

Introduction To Prometheus Monitoring

9 May 2020

Sean Malloy

Introduction

```
package main

import "fmt"

type Presenter struct {
    Name      string
    Employeer string
    Title     string
    Role      string
}

func main() {
    x := Presenter{}
    x.Name = "Sean Malloy"
    x.Employeer = "Kohl's Departments Stores"
    x.Title = "Software Engineer"
    x.Role = "Cloud Platform Automation(OpenShift, k8s, Vault)"
    fmt.Println(x.Name)
    fmt.Println(x.Employeer)
    fmt.Println(x.Title)
    fmt.Println(x.Role)
}
```

Run

Problem Statement

Need a low cost solution for monitoring infrastructure and applications in both dynamic and static environments.

Prometheus To The Rescue!

- Open Source Apache 2.0 License

- CNCF Project(vendor neutral part of Linux Foundation)
- Metrics based monitoring
- Dynamically discover monitoring targets in cloud environments

Software Versions

- Prometheus v2.18.1
- Alertmanager v0.20.0
- node_exporter v0.18.1

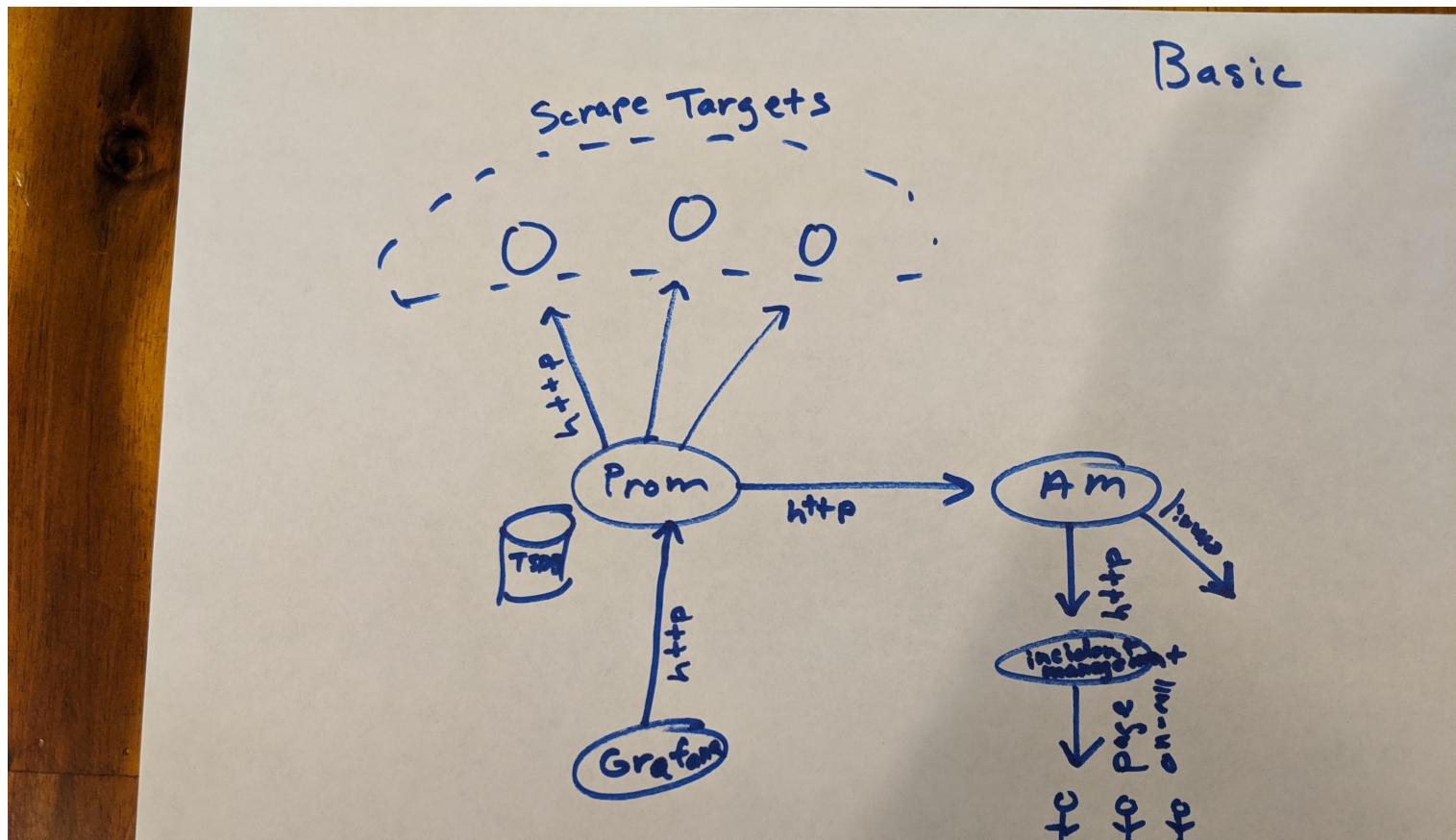
Basics - Architecture

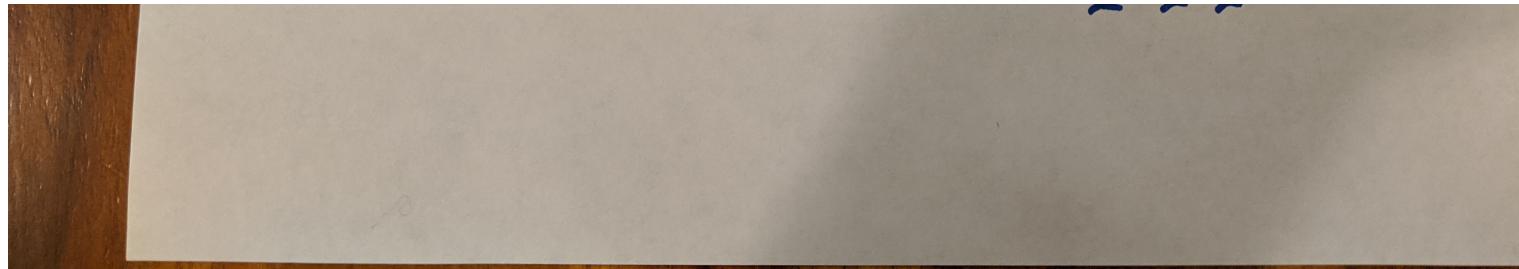
A Prometheus deployment consists of Prometheus, Alertmanager, and Grafana(optional). These components can be run on physical, virtual, containerized infrastructure.

- Metrics, not commands, used for monitoring
- Pull model, not push

- Data stored in Prometheus TSDB
- Exporters(agents) used to collect metrics

Basics - Diagram





Basics - Run Locally

```
#!/bin/bash

node_exporter &
prometheus --config.file="./code/prometheus.yml" &
alertmanager --config.file="./code/alertmanager.yml" &

sleep 15
echo "Prometheus http://localhost:9090"
echo "Alertmanager http://localhost:9093"
```

Run

- Scrape Configurations
- Alert Rules Configuration
- Alertmanager Configuration

Basics - Prometheus Metrics Format

Prometheus exporters expose metrics in a text format(not JSON, not HTML).

```
#!/bin/bash  
  
curl http://localhost:9100/metrics
```

Run

Basics - PromQL and Metric Types

Prometheus has four different metric types.

- Counter
- Gauge
- Histogram
- Summary

prometheus.io/docs/concepts/metric_types/ (https://prometheus.io/docs/concepts/metric_types/)

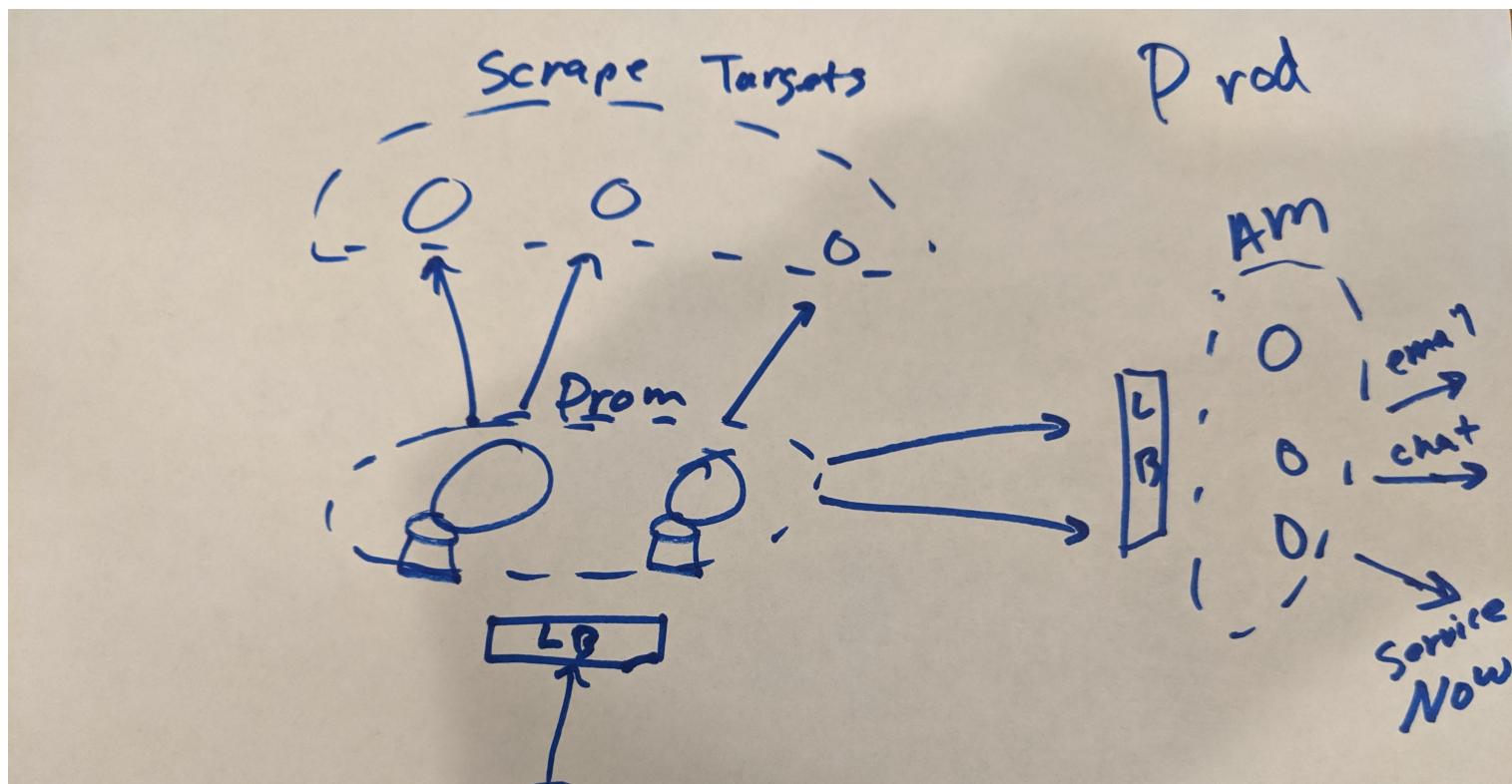
Service Discovery

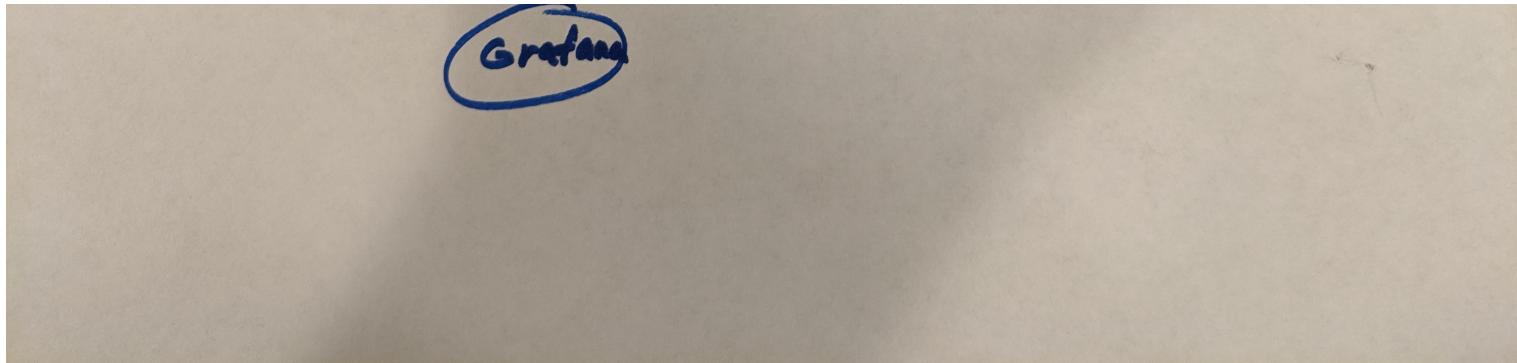
Dynamically discover scrape targets in cloud environments. This example uses Google Cloud Platform.

prometheus.io/docs/prometheus/latest/configuration/configuration/#gce_sd_config

(https://prometheus.io/docs/prometheus/latest/configuration/configuration/#gce_sd_config)

Production Deployment





Other Topics

- Relabeling
- Recording Rules
- List of exporters
- Federation
- Remote read/write
- Grafana
- Open Metrics

References

- prometheus.io (<https://prometheus.io>)
- github.com/prometheus (<https://github.com/prometheus>)
- github.com/prometheus-community (<https://github.com/prometheus-community>)
- grafana.com (<https://grafana.com>)
- www.cncf.io (<https://www.cncf.io>)
- openmetrics.io (<https://openmetrics.io>)

Questions

???

Thank you

Sean Malloy

spinelli85@gmail.com (<mailto:spinelli85@gmail.com>)

<http://seanmalloy.com>

<http://spmalloy.com> (<http://spmalloy.com>)

@spmalloy (<http://twitter.com/spmalloy>)

