Container Native FS Interposer

Julia Hua, Jiawei Xiang, Hilario Gonzalez, Juncheng Cao
Mentors: Vasily Tarasov, Alex Merenstein

November 6, 2024

Recap

Last sprint

- Deploy Jaeger into test cluster
- Flesh out Throttling FS
- Integrate Tracing FS with Jaeger
- Integrate Faulty fs with OTEL SDK

This sprint

- Deploy Grafana into test cluster
- Package and manage deployment with Helm/Helmfile
- Implement metric collection
- Integrate Faulty fs with Jaeger
- Fix TokenBucket implementation

What is Grafana

Grafana

Dashboard anything. Observe everything.

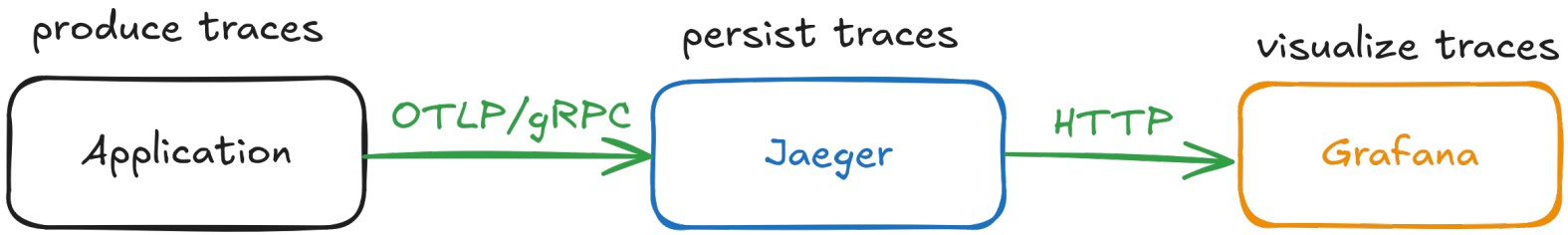
Why Grafana

Jaeger

Jaeger handles the ingestion and storage of traces, but lacks the ability to visualize and analyze them

Grafana

Grafana connects to Jaeger and augments it with data aggregation and visualization capabilities



What is Helm

Helm

The package manager for Kubernetes

Benefits of Helm

- Encapsulate complex applications into `Charts`
- Share `Charts` as static files or container images
- Handle future updates of applications

Charts

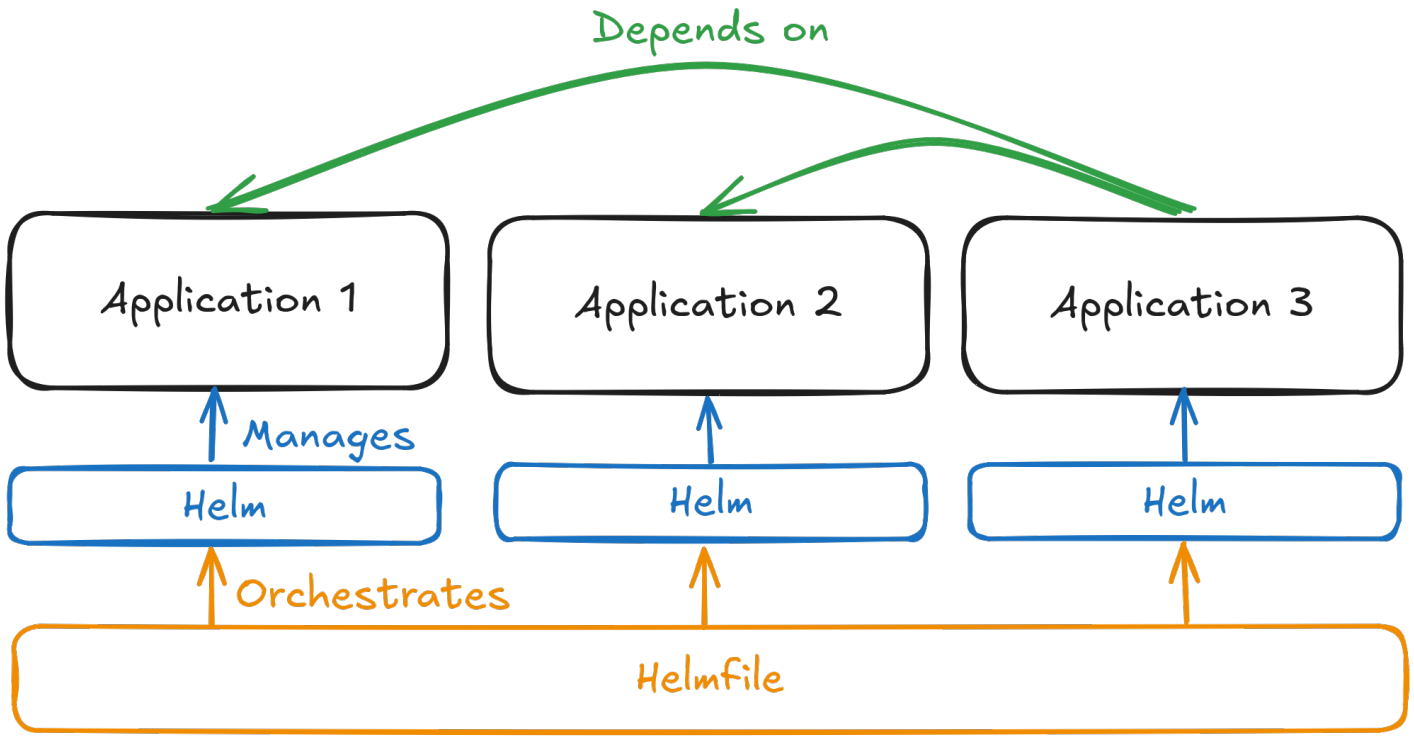
`Charts` are collections of kubernetes manifest templates that can be rendered by Helm

What is Helmfile

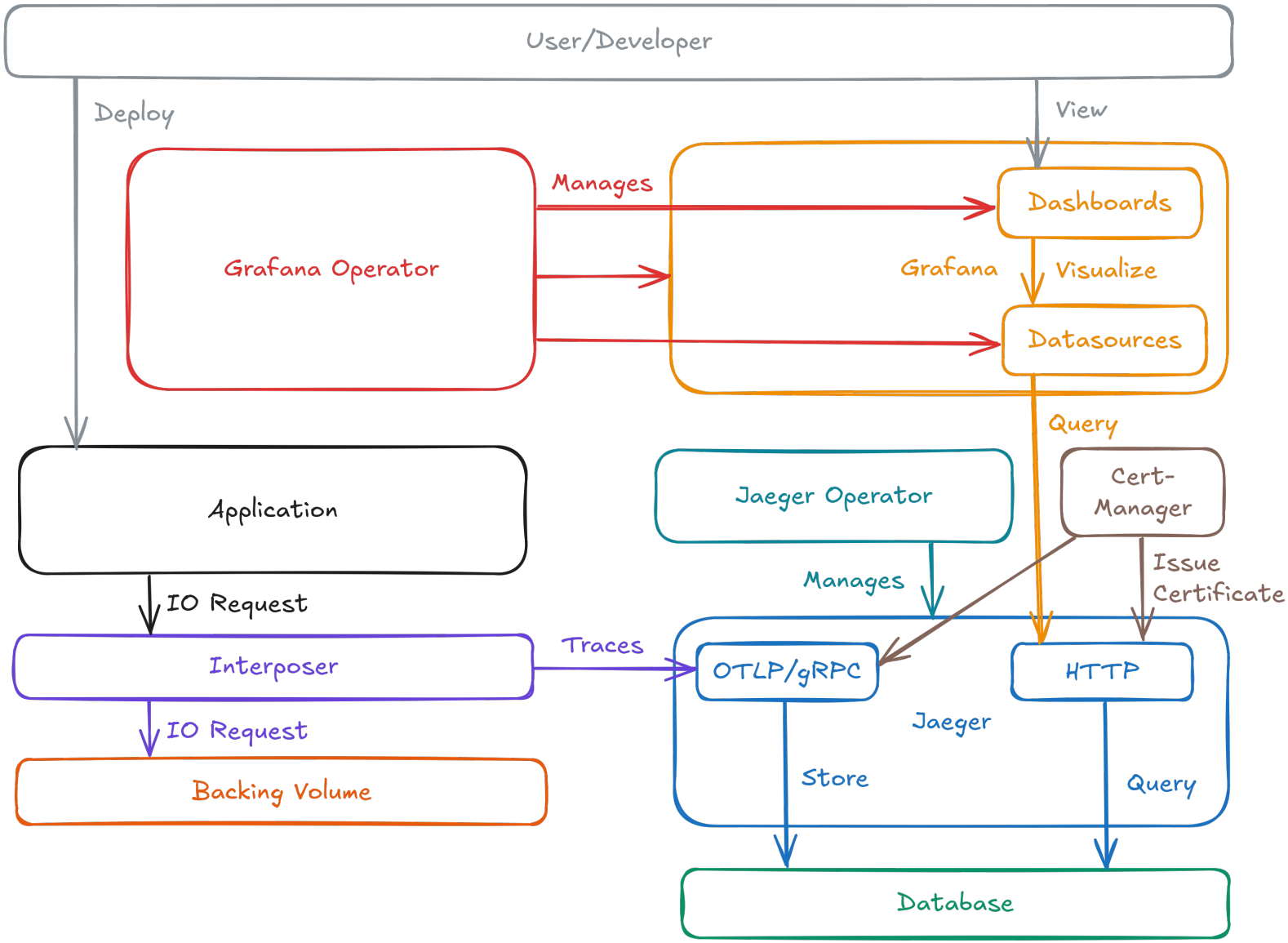
Helmfile

Helmfile is a declarative spec for deploying helm charts

Helm manages individual applications, Helmfile manages a bunch of interdependent applications



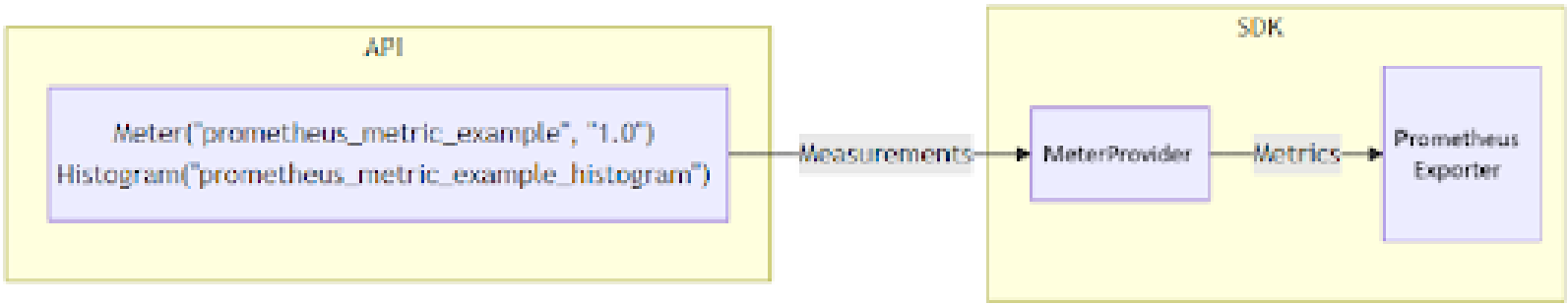
Overview


User/Developer
Deploy
View
Grafana Operator
Manages
Dashboards
Grafana
Visualize
Datasources
Query
Application
Jaeger Operator
Cert-Manager
Manages
Issue Certificate
IO Request
Interposer
Traces
OTLP/gRPC
HTTP
Jaeger
IO Request
Store
Query
Backing Volume
Database

Metric Collection

OpenTelemetry's Metrics Model

- The MeterProvider provides access to Meters.
- A Meter creates Instruments.
- An Instrument reports Measurements.



Metric Collection

Example Instrument Types

- Counter
- Histogram
- Up-down counter

FUSE File System

Gather quantitative and aggregate data on the underlying file system.

- The total number of bytes read and written
- Latency distribution
- The net number of directories created

Implement Jaeger into FaultyFs

Dockerize OTLP exporter install from Otel build

- Previously dockerized installation of Otel libraries
- Need OTLP exporter to export to Jaeger
- Having Problems

Improve Otel instrumentation

- Previously used Otel logging sdk
- Desire to share Jaeger backend
- Implement tracer and spans

Implement Jaeger

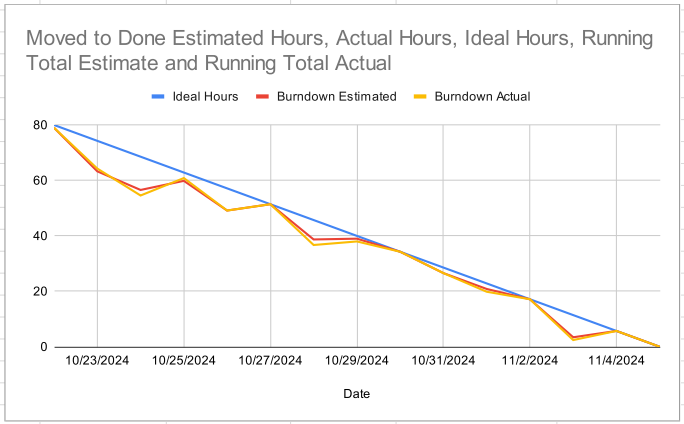
- Dockerize relevant dependencies
- Instrument Jaeger endpoint

Sprint Review

	Title	Estimate Hours	Actual Hours	DoneDate
1	Integrate metric collection layer into the interposer binary #79	2	3	Nov 1, 2024
2	Package the manifests into a helm chart #86	3	2	Oct 25, 2024
3	Deploy grafana into test cluster #88	1	1	Oct 22, 2024
4	Provision grafana with jaeger datasource #89	1	1	Oct 23, 2024
5	Deploy jaeger with prodution strategy to mitigate performance bottleneck #90	2	2	Oct 24, 2024
6	Install cert-manager with helm #91	2	1	Oct 26, 2024
7	Install jaeger-operator with helm #92	2	1	Oct 26, 2024
8	Wait for jaeger operator to be ready #93	1	1	Oct 26, 2024
9	Package csi plugin into helm chart #95	4	6	Oct 28, 2024
10	Use grafana operator in test cluster #96	1	1	Oct 28, 2024
11	Provision grafana datasource with operator #97	1	2	Oct 29, 2024
12	Export metrics to Prometheus backend #98	3	5	Oct 26, 2024
13	Add metric for total number of bytes read/written #101	2	3	Nov 3, 2024
14	Add latency histogram for read/write operations #100	2	2	Oct 31, 2024
15	Add up/down counter for the number of directories created/removed. #99	2	2	Oct 28, 2024
16	Rework TokenBucket implementation #102	10	12	Oct 24, 2024
17	Debug the deadlocks in TokenBucket implementation #103	6	5	Oct 23, 2024
18	Use meson wrap to drop dependency on system fuse library #104	4	4	Oct 23, 2024
19	Prepare for presentation #105	6	6	Nov 3, 2024

Future Plans

- Test the installation in alternative k8s environments
- Grafana dashboard for the metrics
- Make filesystem attributes configurable
- Post processing resolving file path from inode number
- Create parent spans for multiple operations

Burndown

Title	Date	Moved to Done Estimated Hours	Moved to Done Actual Hours	Ideal Hours	Burndown Estimated	Burndown Actual	Team #	Hours/Sprint
Deploy grafana into test cluster	10/22/2024	1	1	80	79	79	*4*	*20*
TokenBucket debug and meson wrap	10/23/2024	11	10	74	63	64		
TokenBucket rework	10/24/2024	12	14	69	57	55		
Package the manifests into a helm chart	10/25/2024	3	2	63	60	61		
Export metrics to Prometheus backend	10/26/2024	8	8	57	49	49		
	10/27/2024	0	0	51	51	51		
Count directories	10/28/2024	7	9	46	39	37		
Provision grafana datasource with operator	10/29/2024	1	2	40	39	38		
	10/30/2024	0	0	34	34	34		
Add latency histogram for read/write operations	10/31/2024	2	2	29	27	27		
Integrate metric collection layer into the interposer binary	11/1/2024	2	3	23	21	20		
	11/2/2024	0	0	17	17	17		
Package the manifests into a helm chart	11/3/2024	8	9	11	3	2		
	11/4/2024	0	0	6	6	6		
	11/5/2024	0	0	0	0	0