4b82-ac64-d16340810ba7, Method: GET, Path: /process, Duration: 578.1µs, Status: 200
> 2024/10/07 05:24:28 Service: test-server2, TraceID: 26587ce8-e74d-4da7-80b2-6d27b9b143f9, SpanID: bb8abf09-d4c2-4ee6-9270-76136e02cabf, ParentSpanID: 6eadc286-7e6f-4a6d-9c8e-beaf99cd4d92, Method: GET, Path: /process, Duration: 545.1µs, Status: 200
> 2024/10/07 05:24:28 Service: test-server2, TraceID: 88ae0ddd-84e3-42e9-838b-0f4751ff7a11, SpanID: d7738327-3bb6-47ed-be65-359eef0b90d2, ParentSpanID: d2f95649-ba0f-42b8-ab8b-4dc6bae2ed40, Method: GET, Path: /process, Duration: 664µs, Status: 200
> 2024/10/07 05:24:27 Service: test-server2, TraceID: 82245e4c-2203-409e-84c1-fa817ca82975, SpanID: bc6aae44-4086-48ab-b490-aeba47ad06d2, ParentSpanID: 6eb073f5-ea27-4927-9626-9396c70f997d, Method: GET, Path: /process, Duration: 548.4µs, Status: 200
> 2024/10/07 05:24:27 Service: test-server2, TraceID: 9d0d1219-2dbc-4113-9171-670d07ca1e3d, SpanID: 18b70cc5-2fd3-41b1-9cba-48f0c5258cf3, ParentSpanID: 003b6503-d4f7-49f7-876b-344107dd347a, Method: GET, Path: /process, Duration: 555.4µs, Status: 200
> 2024/10/07 05:24:27 Service: test-server2, TraceID: 76850cdf-703a-4000-0b34-011ddc1f003a, SpanID: 0f0b5161-5005-41d1-4d0c-4d046e0a30b3, ParentSpanID: , Method: GET, Path: /process, Duration: 594.5µs, Status: 200
```

test-server3.tracelog

```
> 2024/10/07 05:24:29 Service: test-server3, TraceID: b4cf0942-73b4-4b2f-bf5c-56dc8491e216, SpanID: aca90cc7-3d3c-4c45-809d-d939f70abb1d, ParentSpanID: 991a0a4b-a668-4769-a3ff-8a65ba3edb4e, Method: GET, Path: /process, Duration: 9.1µs, Status: 200
> 2024/10/07 05:24:29 Service: test-server3, TraceID: 51c91bd5-795c-4970-b527-ee6ed8451388, SpanID: a8f59dd3-7c00-4c11-902f-02265ab502b6, ParentSpanID: a9d88812-1bd4-4fe6-9db2-17217f69295d, Method: GET, Path: /process, Duration: 10.5µs, Status: 200
> 2024/10/07 05:24:29 Service: test-server3, TraceID: a876b8eb-d7ae-4995-838c-3662e00a9651, SpanID: ffd1eb85-60cc-4b58-8486-9afed66c3fe3, ParentSpanID: 8d45100a-f594-40b0-a9cd-087f87122e8d, Method: GET, Path: /process, Duration: 15.4µs, Status: 200
> 2024/10/07 05:24:29 Service: test-server3, TraceID: 068afeff-ae61-47ed-acbb-32cbdfda0011, SpanID: a099cf35-f3db-4138-a351-8c1f59add1a2, ParentSpanID: ee08cf4a-e589-457f-85f5-60db8d1931b2, Method: GET, Path: /process, Duration: 9.5µs, Status: 200
> 2024/10/07 05:24:29 Service: test-server3, TraceID: 3c6e655e-88cf-4e28-9aca-0d35951e343d, SpanID: 2987b211-494f-405f-992f-046fa26255e7, ParentSpanID: 7d87418e-2b30-4953-82ba-8d4d826299c6, Method: GET, Path: /process, Duration: 10.7µs, Status: 200
> 2024/10/07 05:24:28 Service: test-server3, TraceID: 9331ad2b-5f7b-49a5-a853-6e3c8a6fb492, SpanID: b373665d-1f4a-4fe6-b742-97986366e992, ParentSpanID: 3f9b04fb-d367-46e4-822f-a4109d96e519, Method: GET, Path: /process, Duration: 9.6µs, Status: 200
> 2024/10/07 05:24:28 Service: test-server3, TraceID: ae192401-6892-400b-be65-8cc7689bdbef, SpanID: 14cdc171-6e48-452b-a5a4-d1f5f8d6e6c5, ParentSpanID: 88de2837-77f9-4423-b66d-df1481fa1ccf, Method: GET, Path: /process, Duration: 9.4µs, Status: 200
> 2024/10/07 05:24:28 Service: test-server3, TraceID: e8707b36-bdf9-4b02-bc64-29a8fd515820, SpanID: cba6fdd8-de63-463e-949d-a73d28e1df76, ParentSpanID: bbb03c1b-14c2-4b82-ac64-d16340810ba7, Method: GET, Path: /process, Duration: 9.6µs, Status: 200
> 2024/10/07 05:24:28 Service: test-server3, TraceID: 26587ce8-e74d-4da7-80b2-6d27b9b143f9, SpanID: 1d41bdce-6373-4aa8-b642-00291db8330f, ParentSpanID: 6eadc286-7e6f-4a6d-9c8e-beaf99cd4d92, Method: GET, Path: /process, Duration: 10.7µs, Status: 200
> 2024/10/07 05:24:28 Service: test-server3, TraceID: 88ae0ddd-84e3-42e9-838b-0f4751ff7a11, SpanID: 18d37d8b-eede-4b54-a646-79ee530bae0a, ParentSpanID: d2f95649-ba0f-42b8-ab8b-4dc6bae2ed40, Method: GET, Path: /process, Duration: 12.4µs, Status: 200
> 2024/10/07 05:24:27 Service: test-server3, TraceID: 82245e4c-2203-409e-84c1-fa817ca82975, SpanID: da12e741-3579-41f2-b0f0-bcbad55c5168, ParentSpanID: 6eb073f5-ea27-4927-9626-9396c70f997d, Method: GET, Path: /process, Duration: 12.2µs, Status: 200
> 2024/10/07 05:24:27 Service: test-server3, TraceID: 9d0d1219-2dbc-4113-9171-670d07ca1e3d, SpanID: 274dabbf-dcbf-46b1-806d-bc6289927e25, ParentSpanID: 003b6503-d4f7-49f7-876b-344107dd347a, Method: GET, Path: /process, Duration: 10.3µs, Status: 200
> 2024/10/07 05:24:27 Service: test-server3, TraceID: 76850cdf-703a-4000-0b34-011ddc1f003a, SpanID: 0f0b5161-5005-41d1-4d0c-4d046e0a30b3, ParentSpanID: , Method: GET, Path: /process, Duration: 10.5µs, Status: 200
```

test-server4.tracelog

```
> 2024/10/07 05:24:29 Service: test-server4, TraceID: b4cf0942-73b4-4b2f-bf5c-56dc8491e216, SpanID: beab1668-471d-4ff1-8eb5-6f378557cab2, ParentSpanID: 3dbf526f-e498-4fa6-b4f9-bf64d92433b1, Method: GET, Path: /process, Duration: 8.8µs, Status: 200
> 2024/10/07 05:24:29 Service: test-server4, TraceID: 51c91bd5-795c-4970-b527-ee6ed8451388, SpanID: 9c29f8f7-ed97-4da7-a22a-5c094520d475, ParentSpanID: a10b7665-86d8-4aa3-b483-87c4c305e68c, Method: GET, Path: /process, Duration: 8.9µs, Status: 200
```

Q&A



Thank you!

