# go - How to push metrics to prometheus using client\_golang? - Stack Overflow

http://stackoverflow.com/questions/37611754/how-to-push-metrics-to-prometheus-using-client-golang

Prometheus is a pull-based system, if you want push-based monitoring, you need to use a gateway of some sort. A minimal example (without actually doing anything useful like starting an HTTP listener, or actually doing anything to a metric) follows:

import (

"github.com/prometheus/client\_golang/prometheus"

"net/http"

)

var responseMetric = prometheus.NewHistogram(

prometheus.HistogramOpts{

Name: "request\_duration\_milliseconds",

Help: "Request latency distribution",

Buckets: prometheus.ExponentialBuckets(10.0, 1.13, 40),

})

func main() {

prometheus.MustRegister(responseMetric)

http.Handle("/metrics", prometheus.Handler())

// Any other setup, then an http.ListenAndServe here

}

You then need to configure Prometheus to scrape the /metrics page your binary provides.

package main

import (

"net/http"

"github.com/prometheus/client\_golang/prometheus"

)

var (

cpuTemp = prometheus.NewGauge(prometheus.GaugeOpts{

Name: "cpu\_temperature\_celsius",

Help: "Current temperature of the CPU.",

})

hdFailures = prometheus.NewCounter(prometheus.CounterOpts{

Name: "hd\_errors\_total",

Help: "Number of hard-disk errors.",

})

)

func init() {

prometheus.MustRegister(cpuTemp)

prometheus.MustRegister(hdFailures)

}

func main() {

cpuTemp.Set(65.3)

hdFailures.Inc()

http.Handle("/metrics", prometheus.Handler())

http.ListenAndServe(":8080", nil)

}

`

This might be useful to some.