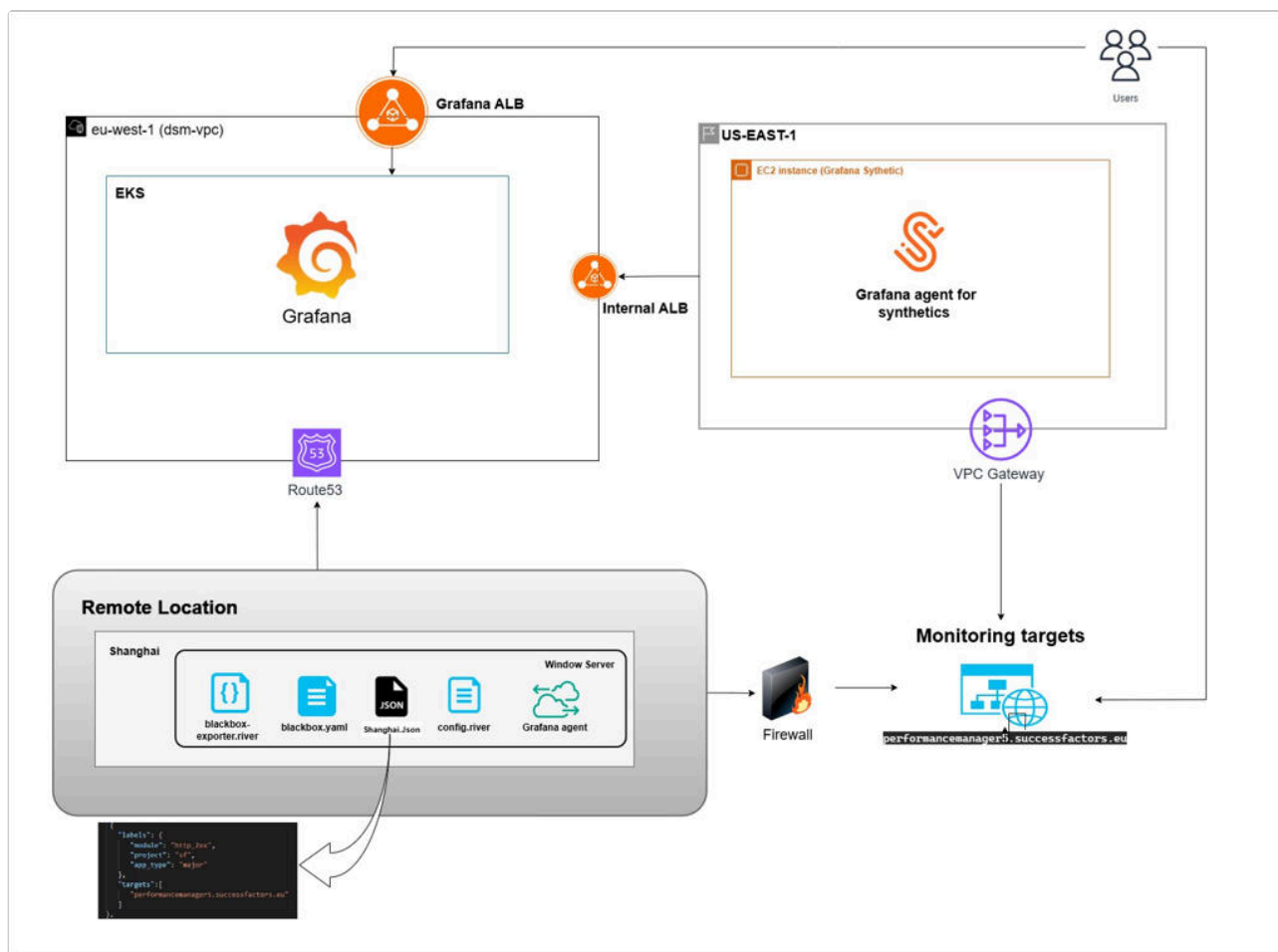


Infrastructure Monitoring with Grafana Agent for New Location Expansion

This document outlines the current infrastructure monitoring setup and provides details on how to expand synthetic monitoring by deploying additional **Grafana Agent** instances at new remote locations. This approach ensures comprehensive monitoring of our application's performance through HTTP path-based synthetic checks, leveraging **Grafana**, **Mimir**, and **Blackbox Exporter**.

By expanding our monitoring footprint with additional virtual machines (VMs) in the new location, we can improve the resilience and scalability of our infrastructure monitoring solution. The goal is to deploy these VMs with Grafana Agent configured to perform regular synthetic checks and send collected data to **Mimir** for visualization and alerting in **Grafana**.



Objective

- **Expand Monitoring Reach:** Deploy additional monitoring infrastructure in a new location to ensure application health is tracked from multiple geographic regions.
- **Ensure Consistency:** Maintain the same monitoring configuration and metrics collection process across all locations to provide uniformity in alerting, reporting, and troubleshooting.
- **Support Scalability:** Add more VMs to handle increased load and ensure high availability and redundancy in monitoring.

Current Monitoring Setup

Overview of Existing Setup

Currently, our monitoring solution involves **Grafana Agents** deployed at remote locations, using the **Blackbox Exporter** to conduct HTTP path-based synthetic checks. These agents collect performance metrics every 5 minutes, which are then pushed to **Mimir** for storage and querying. **Grafana** is used to visualize the data and trigger alerts based on thresholds.

Components

- **Grafana Agent:** Collects and exports monitoring data from remote locations using the Blackbox Exporter.
- **Blackbox Exporter:** Module within the Grafana Agent that performs synthetic HTTP checks to monitor key endpoints such as `/health` or `/login`.
- **Mimir:** Centralized data store for the collected metrics.
- **Grafana:** Visualization tool used for monitoring dashboards and alerting.
- **Loki:** Collecting logs

Data Flow

1. **Synthetic Monitoring:**
 - The Grafana Agent sends HTTP requests to defined application paths at set intervals (every 5 minutes).
 - Metrics are collected regarding response time, availability, and other performance indicators.
2. **Metric Collection:**

- The collected metrics are forwarded to **Mimir**.

3. Visualization:

- The data stored in Mimir is queried by **Grafana** for visualization, helping the team analyze application performance and detect issues.

Expansion Plan for New Location

To extend monitoring to the new location, we need to deploy **additional virtual machines (VMs)** with the Grafana Agent. These VMs will perform synthetic checks on the application from the new location and report metrics back to Mimir.

Grafana Agent Configuration

Configure Grafana Agent for Synthetic Monitoring

On each new VM, configure the **Grafana Agent** to use the **Blackbox Exporter** for HTTP path-based checks. The configuration file (`config.river`) defines the synthetic monitoring tasks.

Here is a sample configuration tailored to the new VM:

```
1 logging {
2   level = "info"
3   format = "logfmt"
4 }
5
6 discovery.http "config" {
7   url = "https://synthetics-config.mx.dsm.app/eu-west-1.json"
8 }
9
10 discovery.relabel "config" {
11   targets = discovery.http.config.targets
12
13   rule {
14     source_labels = ["module"]
15     target_label = "__param_module"
16   }
17
18   rule {
19     source_labels = ["__param_target"]
20     target_label = "target"
21   }
22
23   rule {
24     target_label = "dtap"
25     replacement = "prd"
26   }
27 }
28
29 remote.http "blackbox_config" {
30   url = "https://synthetics-config.mx.dsm.app/blackbox_exporter.yaml"
31 }
```

```

32 module.http "blackbox_exporter" {
33     url = "https://synthetics-config.mx.dsm.app/blackbox.river"
34     arguments {
35         config = remote.http.blackbox_config.content
36         targets = discovery.relabel.config.output
37         location = "geleen"
38     }
39 }
40
41 prometheus.scrape "synthetics_config" {
42     targets = module.http.blackbox_exporter.exports.targets
43     forward_to = [prometheus.remote_write.mimir.receiver]
44     scrape_interval = "5m"
45 }
46
47 prometheus.remote_write "mimir" {
48     endpoint {
49         url = "https://mimir.mx.dsm.app/api/v1/push"
50     }
51 }
52
53 loki.source.journal "read" {
54     forward_to = [loki.write.gateway.receiver]
55     labels = {dtap = "prd", location = "eu-west-1"}
56 }
57
58 loki.write "gateway" {
59     endpoint {
60         url = "https://loki.mx.dsm.app/loki/api/v1/push"
61     }
62 }

```

- **Target URLs:** Replace with application paths that need to be monitored.
- **Scrape Interval:** Data will be captured every 5 minutes.
- **Instance Label:** Differentiate this instance label for easy identification in Grafana and Mimir.

Here is the location.json

```

1  [
2      {
3          "labels" : {
4              "module" : "http_2xx",
5              "project": "mendix",
6              "app_type": "critical",
7              "alert_type": "P1"
8          },
9          "targets" : [
10             "supermarketone.mx.dsm.app",
11             "demo.mx.dsm.app"
12         ]
13     },
14     {
15         "labels" : {
16             "module" : "http_2xx",
17             "project": "mendix",

```

```
18         "app_type": "major",
19         "alert_type": "P2"
20     },
21     "targets" : [
22         "supermarketone.mx.dsm.app",
23         "demo.mx.dsm.app"
24     ]
25 }
```

Supported Platforms

Linux

- **Minimum Version:** Kernel 2.6.32 or later
- **Architectures:** AMD64, ARM64

Windows

- **Minimum Version:** Windows Server 2016 or later, or Windows 10 or later
- **Architectures:** AMD64

macOS

- **Minimum Version:** macOS 10.13 or later
- **Architectures:** AMD64 (Intel), ARM64 (Apple Silicon)

FreeBSD

- **Minimum Version:** FreeBSD 10 or later
- **Architectures:** AMD64

Resource Estimation

- **CPU:** 1 core
- **Memory:** 2 GB
- **Bandwidth:** 1.5 MB/sec

For more information on estimating resource usage, refer to the [official documentation](#).

Conclusion

Expanding our synthetic monitoring infrastructure by deploying additional VMs in a new location will significantly enhance our application's monitoring capabilities, improve response times, and provide early warnings for potential issues. This will help maintain high availability and ensure that our application performs optimally across different regions.

For more details please read

1. [Grafana Synthetics \(HTTP based\)](#)