Monitor Go with Prometheus + Grafana

Application Monitoring

Grafana

Create File Main.go

To start, create a Golang application using the Gin framework with a file named `main.go`.

```go
// main.go
package main

import (
  "go-gin-prometheus/metrics"
  "net/http"

  "github.com/gin-gonic/gin"
  "github.com/prometheus/client_golang/prometheus/promhttp"
)

func main() {
  metrics.RegisterMetrics()

  r := gin.Default()

  r.POST("/create", func(c *gin.Context) {
    metrics.CreateCounter.Inc()

    c.JSON(http.StatusOK, JsonResponse{
      Code: http.StatusOK, Message: "Resource created successfully",
    })
  })

  r.GET("/read", func(c *gin.Context) {
    metrics.ReadCounter.Inc()

    c.JSON(http.StatusOK, JsonResponse{
      Code: http.StatusOK, Message: "Resource read successfully",
    })
  })

  r.PUT("/update", func(c *gin.Context) {
    metrics.UpdateCounter.Inc()

    c.JSON(http.StatusOK, JsonResponse{
      Code: http.StatusOK, Message: "Resource updated successfully",
    })
  })

  r.DELETE("/delete", func(c *gin.Context) {
    metrics.DeleteCounter.Inc()

    c.JSON(http.StatusOK, JsonResponse{
      Code: http.StatusOK, Message: "Resource deleted successfully",
    })
  })

  r.GET("/metrics", gin.WrapH(promhttp.Handler()))

  r.Run(":8080")
}

type JsonResponse struct {
  Code    int    `json:"code"`
  Message string `json:"message"`
}
```

Create File metric.go

Next, create a folder named `metrics`, where inside that folder there is a file named `metrics.go`.

```go
package metrics

import (
  "github.com/prometheus/client_golang/prometheus"
)

var (
  CreateCounter = prometheus.NewCounter(prometheus.CounterOpts{
    Name: "gin_create_req_total",
    Help: "Total user hit of create requests",
  })
  ReadCounter = prometheus.NewCounter(prometheus.CounterOpts{
    Name: "gin_read_req_total",
    Help: "Total user hit of read requests",
  })
  UpdateCounter = prometheus.NewCounter(prometheus.CounterOpts{
    Name: "gin_update_req_total",
    Help: "Total user hit of update requests",
  })
  DeleteCounter = prometheus.NewCounter(prometheus.CounterOpts{
    Name: "gin_delete_req_total",
    Help: "Total user hit of delete requests",
  })
)

func RegisterMetrics() {
  prometheus.MustRegister(CreateCounter, ReadCounter, UpdateCounter,
DeleteCounter)
}
```

GRAFANA MOCKUP

SEE YOU

.....