



From k9s to OpenTelemetry: A guide to observe (Java apps) in Kubernetes

Matthias Häußler
Chief Technologist, Novatec

#whoami



matthiashaeussler



@maeddes



Chief Technologist



Training &
Certification



Hochschule
für Technik
Stuttgart

ESSLINGEN
UNIVERSITY

For people and technology.

Distributed Systems



<https://speakerdeck.com/maeddes/whats-going-on-in-my-cluster>



observability



java

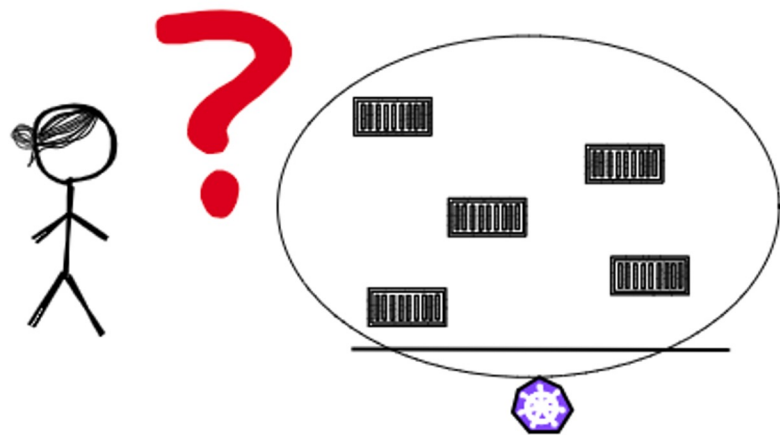
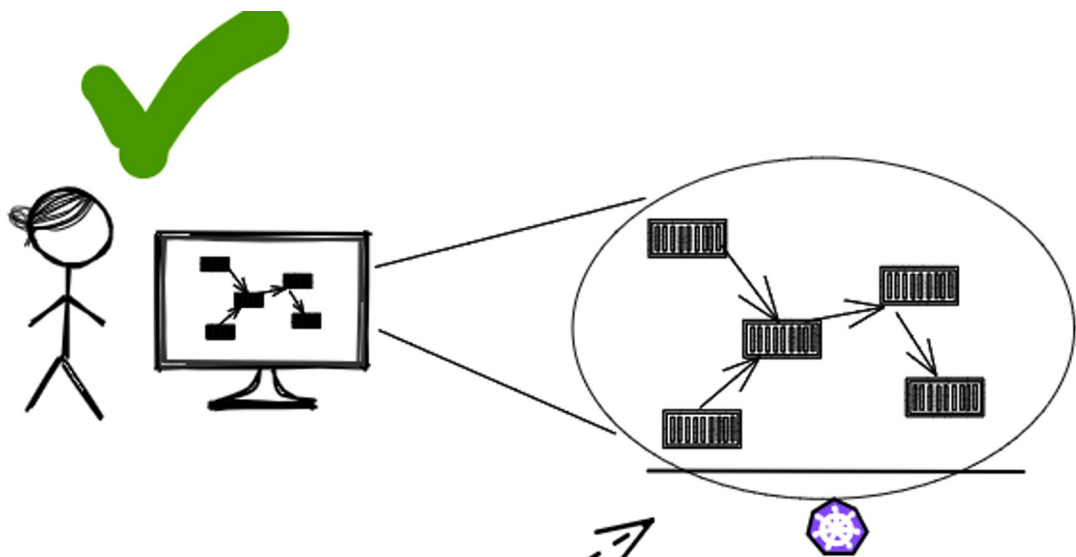


kubernetes

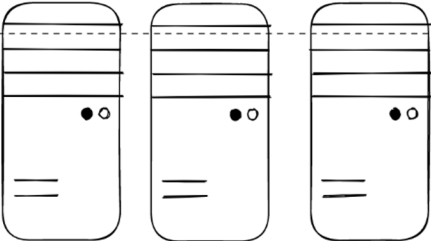
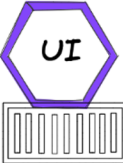
Agenda

1. Intro
2. Kubernetes API
3. Prometheus & Grafana
4. Service Mesh
 - Sidecar-based
 - eBPF-based
5. Application-based

Why?

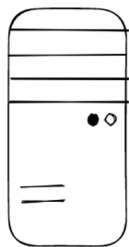


application/workloads

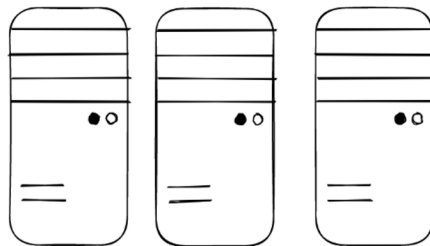


nodes

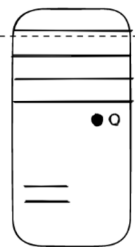
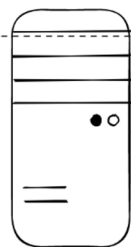
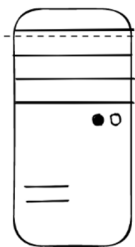
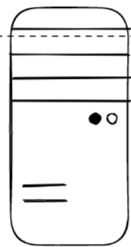
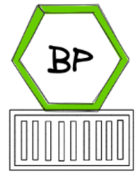
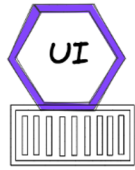
control plane



worker nodes



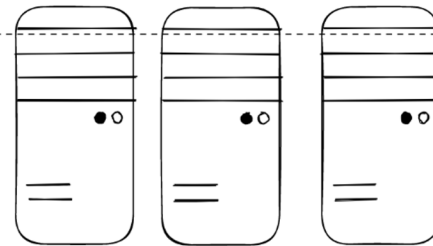
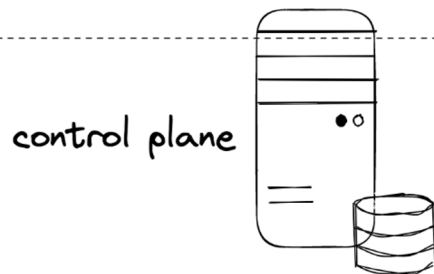
application/workloads



worker nodes

pod
replicaset
ingress
service
deployment

API Objects



worker nodes

replicaset

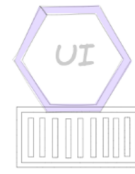
pod

ingress

service

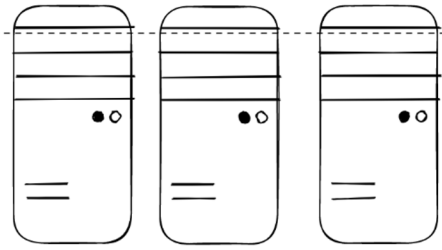
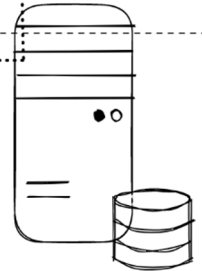
deployment

API Server



API Server

control plane



worker nodes



pod replicaset

pod

ingress

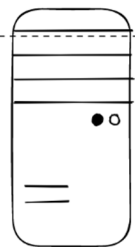
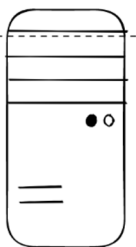
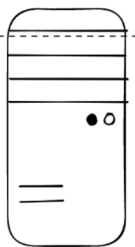
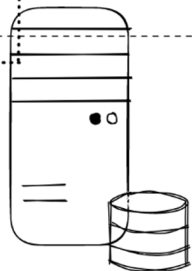
service

deployment



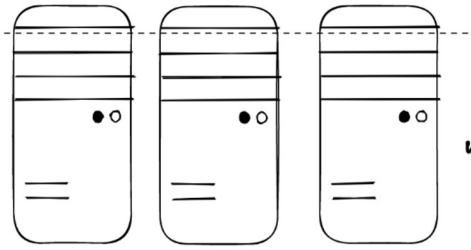
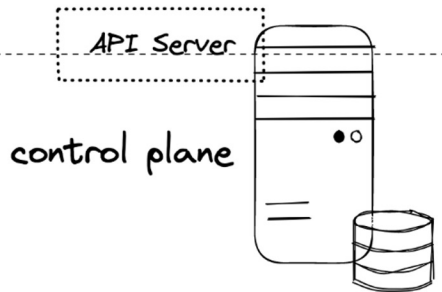
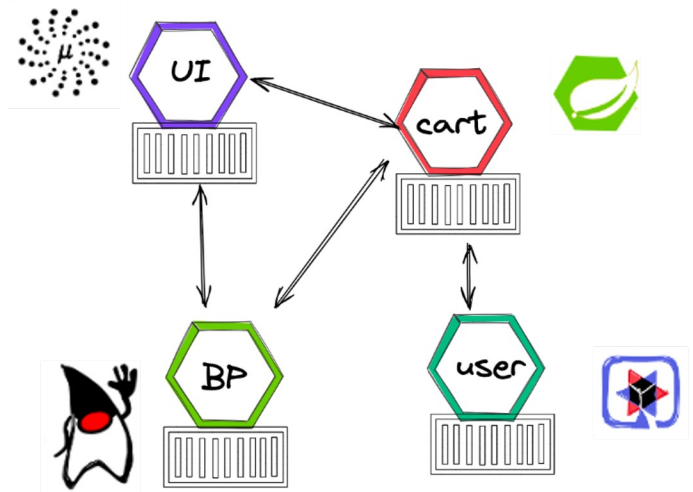
API Server

control plane

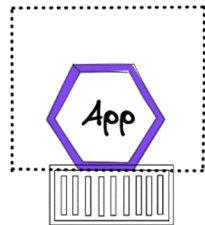


worker nodes

replicaset
pod
ingress
service
deployment



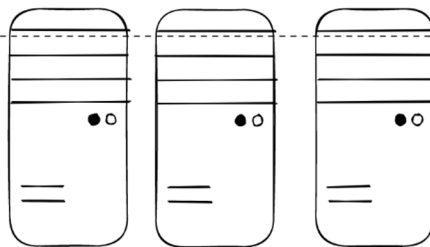
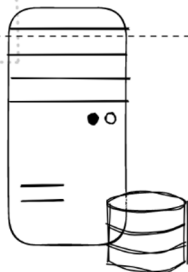
worker nodes



application

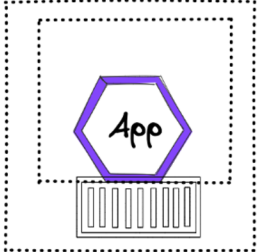


control plane



worker nodes

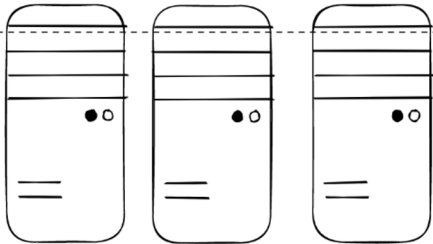
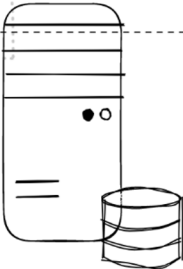




application
container

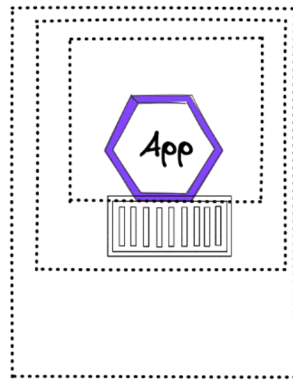


control plane



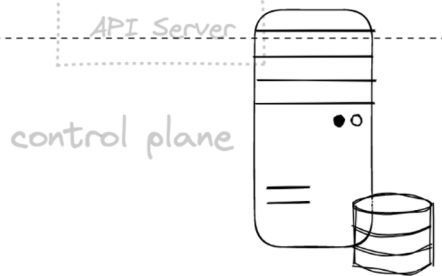
worker nodes



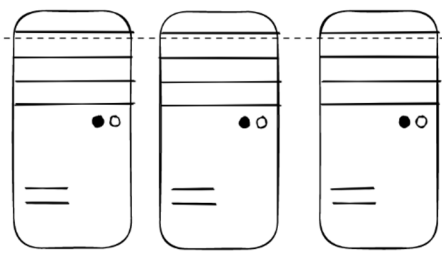


application
container
pod

API Server

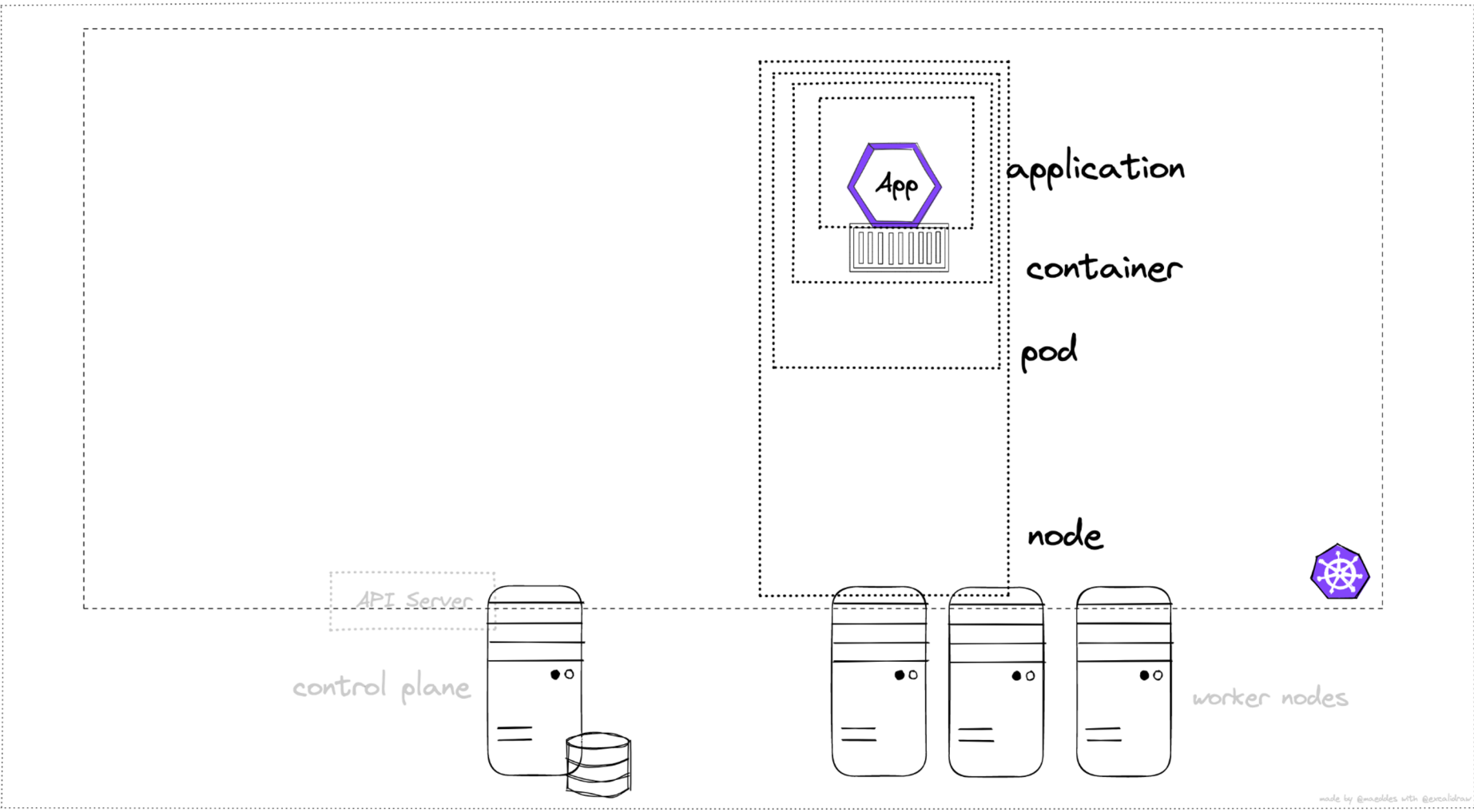


control plane



worker nodes





observe



application

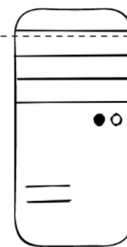
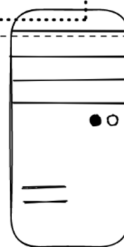
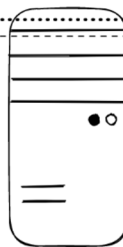
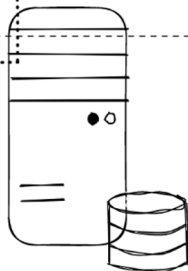
container

pod

node

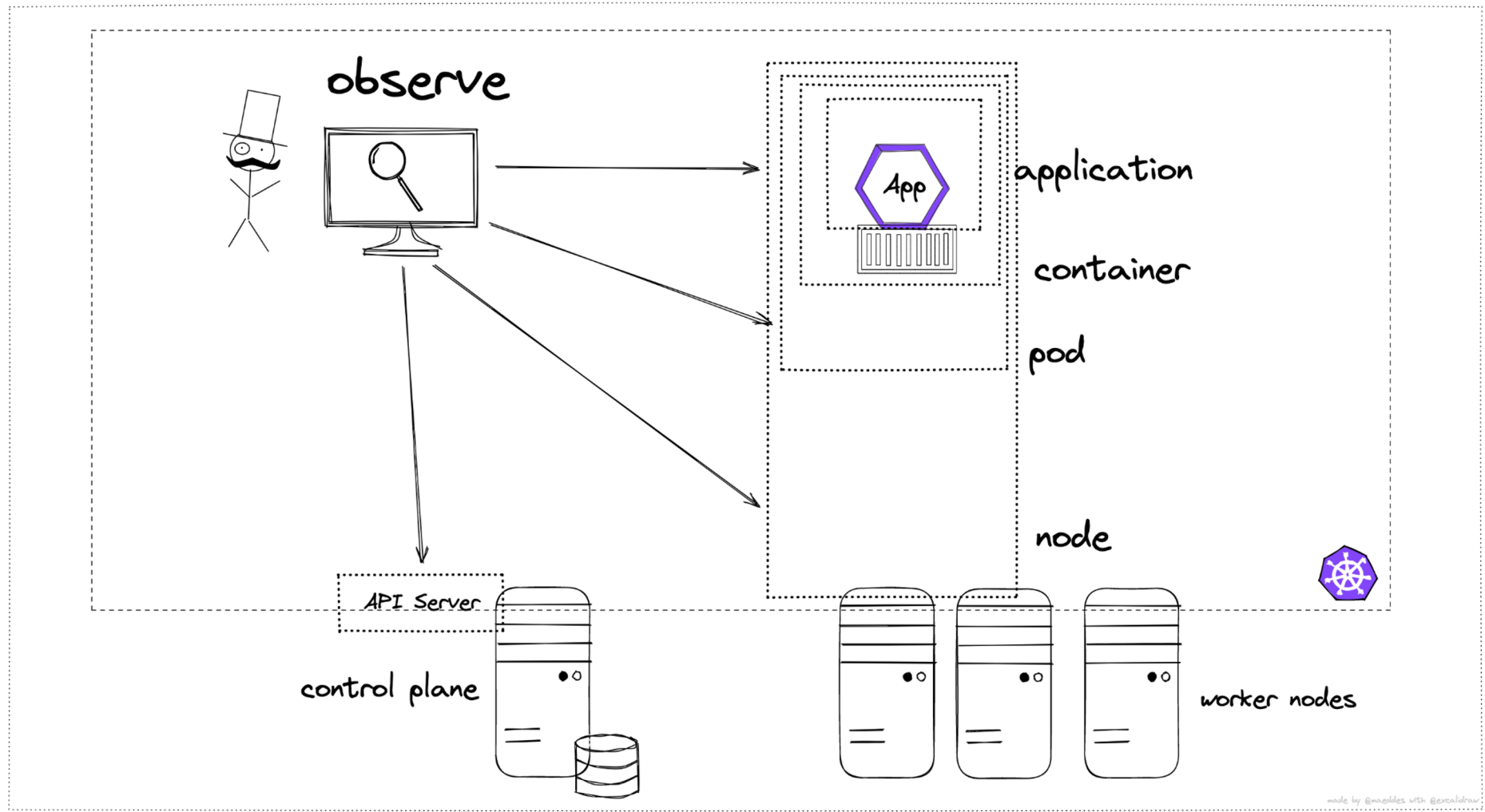
API Server

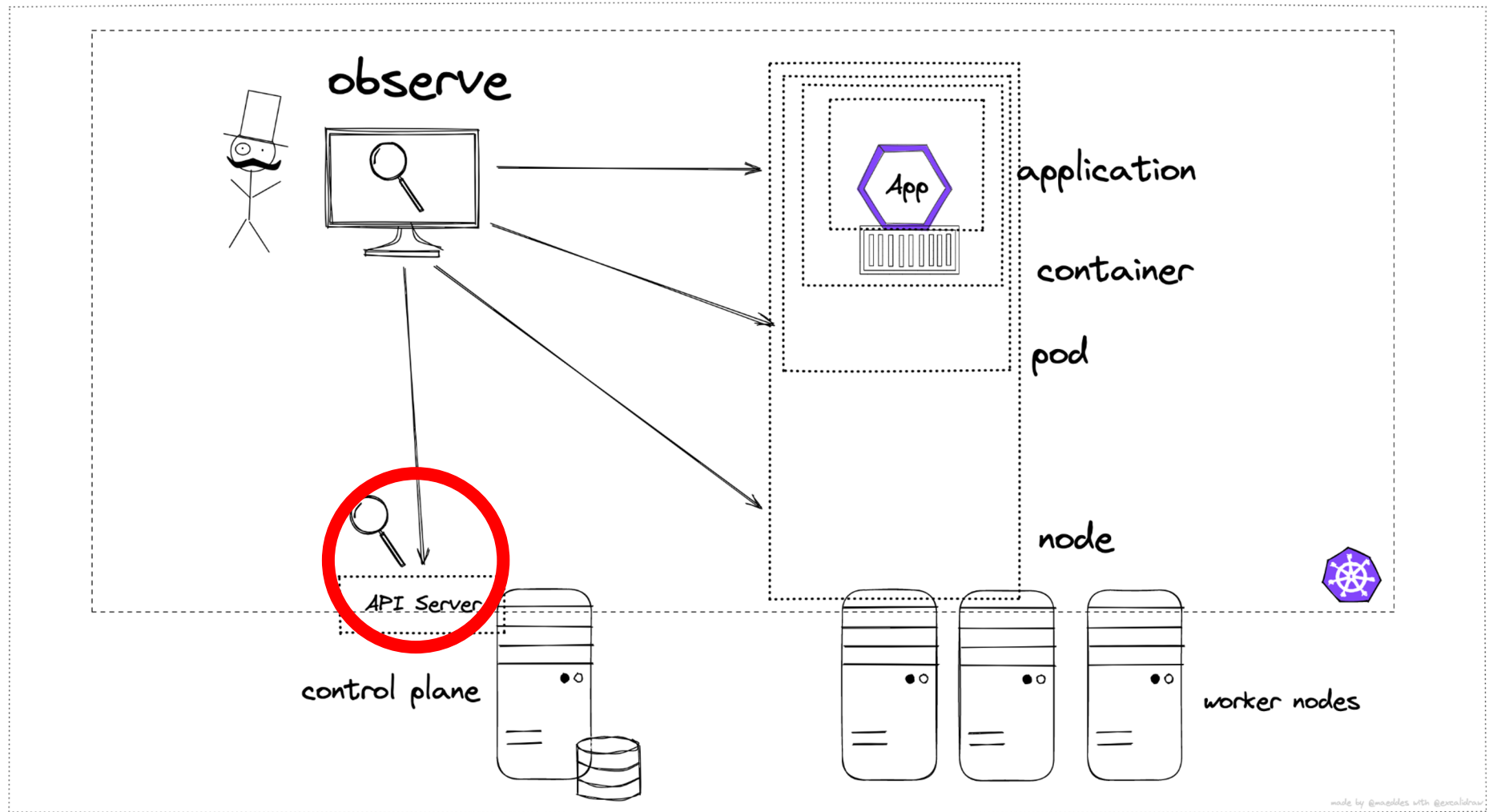
control plane

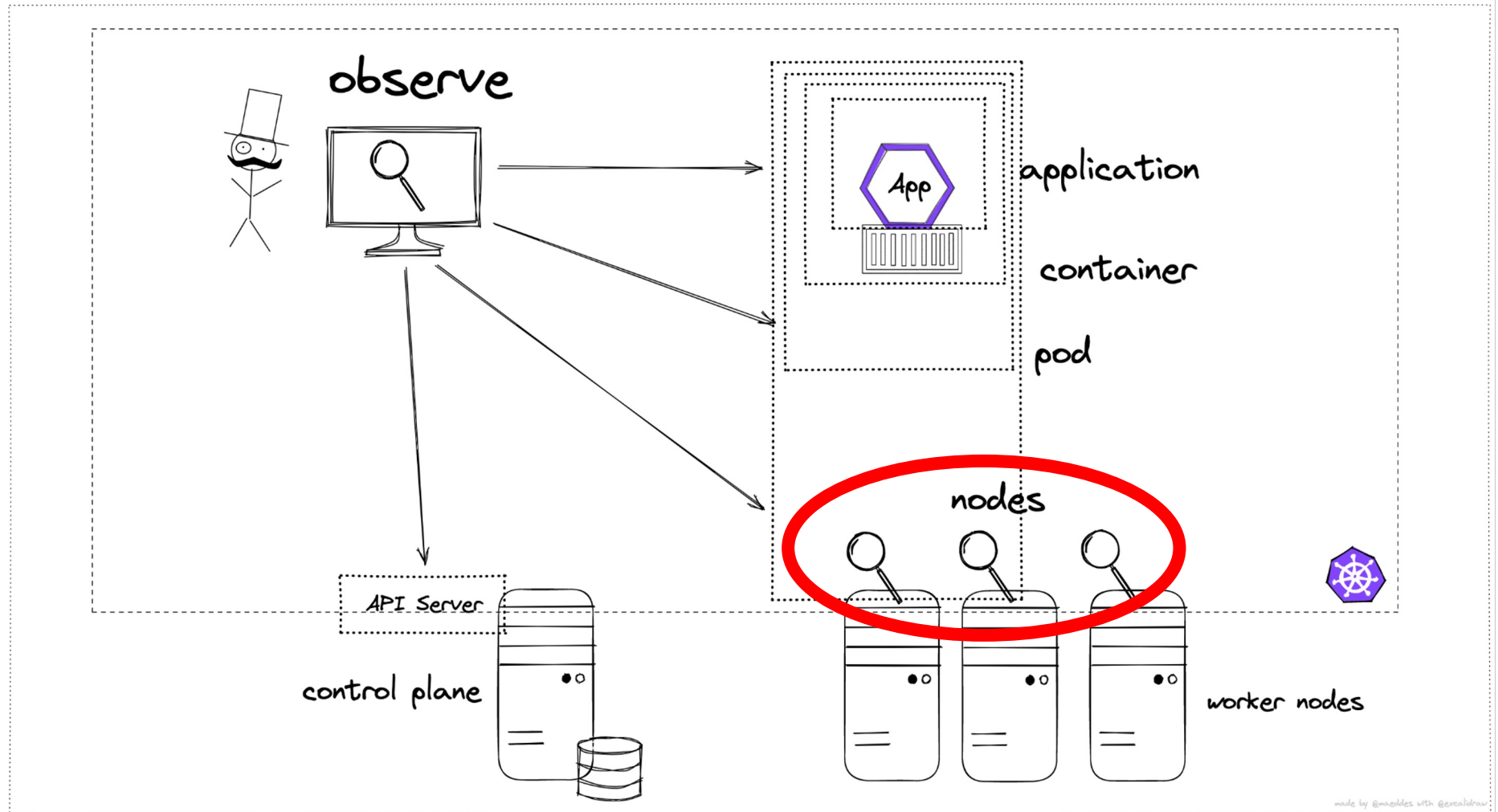


worker nodes

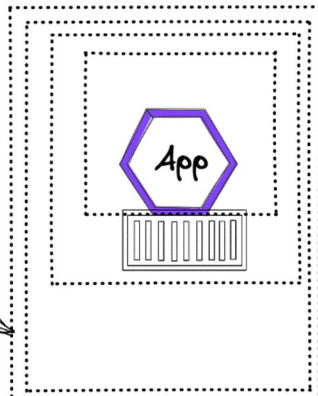






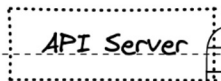


observe



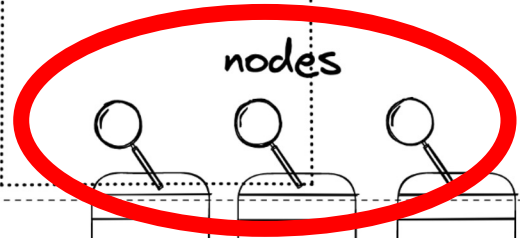
application container

pod



API Server

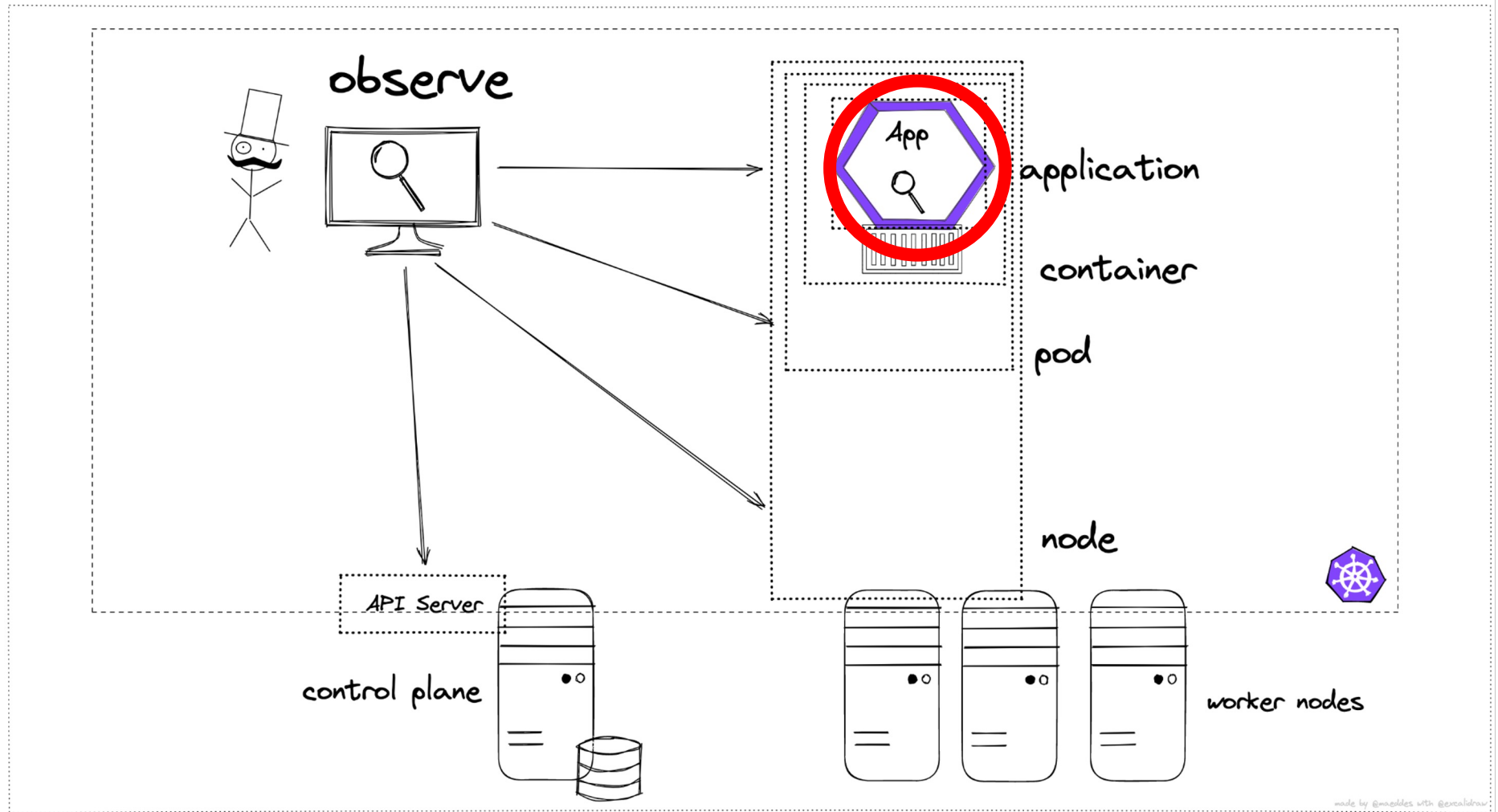
control plane

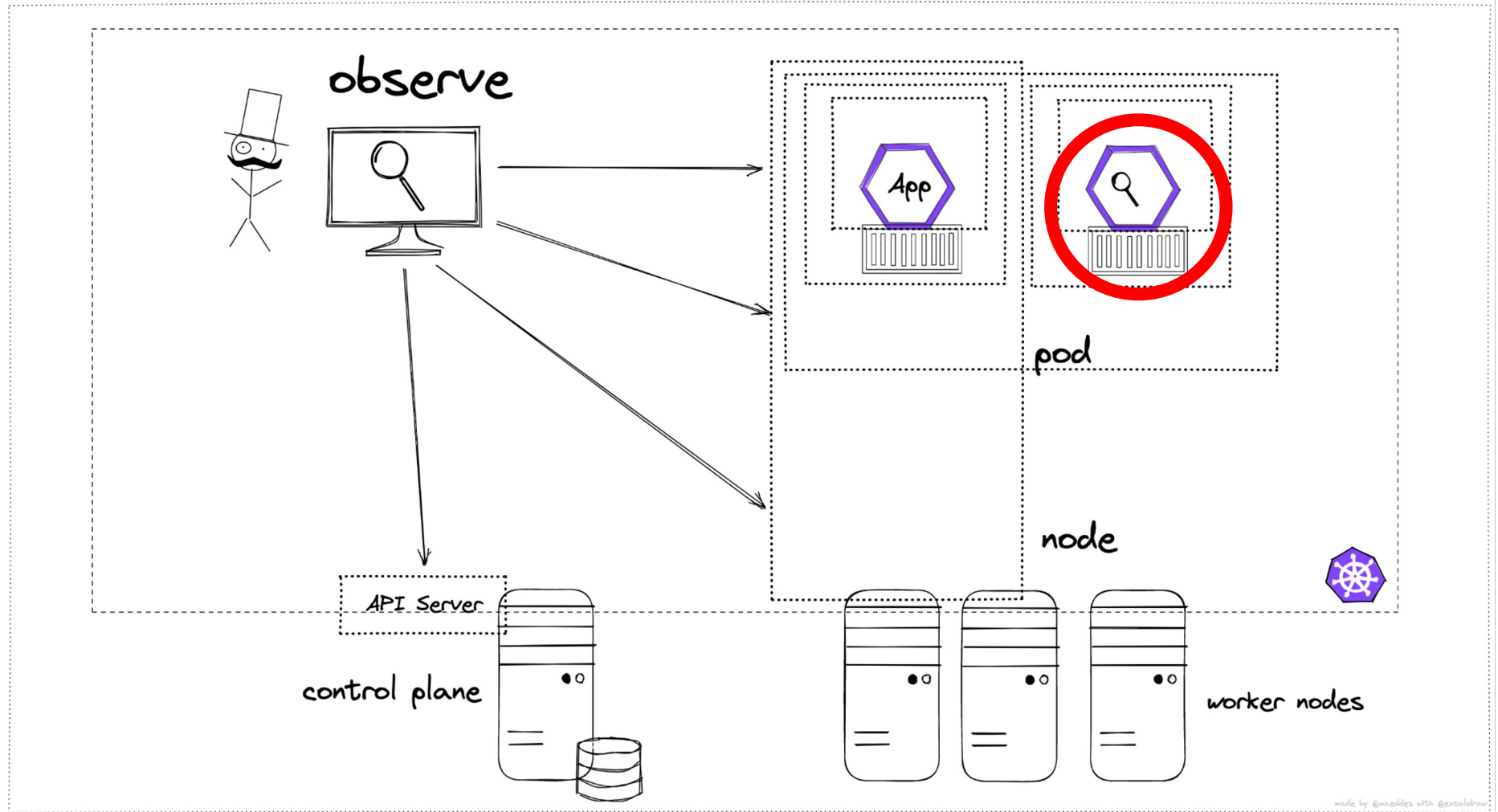


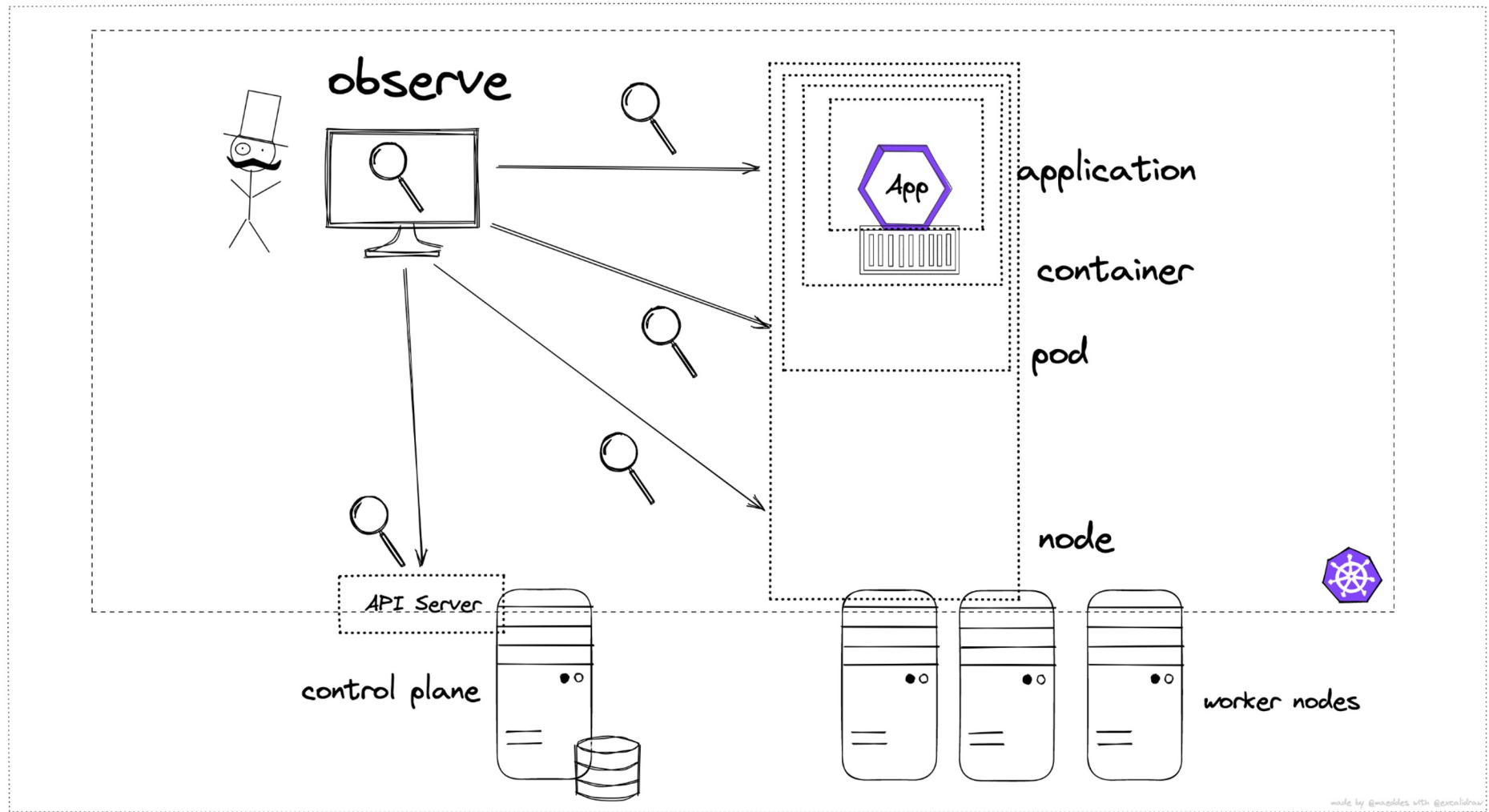
nodes

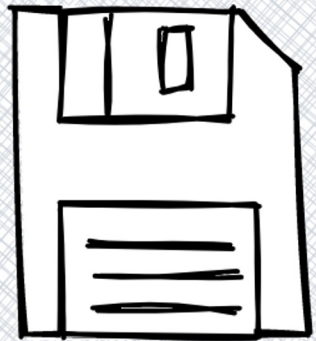
worker nodes











tools?

Application Definition & Image Build

Continuous Integration & Delivery

Scheduling & Orchestration

Coordination & Service Discovery

Remote Procedure Call

Service Proxy

API Gateway

Service Mesh

Cloud Native Storage

Container Runtime

Cloud Native Network

Automation & Configuration

Container Registry

Security & Compliance

Key Management

Provisioning

Orchestration & Management

Alpha Definition and Development

Platform

Certified Kubernetes - Distribution

Certified Kubernetes - Hosted

Certified Kubernetes - CNO

Certified Kubernetes - Installer

PaaS/Container Service

Observability and Analysis

Monitoring

Logging

Serverless

Members

CD Foundation Landscape

Detail View – Observability Technologies

Monitoring

 Prometheus CNCF Graduated	 cortex CNCF Incubating	 OPENMETRICS CNCF Incubating	 Thanos CNCF Incubating	 Alibaba Cloud Application Real-Time Monitoring Service	 Amazon CloudWatch	 APFDYNAMICS	 Application Real-Time Monitoring Service					
 Aternity	 Azure Monitor	 beats	 bluemeter	 botkube	 catchpoint	 centreon	 checkmk	 chronosphere	 云控至	 DATADOG	 DspFlow	
 dynatrace	 epsagon	 falco	 FALCON	 FONIO	 foresight	 Google Stackdriver	 Gradle	 Grafana	 Mimir	 Phlare	 graphite	
 观澜云 GIGAMON	 HEADLAMP	 Honeybadger	 HUBBLE	 icinga	 Influxdata	 INSPEKTOR BUDGET	 INSTANA	 IRONdb	 KOSOFT	 KUBERNETES	 kiali	
 kuberhealth	 Last9	 LeanIX	 LINDb	 LogicMonitor	 logz.io	 MG	 mackerel	 MICROMETER	 Nagios	 NETDATA	 Nëtis	
 new relic	 NexClipper	 Nightingale	 NODESOURCE	 OpenTracing	 opstrace	 OverOps	 PIXIE	 Promscale	 replex	 ROOKOUT	 Sensu	
 SENTRY	 SIDEKICK	 SignalFx	 Skooner	 SOSIOVIO	 StackState	 sysdig	 腾讯云	 TRaaS BOS	 trickster	 VECTOR By Tencent	 VICTORIA METRICS	
 virtasant	 vmware Aria Operations for Applications	 CloudHealth	 weave scope	 WhatTap	 ZABBIX							

Logging

 fluentd CNCF Graduated	 Alibaba Cloud Log Service	 DataSet	 elastic	 Grafana loki	 graylog	 humio	 Loggie	 LOGGLY	 LOGIQ	 logstash
 mezmo	 OpenSearch	 Pandora	 parseable	 日志易 ningyong.com	 sematext	 splunk>	 sumo logic	 腾讯云 Log Explorer	 TRINK	



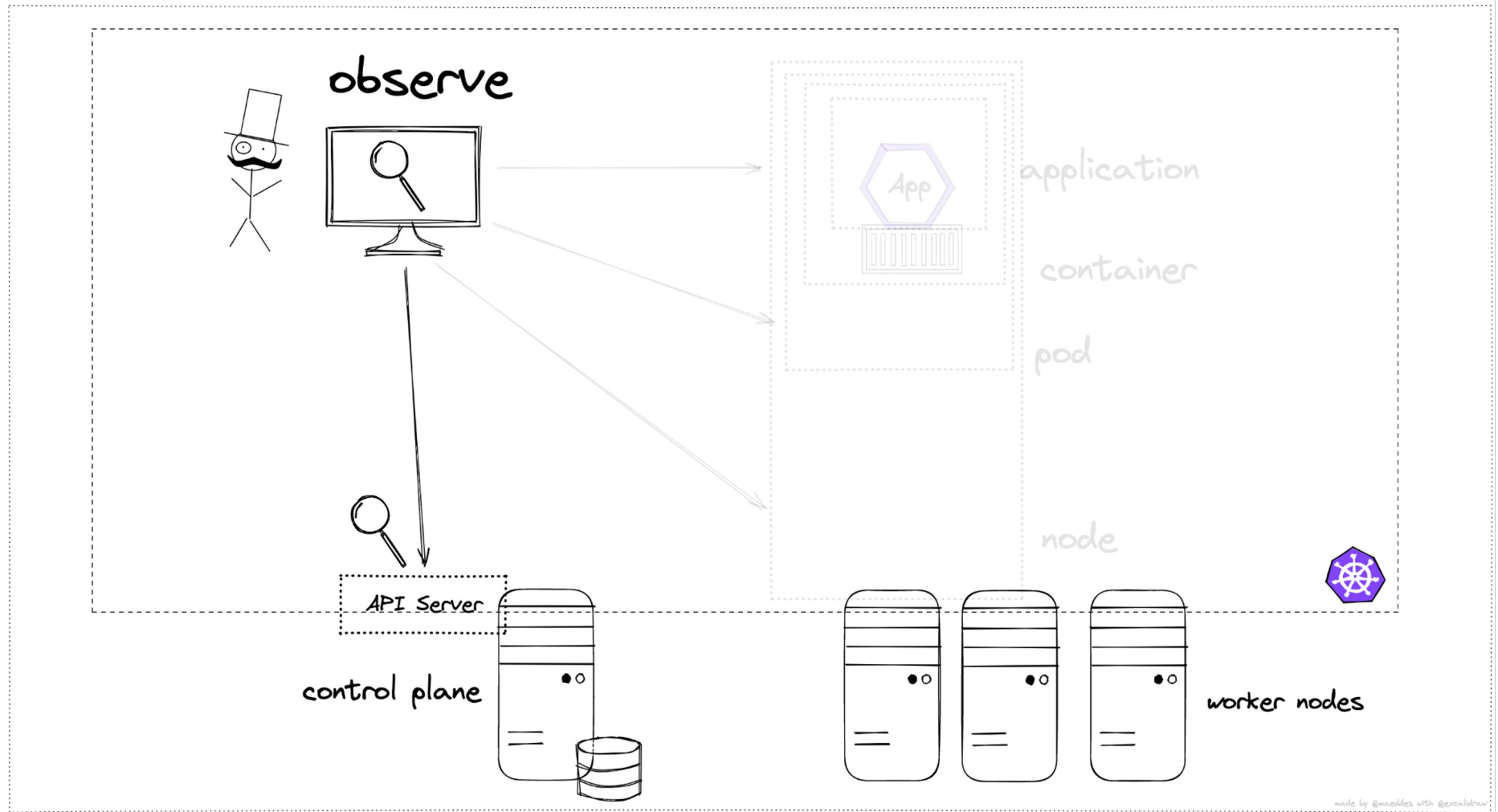
Tracing

 JAEGER CNCF Graduated	 OpenTelemetry CNCF Incubating	 Aspecto	 EaseAgent	 elastic apm	 Grafana Tempo	 Holoos	 karymbus	 Lightstep	 OPENTRACING
 PINPOINT	 SkyWalking	 SOFATracer	 Spring Cloud Sleuth	 TelemetryDB	 Teltrace	 tracetest	 ZIPKIN		





Kubernetes API



observe

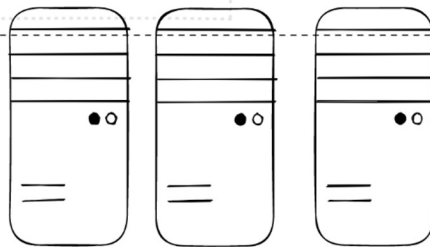
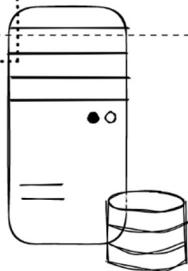


```
kubectl get ...  
kubectl describe ...  
kubectl logs ...  
kubectl debug ...  
kubectl exec ...
```



API Server

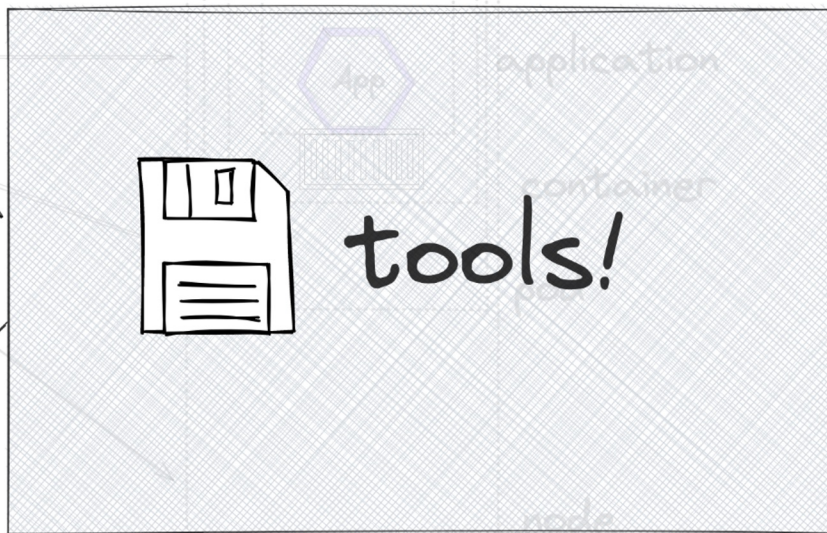
control plane



worker nodes

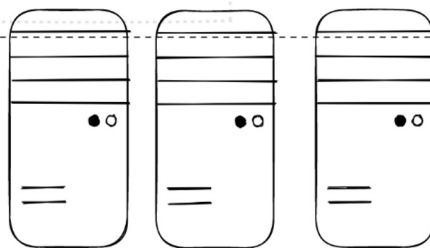
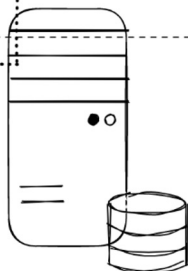


observe



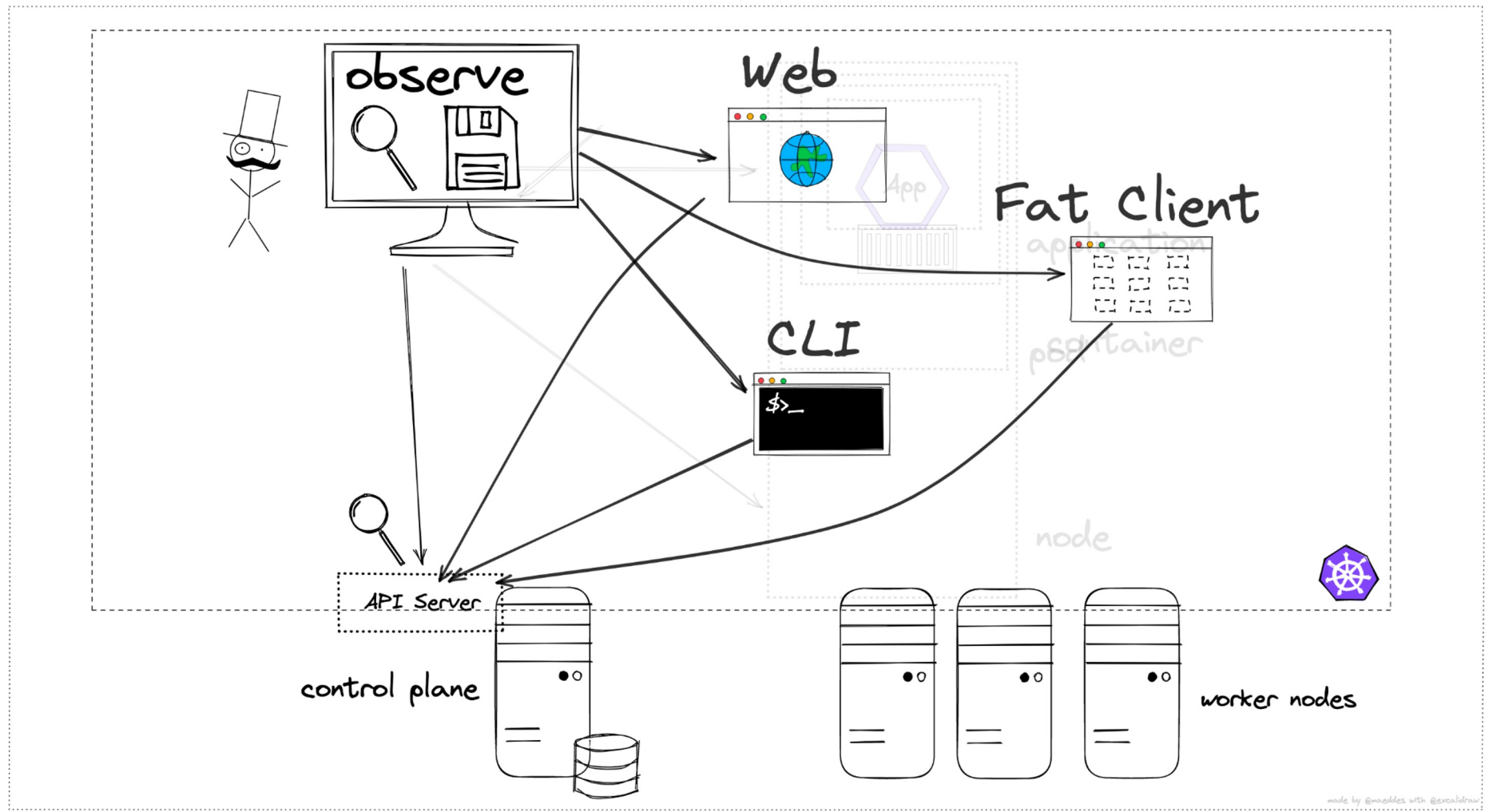
API Server

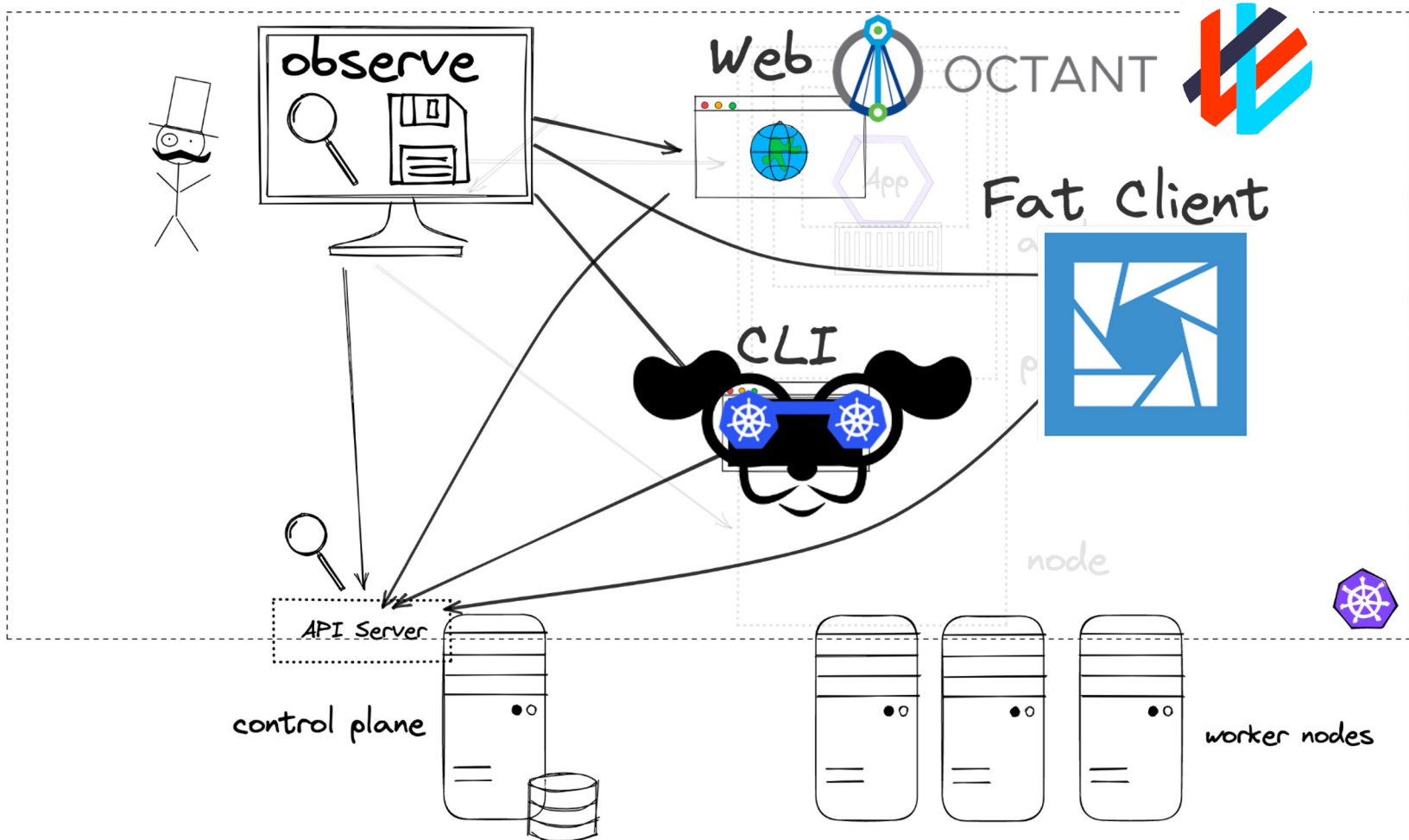
control plane



worker nodes





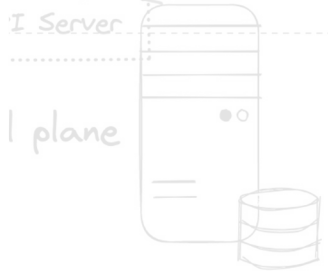




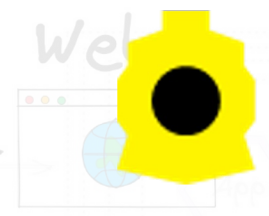
Deprecated & Discontinued



good OSS friend



"Commercial"



HEADLAMP

Fat Client

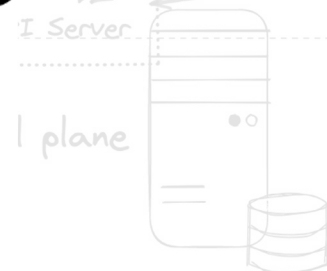


new kids



Skooner

good OSS friend



worker nodes

```

Context: mhsdemo          <0> all          <ctrl-d> Delete          <shift-f> Port-Forward
Cluster: mhsdemo         <1> istio-system      <d> Describe           <r> Restart
User:   clusterUser_mhsdemo_mhsdemo <2> spring-petclinic <e> Edit                <s> Scale
K9s Rev: v0.26.0 ⚡ v0.26.3 <3> default          <?> Help              <y> YAML
K8s Rev: v1.21.9
CPU:    24%
MEM:    9%

```



Deployments (all) [27]

NAMESPACE↑	NAME	READY	UP-TO-DATE	AVAILABLE	AGE
argocd	argocd-applicationset-controller	1/1	1	1	109d
argocd	argocd-dex-server	1/1	1	1	109d
argocd	argocd-notifications-controller	1/1	1	1	109d
argocd	argocd-redis	1/1	1	1	109d
argocd	argocd-repo-server	1/1	1	1	109d
argocd	argocd-server	1/1	1	1	109d
default	aks-helloworld-one	1/1	1	1	150d
default	aks-helloworld-two	1/1	1	1	150d
default	ingress-nginx-controller	1/1	1	1	150d
hse	hse-deploy	5/5	5	5	69d
istio-system	grafana	1/1	1	1	110d
istio-system	istio-ingressgateway	1/1	1	1	110d
istio-system	istiod	1/1	1	1	110d
istio-system	jaeger	1/1	1	1	110d
istio-system	kiali	1/1	1	1	110d
istio-system	prometheus	1/1	1	1	110d
kube-system	coredns	2/2	2	2	159d
kube-system	coredns-autoscaler	1/1	1	1	159d
kube-system	konnectivity-agent	2/2	2	2	104d
kube-system	metrics-server	1/1	1	1	159d
mhsdemo-ns-jenkins	mhsdemo-ns-jenkins-nginx-class-ingress-nginx-controller	1/1	1	1	150d
spring-petclinic	api-gateway	1/1	1	1	101d
spring-petclinic	customers-service	0/1	1	0	101d
spring-petclinic	vets-service	0/1	1	0	101d
spring-petclinic	vets-service-v02	0/1	1	0	100d
spring-petclinic	visits-service	0/1	1	0	101d
spring-petclinic	wavefront-proxy	0/1	1	0	101d

<deployment>



Namespace Overview

Workloads

Overview

Cron Jobs

Daemon Sets

Deployments

Jobs

Pods

Replica Sets

Replication Controllers

Stateful Sets

Discovery and Load Balancing

Config and Storage

Custom Resources

RBAC

Events

Namespace Overview

Namespace module shows all resources related to currently selected namespace Use dropdown at the top to change the selected namespace

Pods

Pods

	Name	Labels	Ready	Phase	Status	Restarts	Node	Age
⋮	api-gateway-79545bcd86-djptt	app:api-gateway security.istio.io/tlsMo... 2+	2/2	Running	Running	0	aks-nodepool1-26169201-vmss00000r	6m
⋮	customers-db-mysql-0	app.kubernetes.io/... app.kubernetes.io/... 6+	2/2	Running	Running	0	aks-nodepool1-26169201-vmss00000r	6m
⋮	customers-service-7b45c4f5f7-nxhsw	app:customers-ser... security.istio.io/tls... 2+	2/2	Running	Running	1	aks-nodepool1-26169201-vmss00000r	6m
⋮	vets-db-mysql-0	app.kubernetes.io/... app.kubernetes.io/... 6+	2/2	Running	Running	0	aks-nodepool1-26169201-vmss00000r	6m
⋮	vets-service-79dbb4bd8b-s85dz	app:vets-service security.istio.io/tlsMo... 3+	2/2	Running	Running	1	aks-nodepool1-26169201-vmss00000r	6m
⋮	vets-service-v02-5c4f69749f-dg7dm	app:vets-service security.istio.io/tlsMo... 3+	2/2	Running	Running	1	aks-nodepool1-26169201-vmss00000r	6m
⋮	visits-db-mysql-0	app.kubernetes.io/... app.kubernetes.io/... 6+	2/2	Running	Running	0	aks-nodepool1-26169201-vmss00000r	6m
⋮	visits-service-6d59cc5f9f-v6pt9	app:visits-service security.istio.io/tlsMo... 2+	2/2	Running	Running	2	aks-nodepool1-26169201-vmss00000r	6m
⋮	wavefront-proxy-6b58d6f557-64bcm	app:wavefront-pro... security.istio.io/tls... 2+	1/2	Running	CrashLoopBackOff	5	aks-nodepool1-26169201-vmss00000r	6m

Items per page 10 1 - 9 of 9 items



Lens mhsdemo

Cluster Nodes Workloads

Overview Pods Deployments DaemonSets StatefulSets ReplicaSets Jobs CronJobs

Namespaces: default, spring- Search...

Pods 12 items

Name	Namespace	Containers	Restarts	Controlled By	Node	QoS	Age	Status
aks-helloworld-one-56c7b8d79d-tvm5s	default	■	0	ReplicaSet	aks-nodepool1-2616f	BestEffort	59d	Running
aks-helloworld-two-58bbb47f58-hftb8	default	■	0	ReplicaSet	aks-nodepool1-2616f	BestEffort	59d	Running
api-gateway-79545bcd86-djptt	spring-petclinic	■ ■ ■	0	ReplicaSet	aks-nodepool1-2616f	Burstable	9m31s	Running
customers-db-mysql-0	spring-petclinic	■ ■ ■	0	StatefulSet	aks-nodepool1-2616f	Burstable	9m33s	Running
customers-service-7b45c4f5f7-nxhsw	spring-petclinic	■ ■ ■	1	ReplicaSet	aks-nodepool1-2616f	Burstable	9m31s	Running
ingress-nginx-controller-756f546d89-6xb...	default	■	0	ReplicaSet	aks-nodepool1-2616f	Burstable	59d	Running
vets-db-mysql-0	spring-petclinic	■ ■ ■	0	StatefulSet	aks-nodepool1-2616f	Burstable	9m33s	Running
vets-service-79dbb4bd8b-s85dz	spring-petclinic	■ ■ ■	1	ReplicaSet	aks-nodepool1-2616f	Burstable	9m32s	Running
vets-service-v02-5c4f69749f-dg7dm	spring-petclinic	■ ■ ■	1	ReplicaSet	aks-nodepool1-2616f	Burstable	9m32s	Running
visits-db-mysql-0	spring-petclinic	■ ■ ■	0	StatefulSet	aks-nodepool1-2616f	Burstable	9m33s	Running
visits-service-6d59cc5f9f-v6pt9	spring-petclinic	■ ■ ■	2	ReplicaSet	aks-nodepool1-2616f	Burstable	9m31s	Running
wavefront-proxy-6b58d6f557-64bcm	spring-petclinic	■ ■ ■	6	ReplicaSet	aks-nodepool1-2616f	Burstable	9m32s	CrashLoopB...

CLUSTER

Namespaces

Nodes

CRDs

WORKLOADS

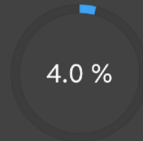
STORAGE

NETWORK

SECURITY

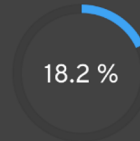
CONFIGURATION

CPU Usage



0.63 / 16 units

Memory Usage



22.82 / 125.61 GB

Pods



Events Warnings



Type	Name	Namespace	Reason	Age
Pod	otel-demo-featureflagservice-84b8575847-mp87z	otel	Pulled	7m
Ingress	otel	otel	Sync	46m
Pod	ingress-nginx-controller-6b8bfd7f69-fngth	ingress-nginx	RELOAD	1h
Pod	otel-demo-featureflagservice-84b8575847-mp87z	otel	BackOff	2h
Pod	wavefront-proxy-7f4b865d59-z6745	spring-petclinic	BackOff	5h

Cluster Overview

 type to filter 


CLUSTER



NODES



NAMESPACES



WORKLOADS



STORAGE



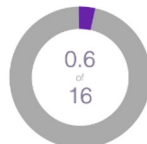
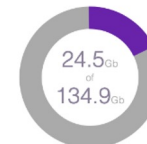
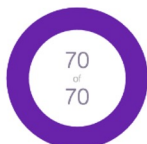
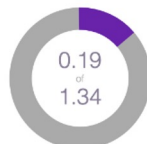
ACCOUNTS








PROFILE



APPLY


NODES
READY VS ALL

NODE CPU USE
USED VS AVAILABLE

NODE RAM USE
USED VS AVAILABLE

PODS
READY VS REQUESTED

POD CPU USE
ACTUAL VS RESERVED

POD RAM USE
ACTUAL VS RESERVED

TYPE	NAME	TIME ^	REASON	EVENT
 POD	ingress-nginx:ingress-nginx-controller-6b8bfd7169-fngth	1 h	RELOAD	NGINX reload triggered due to a change in configuration
 INGRESS	otel:otel	55 m	Sync	Scheduled for sync
 POD	otel:oteldemo-featureflagservice-84b8575847-mp87z	16 m	Pulled	Container image 'ghcr.io/open-telemetry/demo:1.4.0-featureflagservice' already present on machine
 POD	otel:oteldemo-featureflagservice-84b8575847-mp87z	1 h	BackOff	Back-off restarting failed container
 POD	spring-petclinic:wavefront-proxy-714b865d59-z6745	4 h	BackOff	Back-off restarting failed container

observe

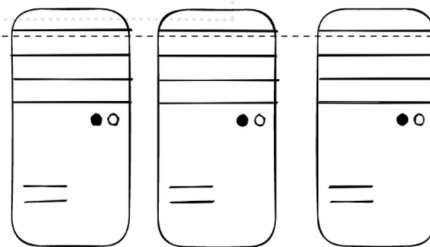
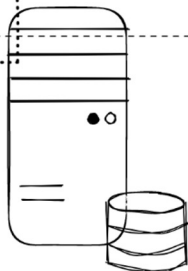


```
kubectl get ...  
kubectl describe ...  
kubectl logs ...  
kubectl debug ...  
kubectl exec ...
```



API Server

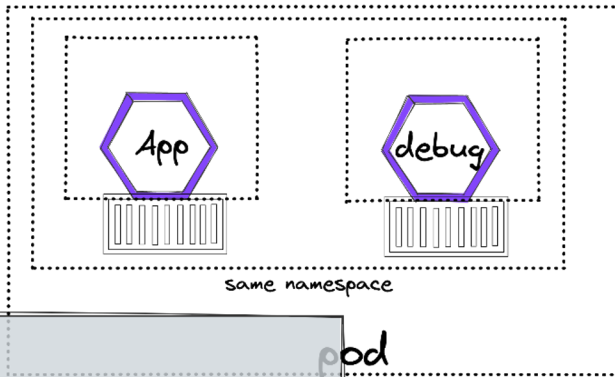
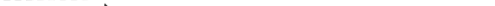
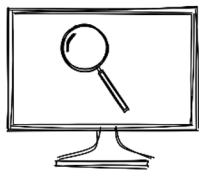
control plane



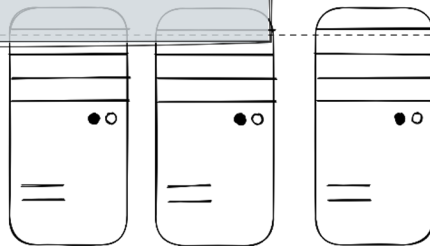
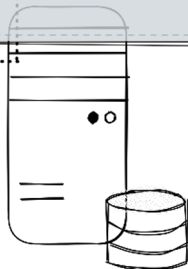
worker nodes



observe



control plane



worker nodes

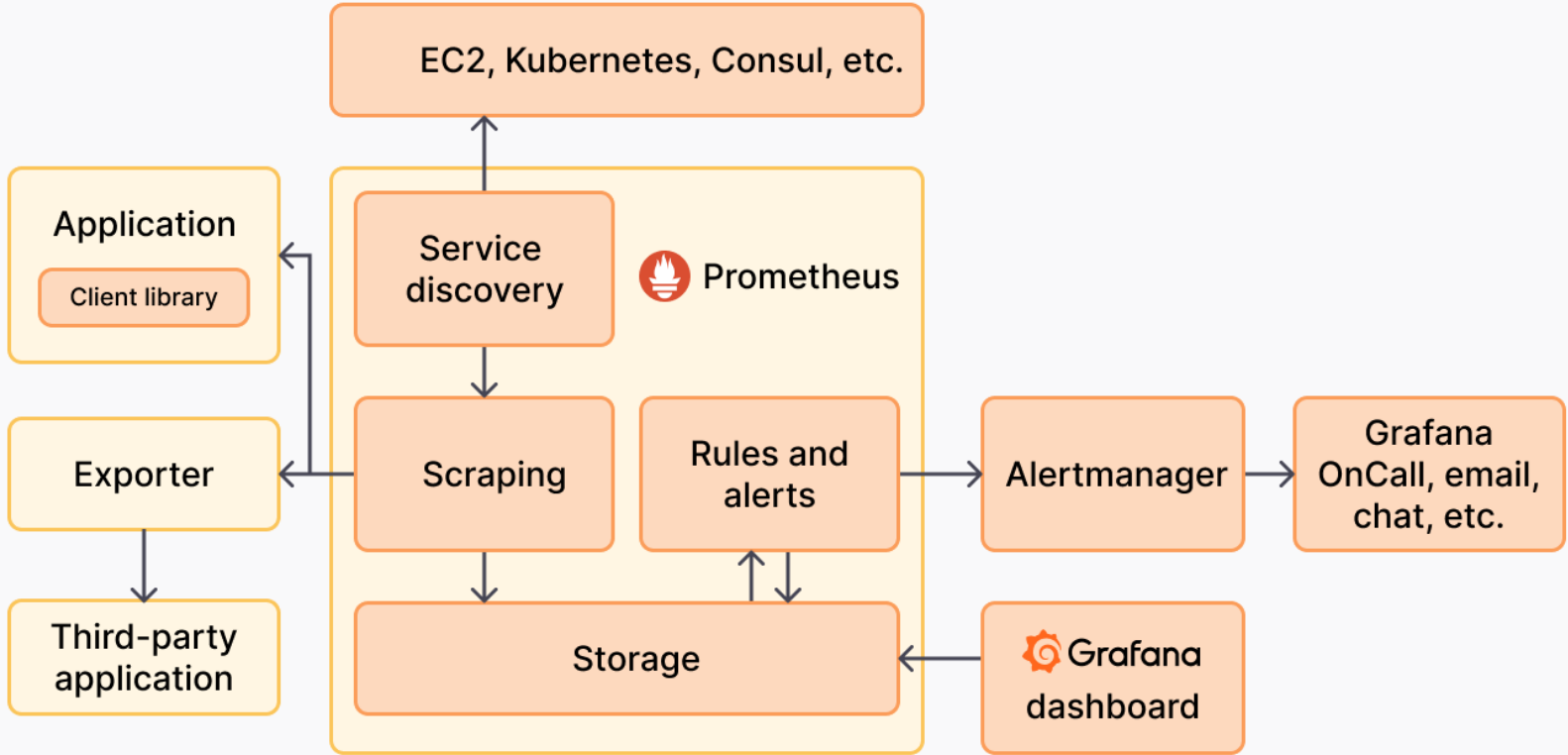


Characteristics

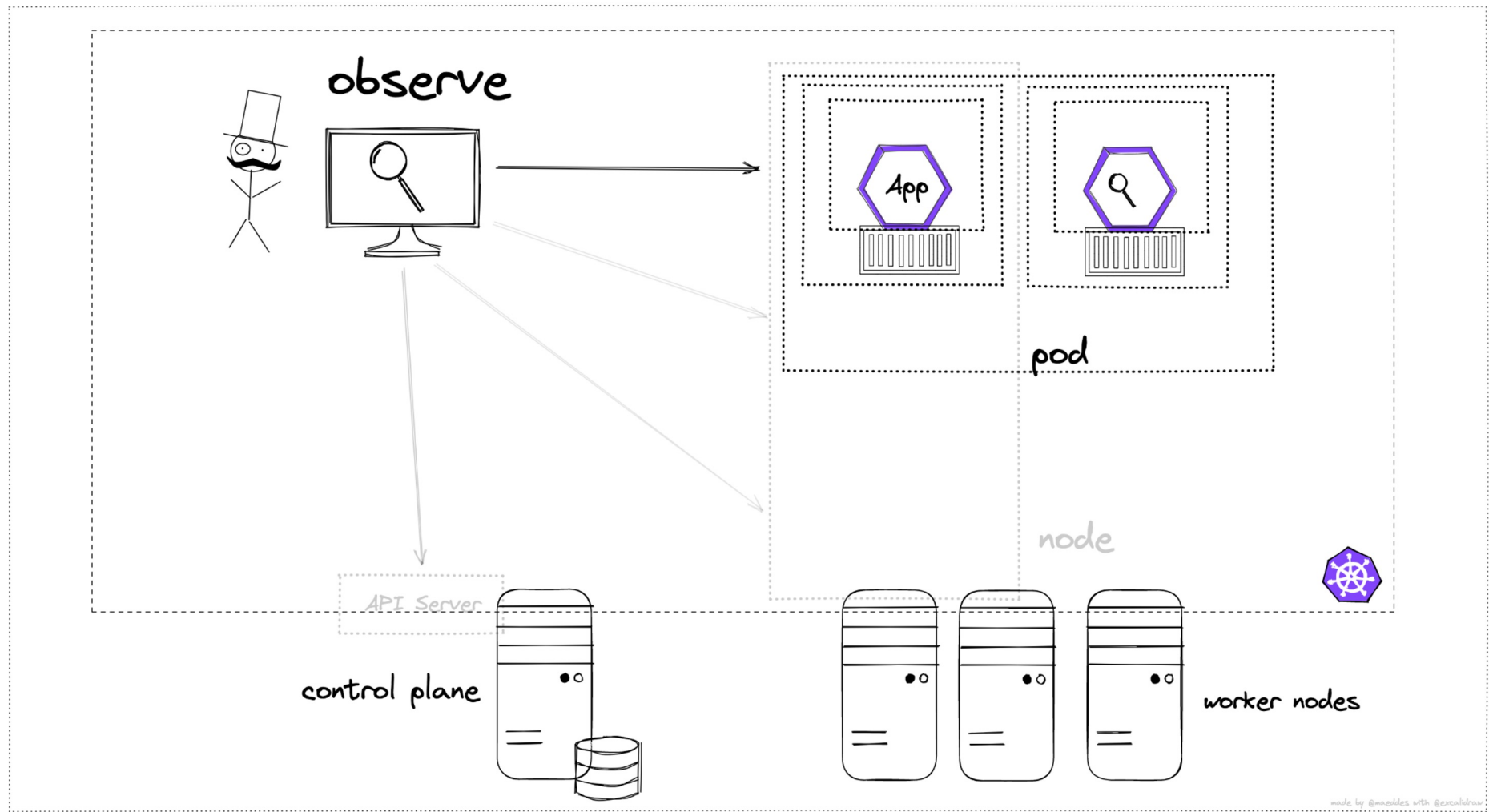
- Least intrusive
- No change to the cluster or any apps required
- Helpful for understanding Kubernetes and getting an overview

- Little network insights
- No service connections visible

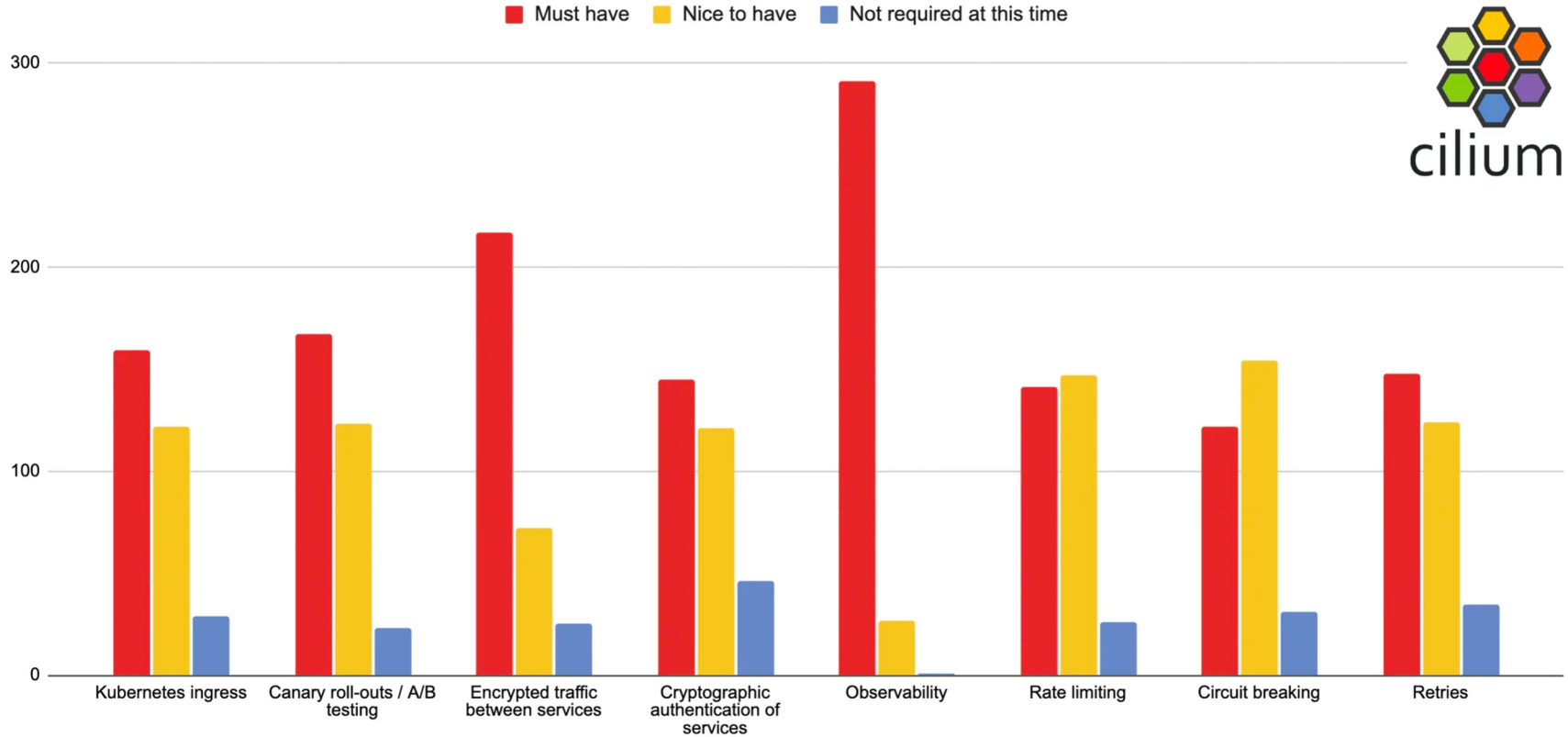
Prometheus
&
Grafana



pod-based
(Service Mesh)



What features of a Service Mesh interest you most?



replica/stateful/daemon set



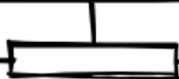
pod



replica/stateful/daemon set



pod



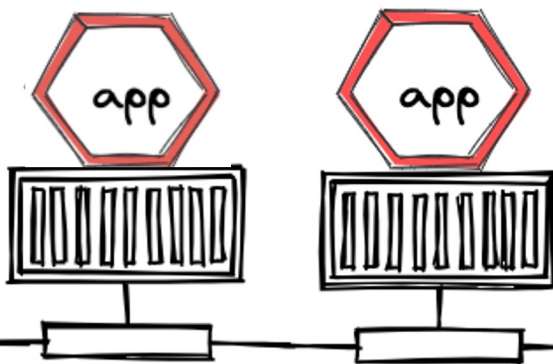
network traffic



replica/stateful/daemon set



pod



network traffic

multiple containers in pod share network

replica/stateful/daemon set



pod

proxy



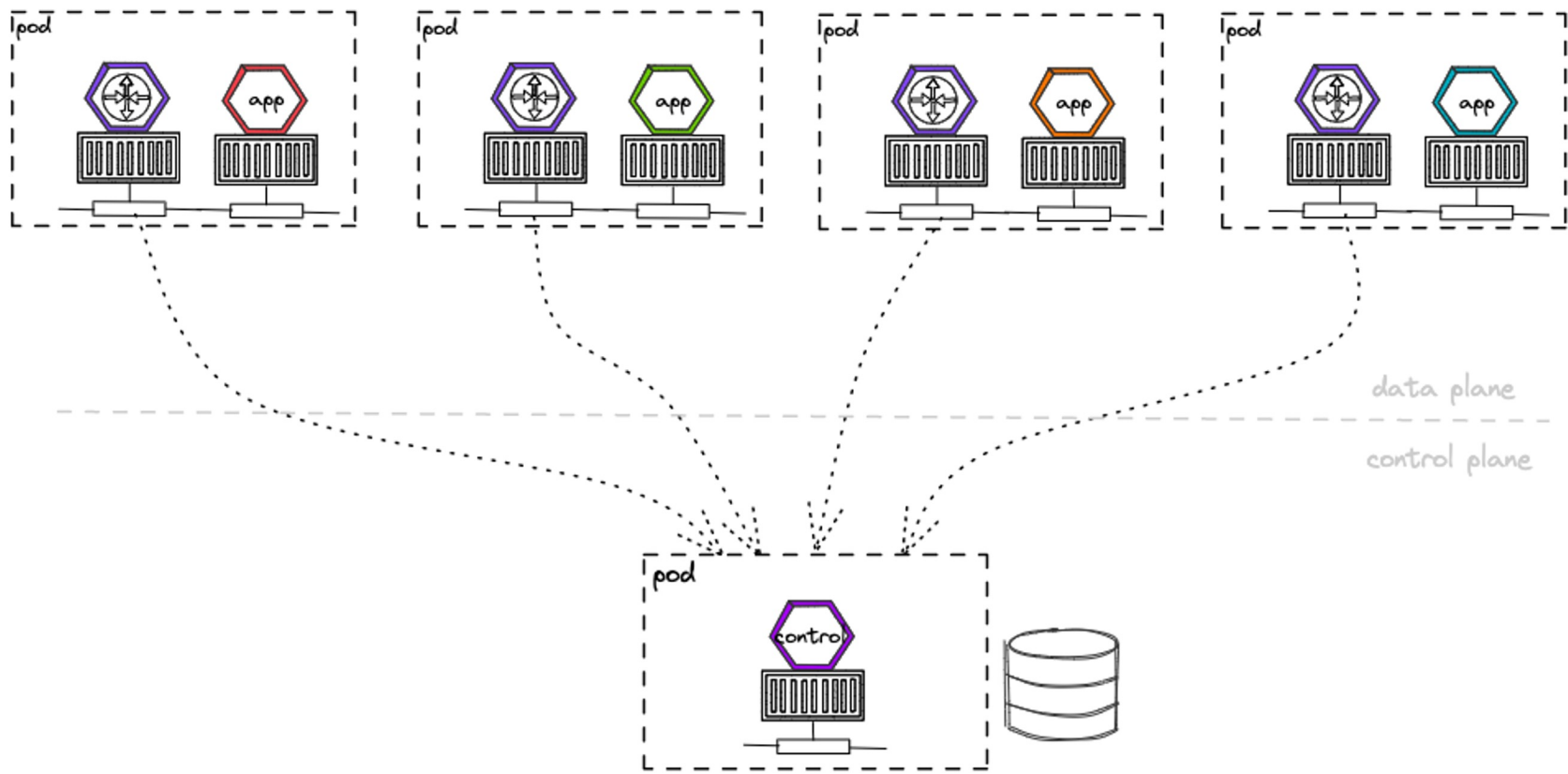
app

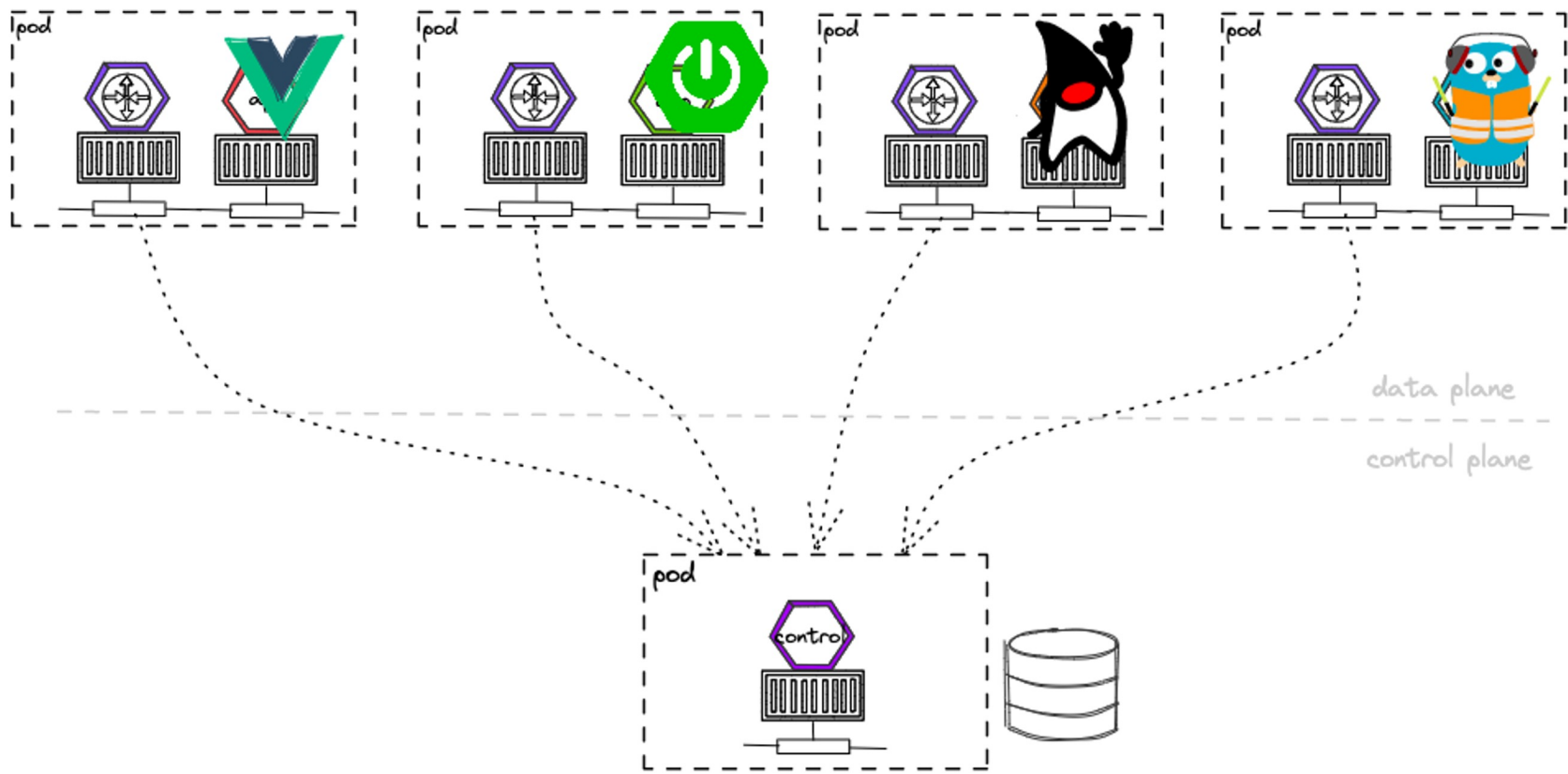


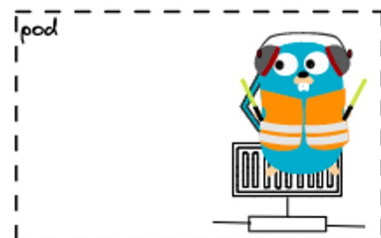
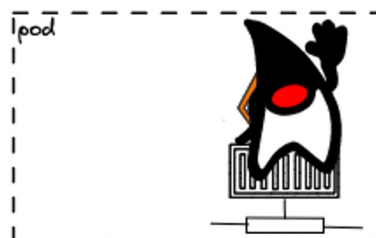
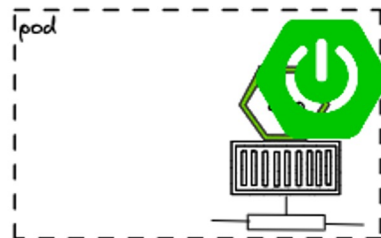
network traffic



multiple containers in pod share network

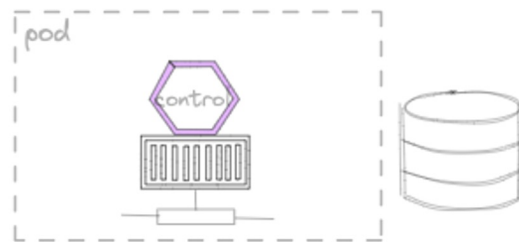


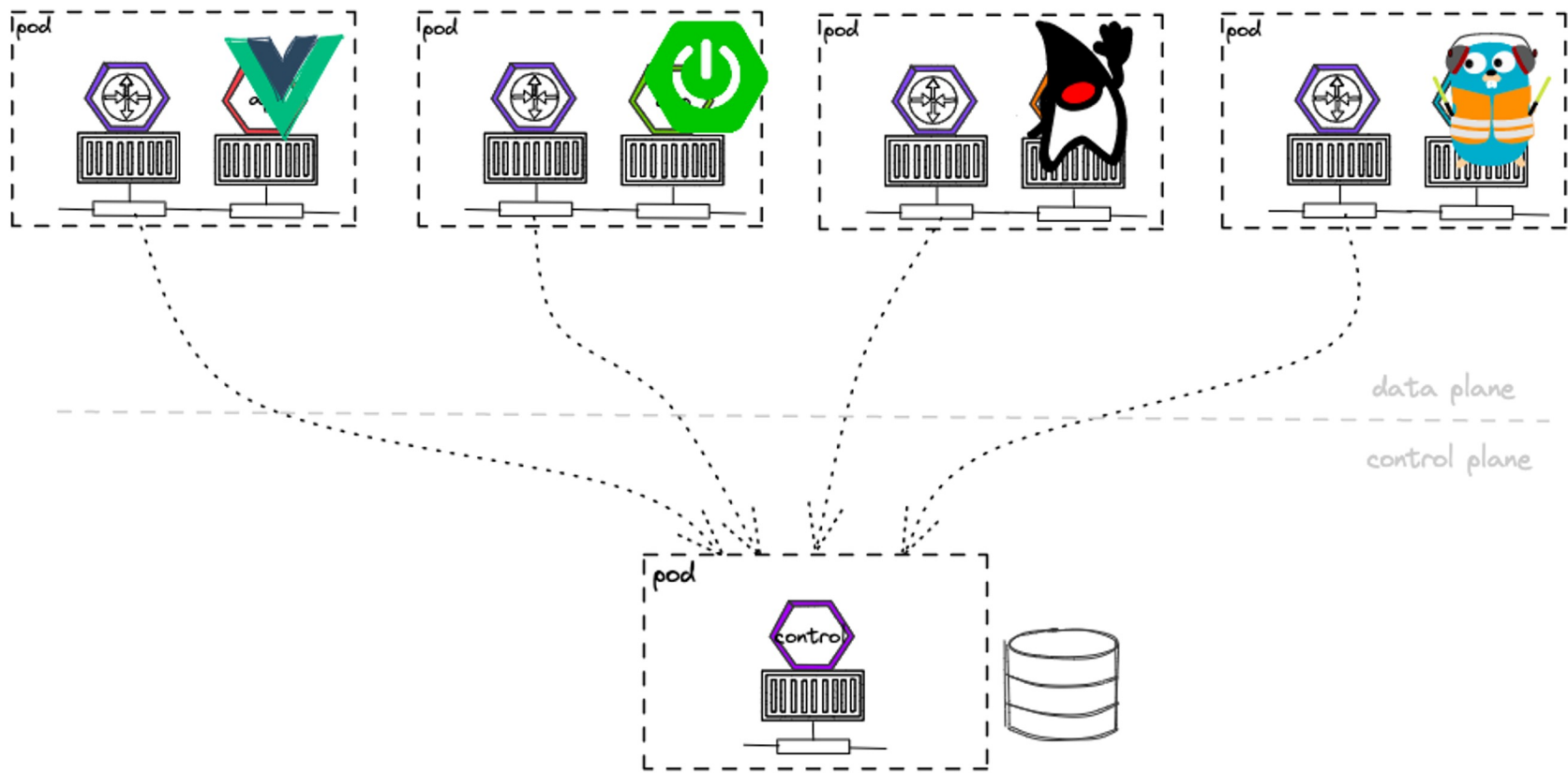


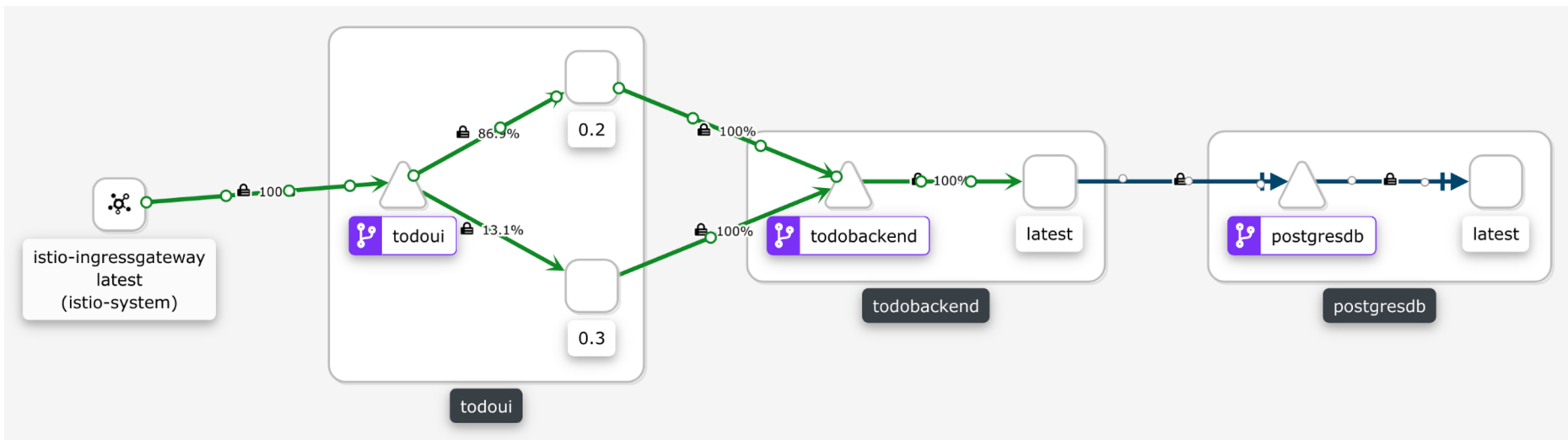


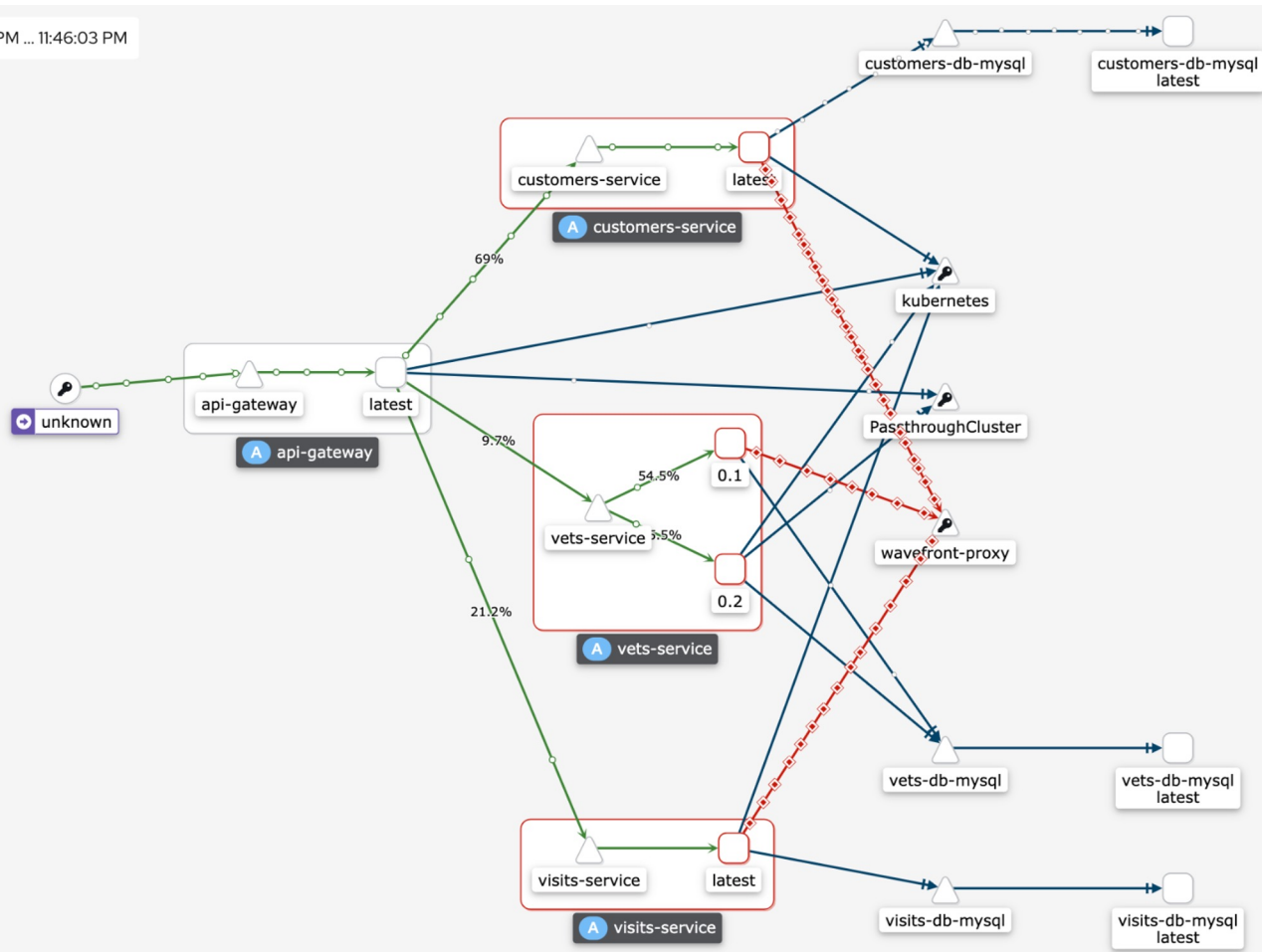
data plane

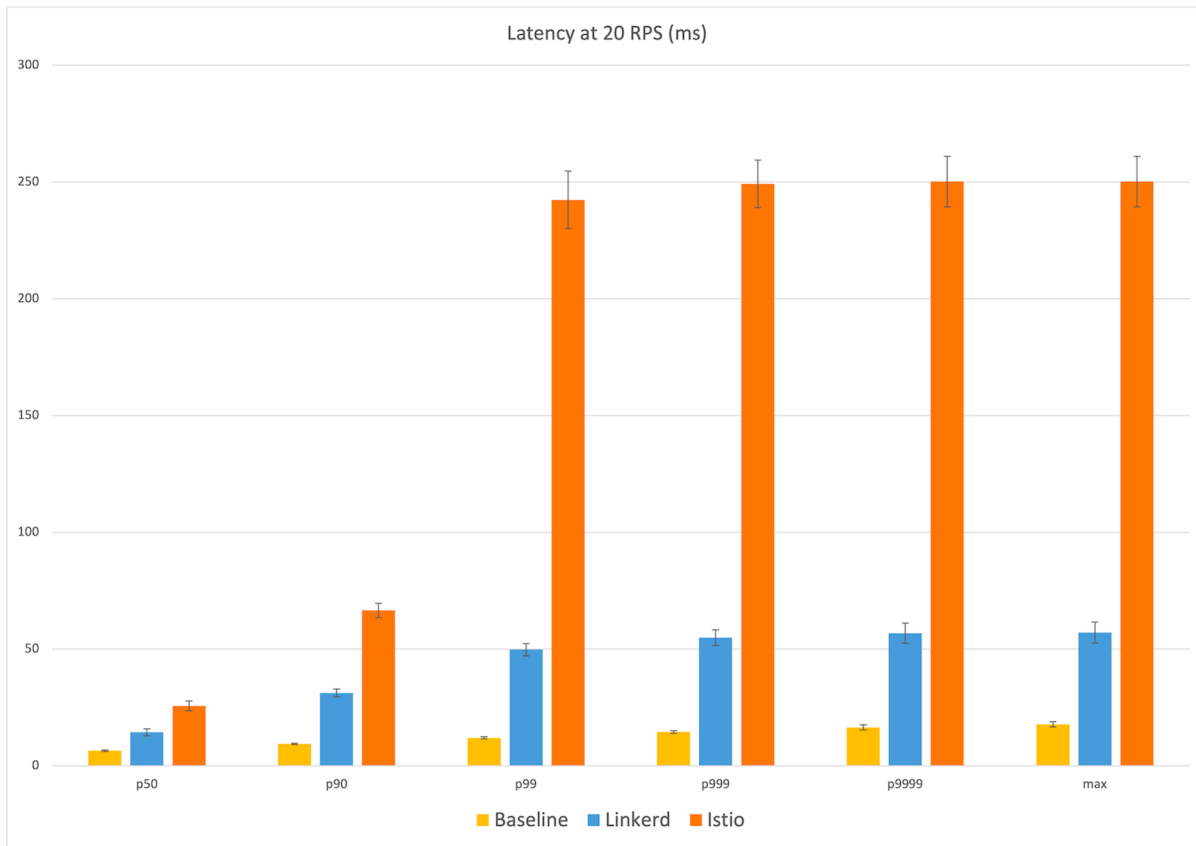
control plane











<https://www.cncf.io/blog/2021/12/17/benchmarking-linkerd-and-istio-2021-redux/>

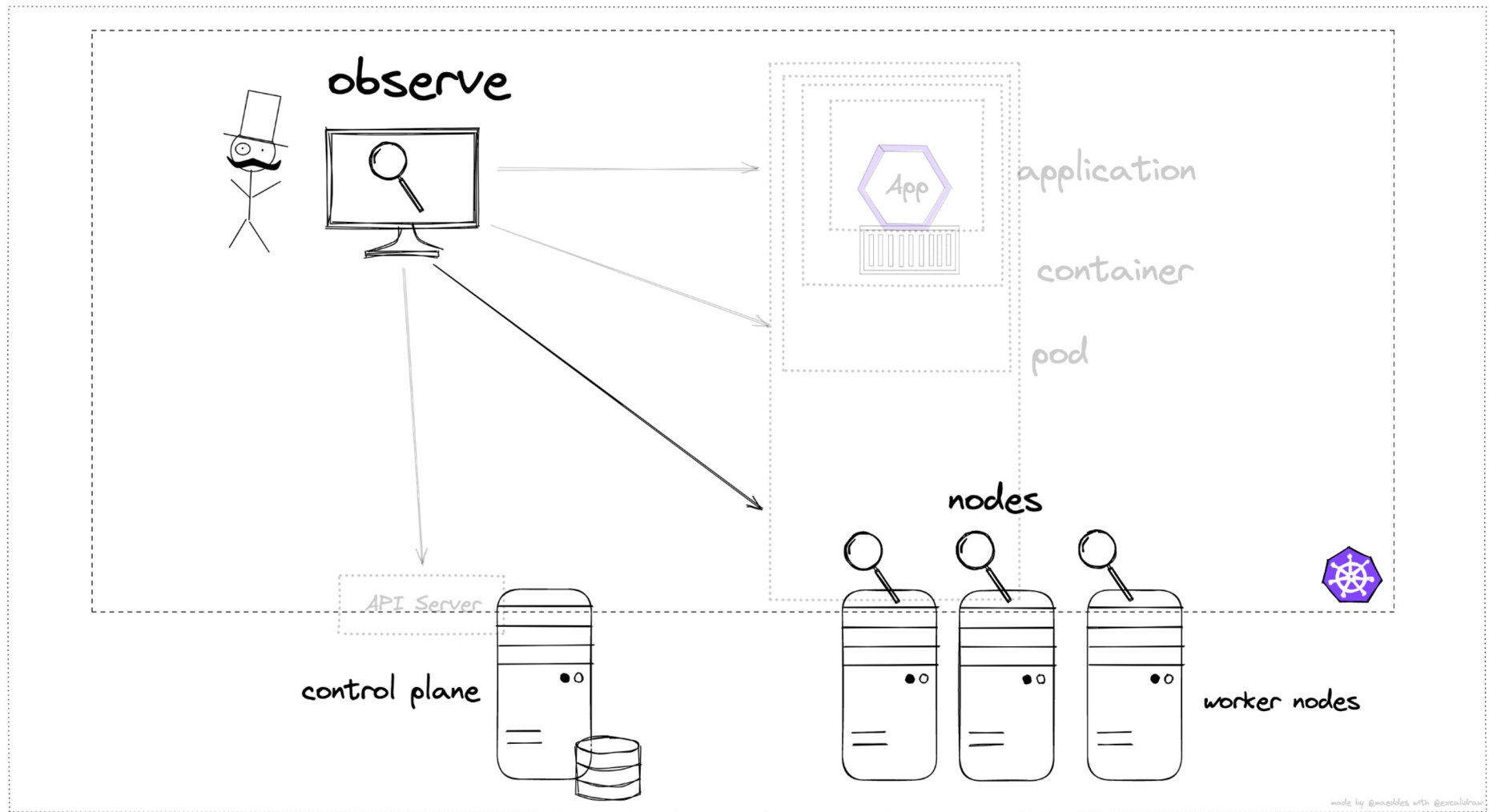
Characteristics

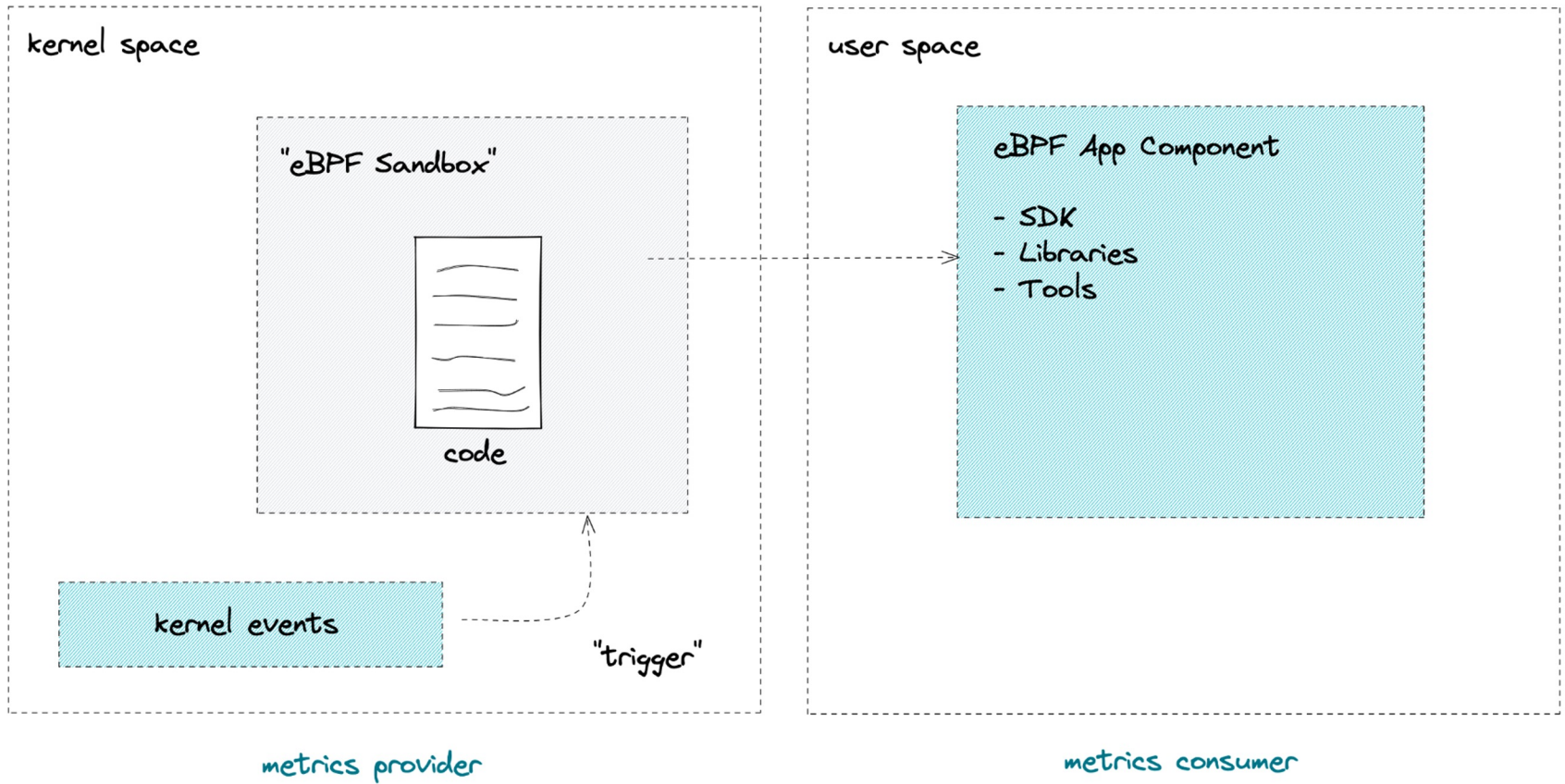
- Extends Kubernetes for limitations in network traffic awareness and shaping capabilities
- Concept of injecting sidecar proxy into each pod to have control and insight into entire network flow
- No change to application or application container required
- No application-level metrics

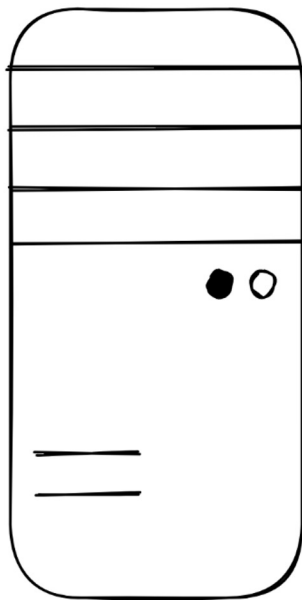
node-based
(eBPF)



extended Berkeley Packet Filter

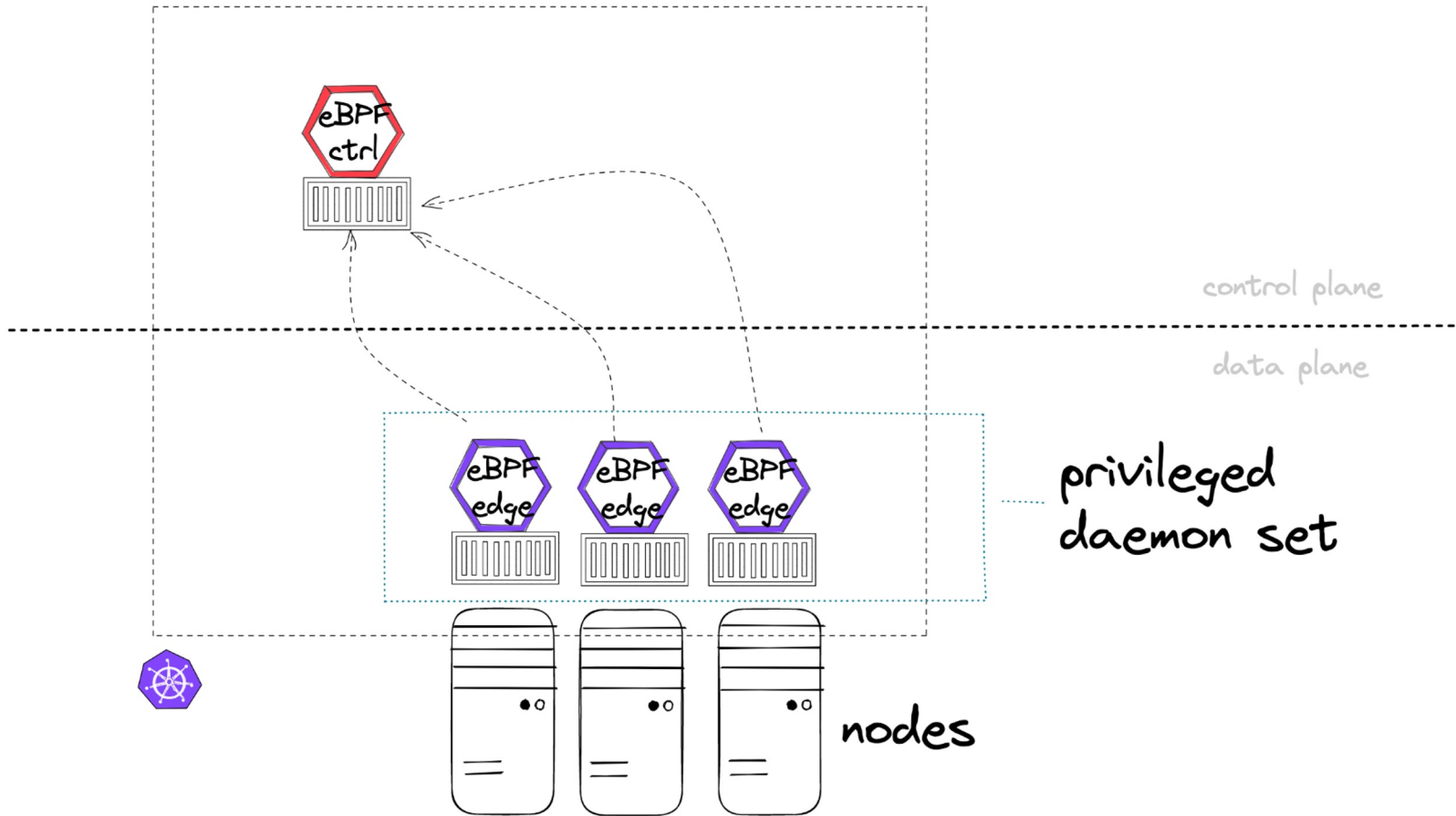


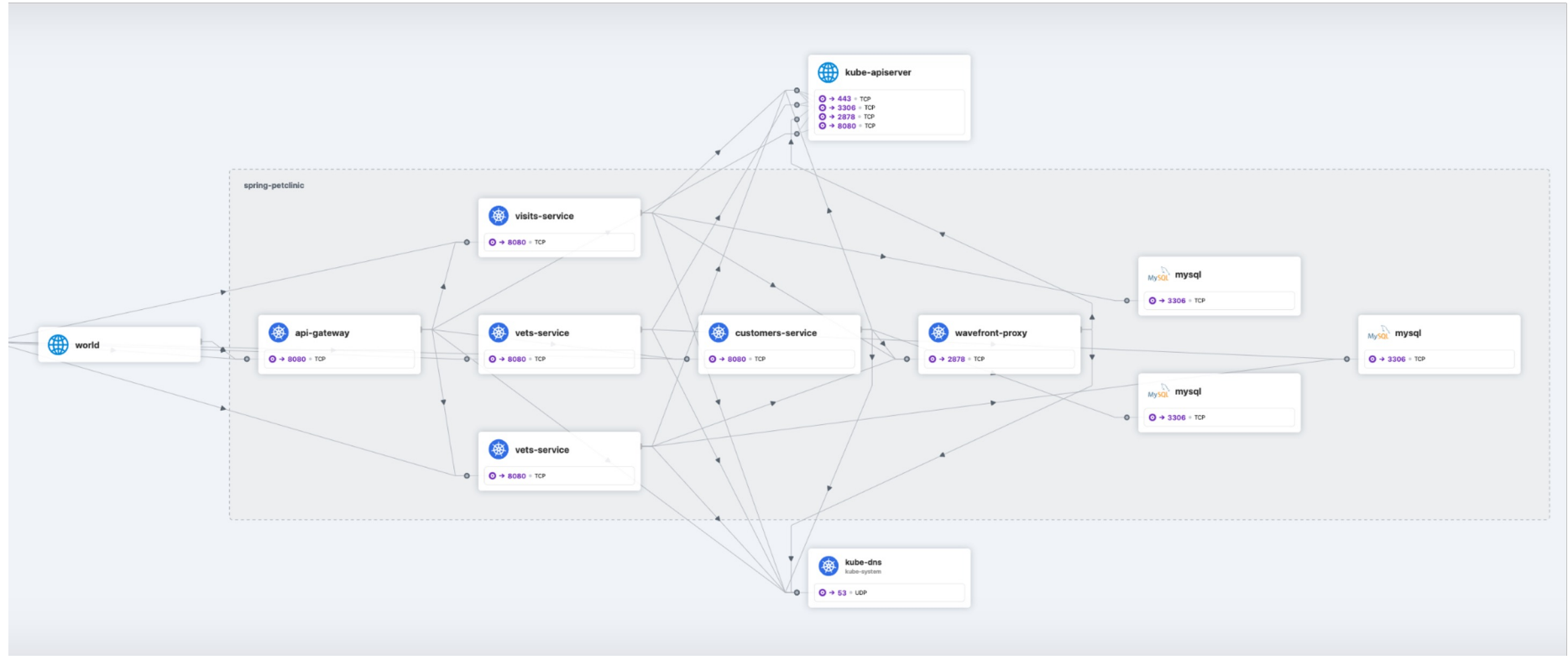


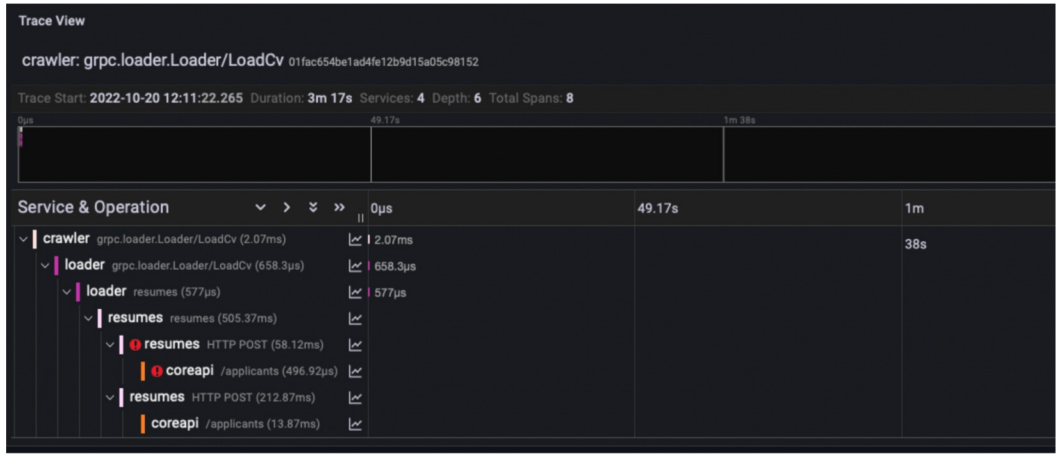
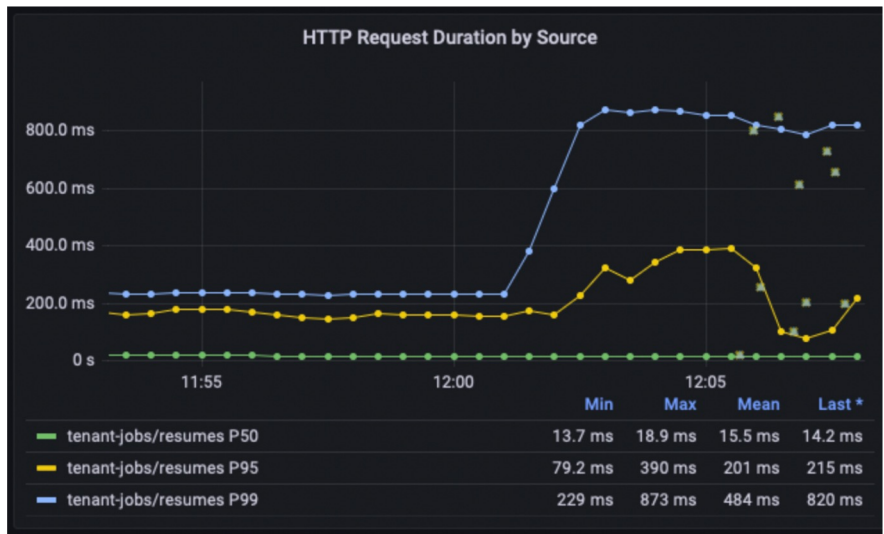


Kubernetes Worker Node









Service Map

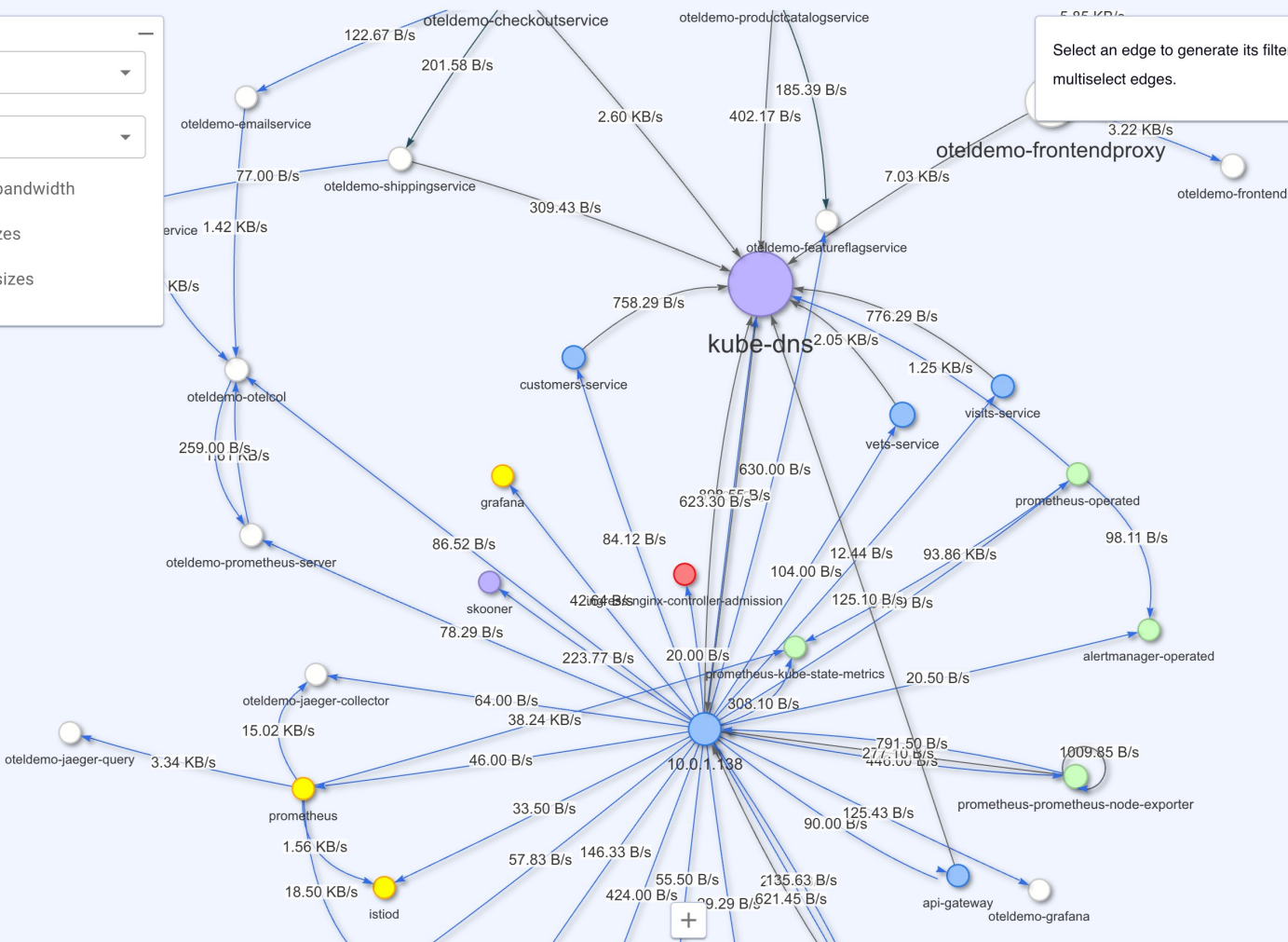


Edges
Bandwidth

Nodes
Resolved Name

- Show cumulative bandwidth
- Include request sizes
- Include response sizes

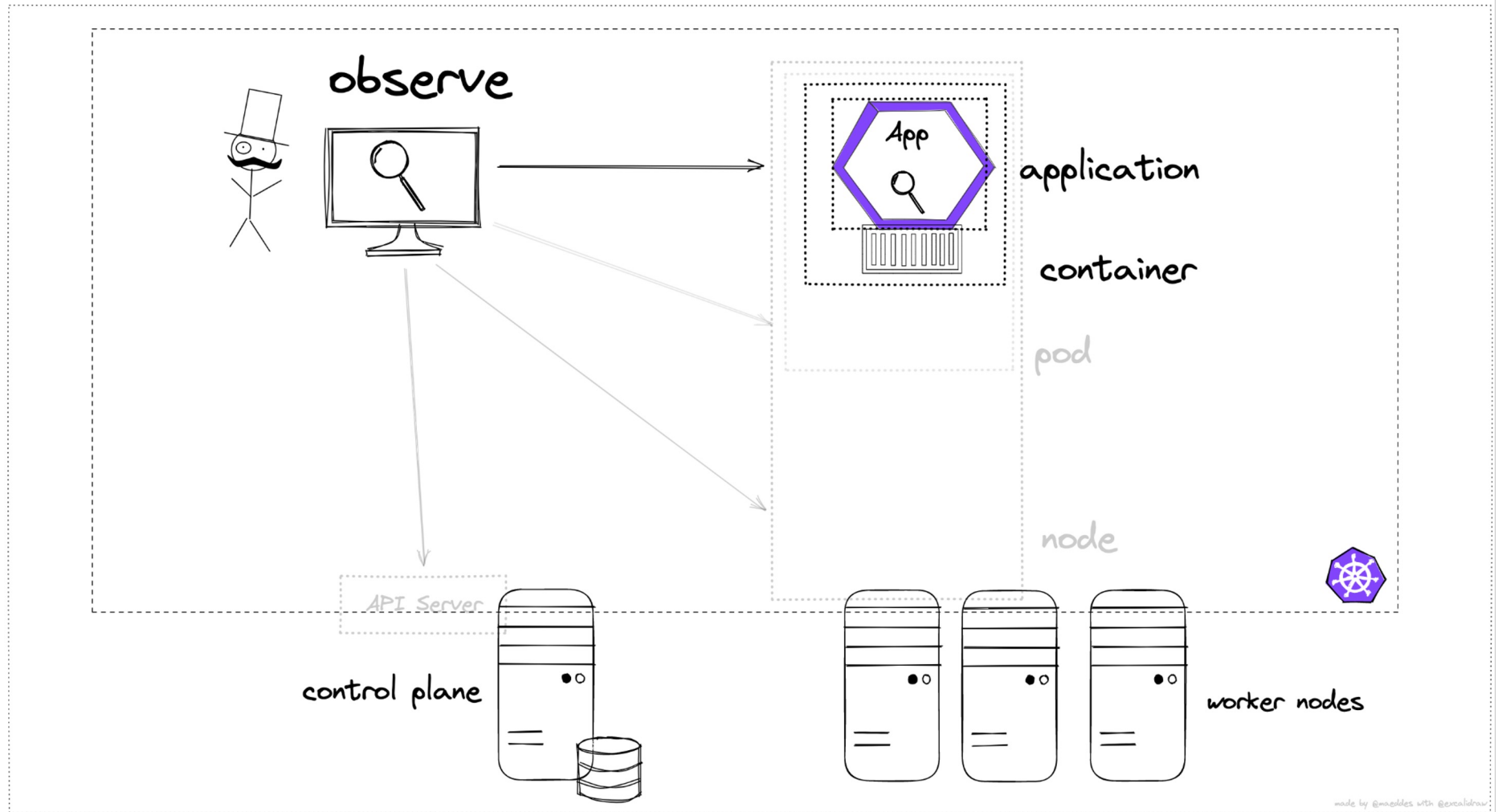
Select an edge to generate its filter. + to multiselect edges.



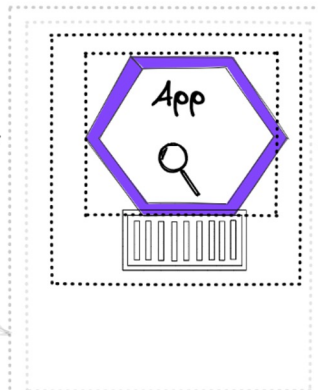
Characteristics

- Injection of proxy component on node instead of pod level
- Linux low-level functionality leveraged for Kubernetes observability
- Fast growing technology in CNCF landscape
- Application and application container untouched
- Cluster needs to be configured once

application-based



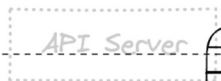
observe



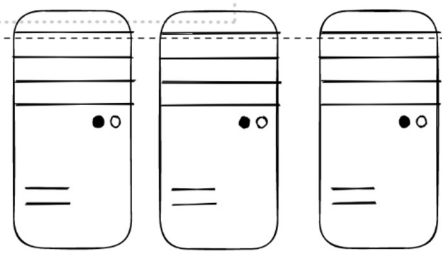
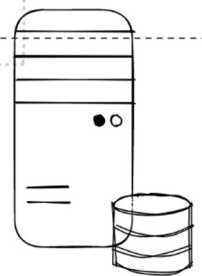
application
container

pod

node

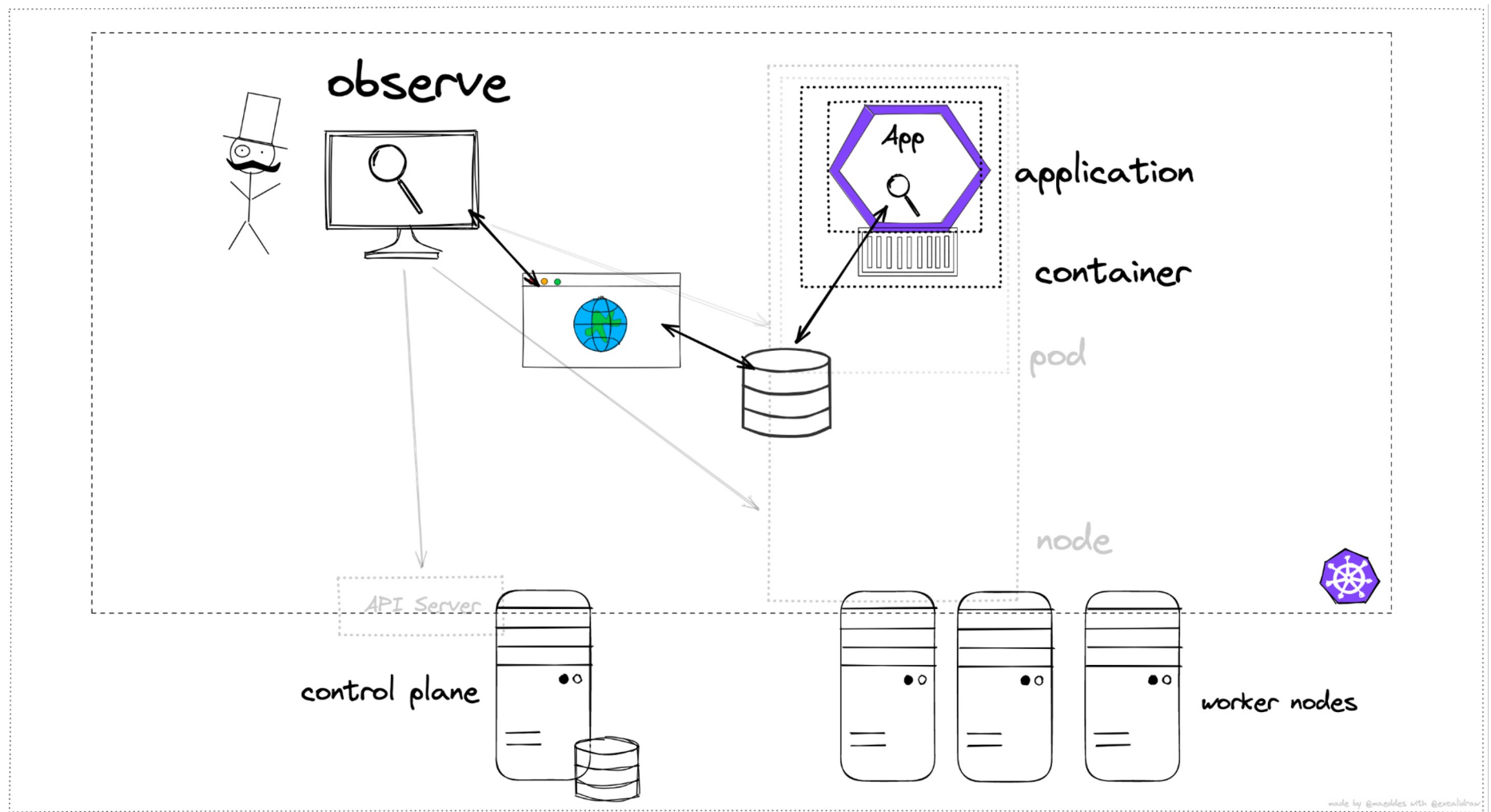


control plane

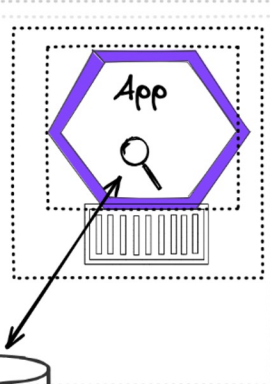
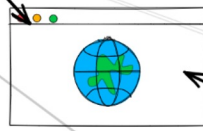


worker nodes





observe



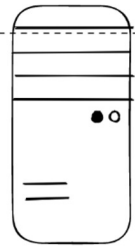
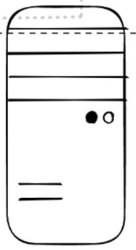
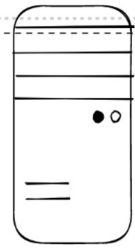
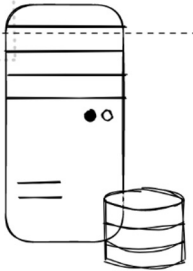
application
container

pod

node

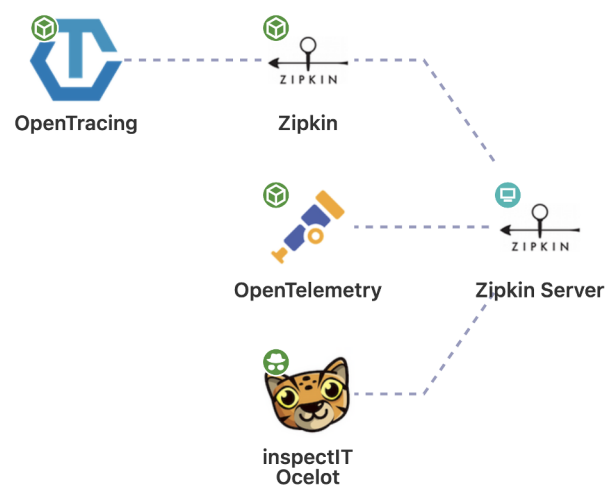
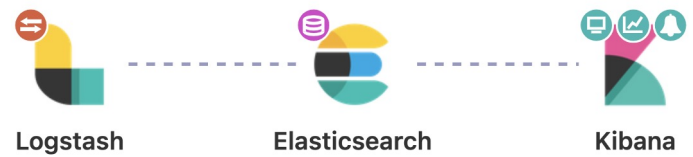
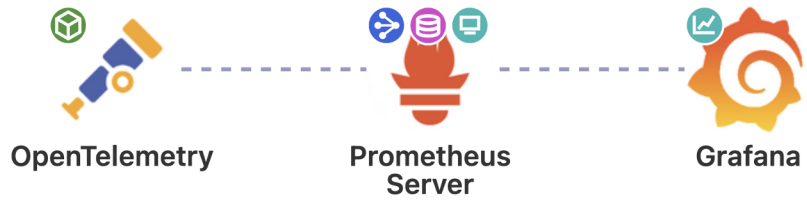
API Server

control plane



worker nodes







OpenTelemetry

Microservices

App Code

- OTel Auto. Inst.
- OTel API
- OTel SDK

3rd party service

OTLP

OTel Collector

Time Series Databases

Trace Databases

Observability Frontends & APIs

Column Stores

Kubernetes

OTLP

L7 Proxy

OTLP



OTLP

Shared Infra

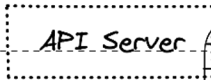
Managed DBs

APIs

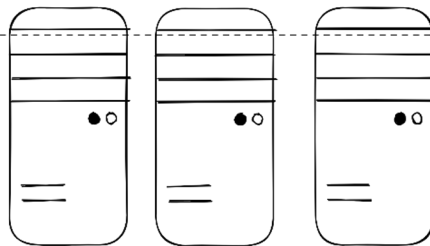
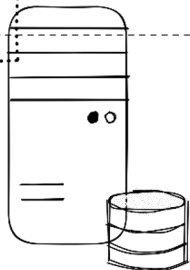
Client Instrumentation



observe



control plane



worker nodes

Language coverage

Next, you can deep dive into the documentations for the [language](#) you are using:

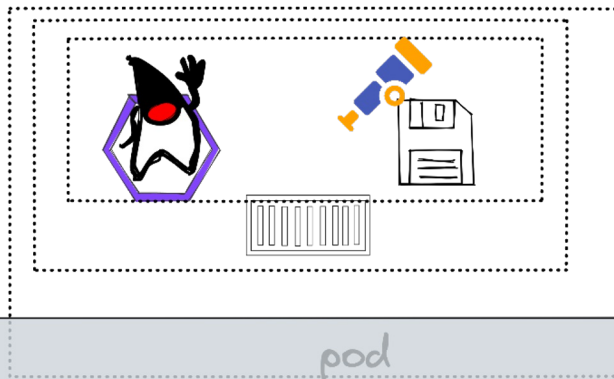
- [C++](#)
- [.NET](#)
- [Erlang / Elixir](#)
- [Go](#)
- [Java](#)
- [JavaScript / TypeScript](#)
- [PHP](#)
- [Python](#)
- [Ruby](#)
- [Rust](#)
- [Swift](#)
- [Other](#)

Automatic instrumentation is available for the following languages:

- [.NET](#)
- [Java](#)
- [JavaScript](#)
- [PHP](#)
- [Python](#)

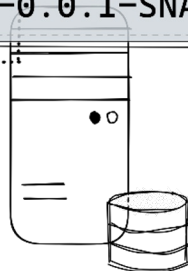


observe

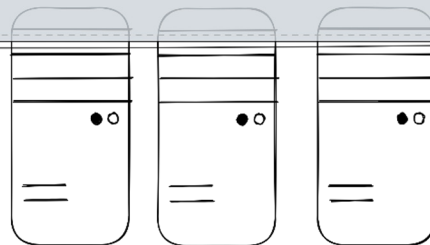


```
FROM eclipse-temurin:17
RUN mkdir -p /opt/todoui
WORKDIR /opt/todoui
ADD https://github.com/open-telemetry/opentelemetry-java-instrumentation/releases/latest/download/opentelemetry-javaagent.jar .
COPY target/todoui-0.0.1-SNAPSHOT.jar /opt/todoui
CMD ["java", "-javaagent:/opt/todoui/opentelemetry-javaagent.jar",
"-jar", "todoui-0.0.1-SNAPSHOT.jar"]
```

control plane

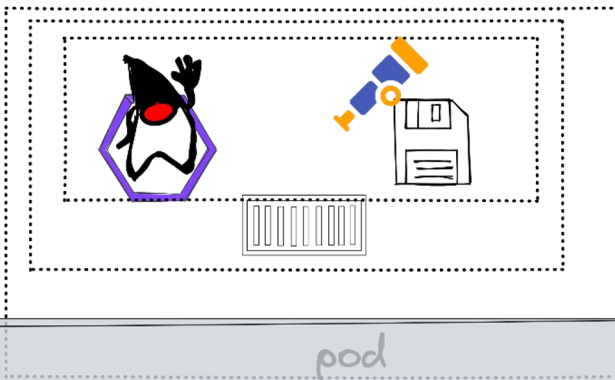


API Server



worker nodes

observe

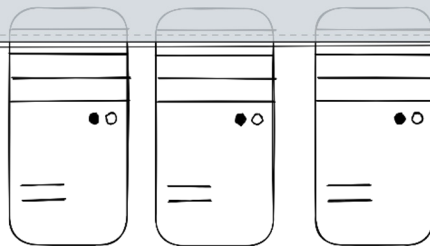
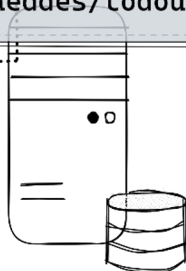


containers:

- name: todo-app-ui
- env:
 - name: OTEL_EXPORTER_JAEGER_ENDPOINT
 - value: http://oteldemo-otelcol.otel.svc.cluster.local:14250
 - name: OTEL_SERVICE_NAME
 - value: todo-app-ui
 - name: OTEL_TRACES_EXPORTER
 - value: jaeger
- image: maeddes/todoui:v23otel

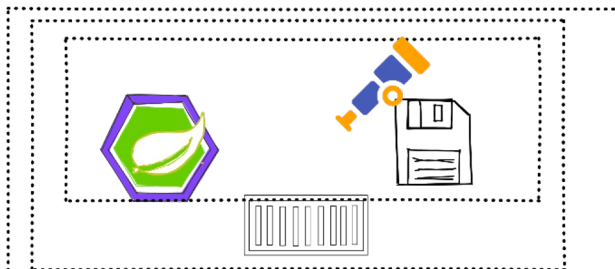
API Server

control plane



worker nodes

observe

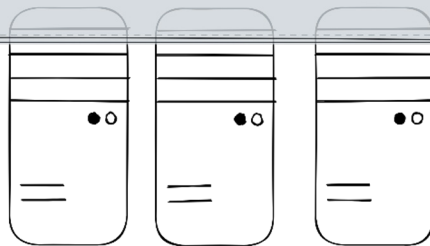
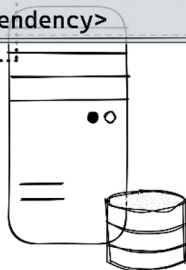


```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-sleuth</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-sleuth-otel-autoconfigure</artifactId>
</dependency>
<dependency>
  <groupId>io.opentelemetry</groupId>
  <artifactId>opentelemetry-exporter-otlp</artifactId>
  <version>1.23.1</version>
</dependency>
```

pod

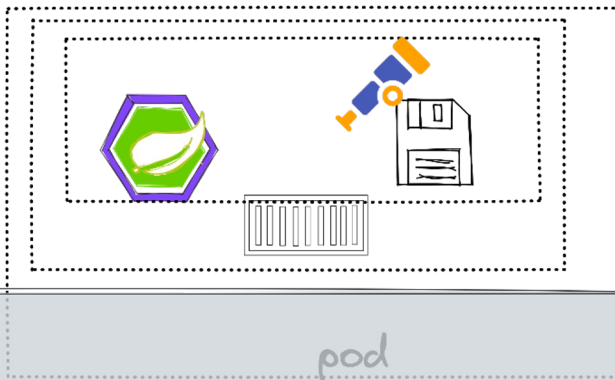
API S...

control plane



worker nodes

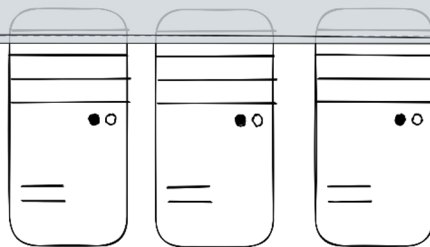
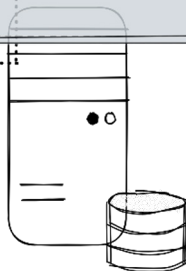
observe



```
spring.application.name=todoui  
spring.sleuth.otel.config.trace-id-ratio-based=1.0  
spring.sleuth.otel.exporter.otlp.endpoint=http://collector:4317
```

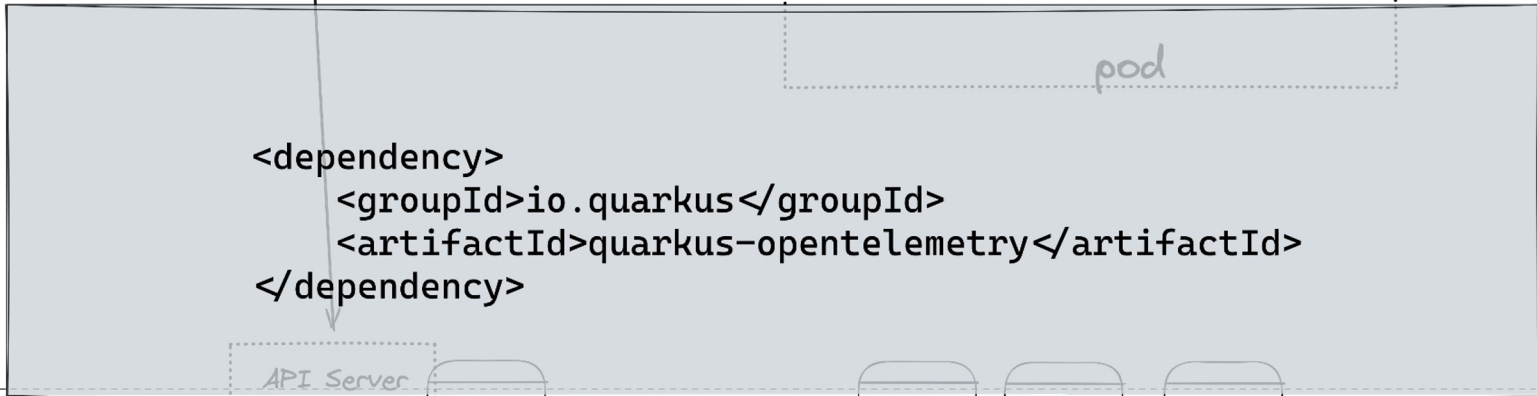
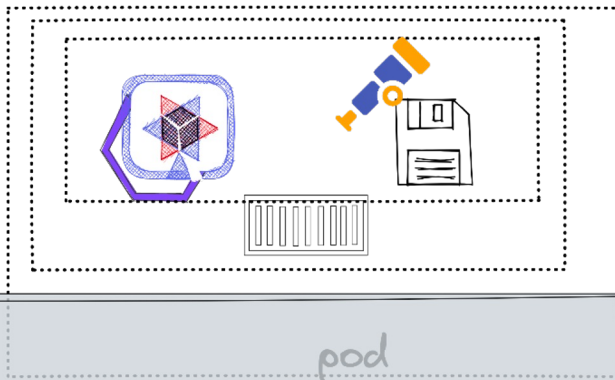
API Server

control plane



worker nodes

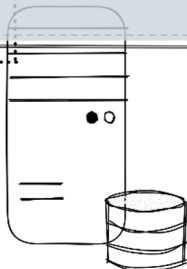
observe



```
<dependency>  
  <groupId>io.quarkus</groupId>  
  <artifactId>quarkus-opentelemetry</artifactId>  
</dependency>
```

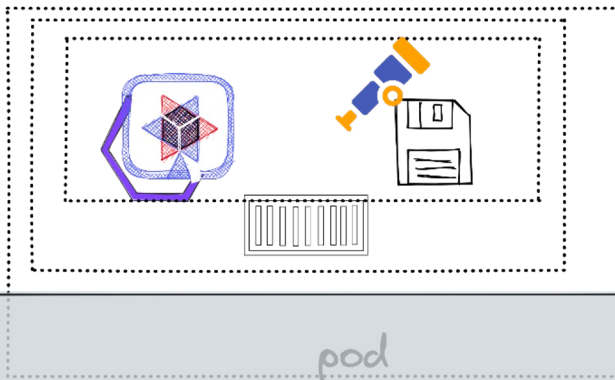
API Server

control plane



worker nodes

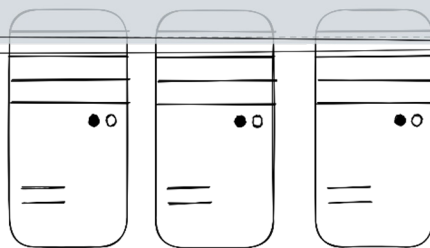
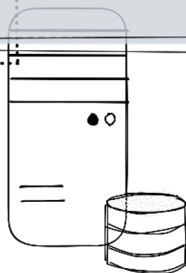
observe



```
quarkus.application.name=todo-app-quarkus
quarkus.opentelemetry.enabled=true
quarkus.opentelemetry.tracer.exporter.otlp.endpoint=http://collector:4317
```

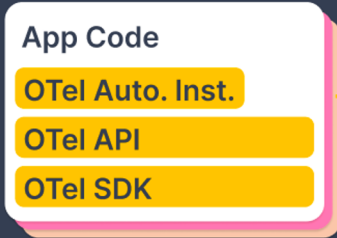
API Server

control plane



worker nodes

Microservices



OTLP

3rd party service

OTel Collector

Kubernetes

OTLP

L7 Proxy

OTLP



OTLP

Shared Infra

Time Series Databases

Trace Databases

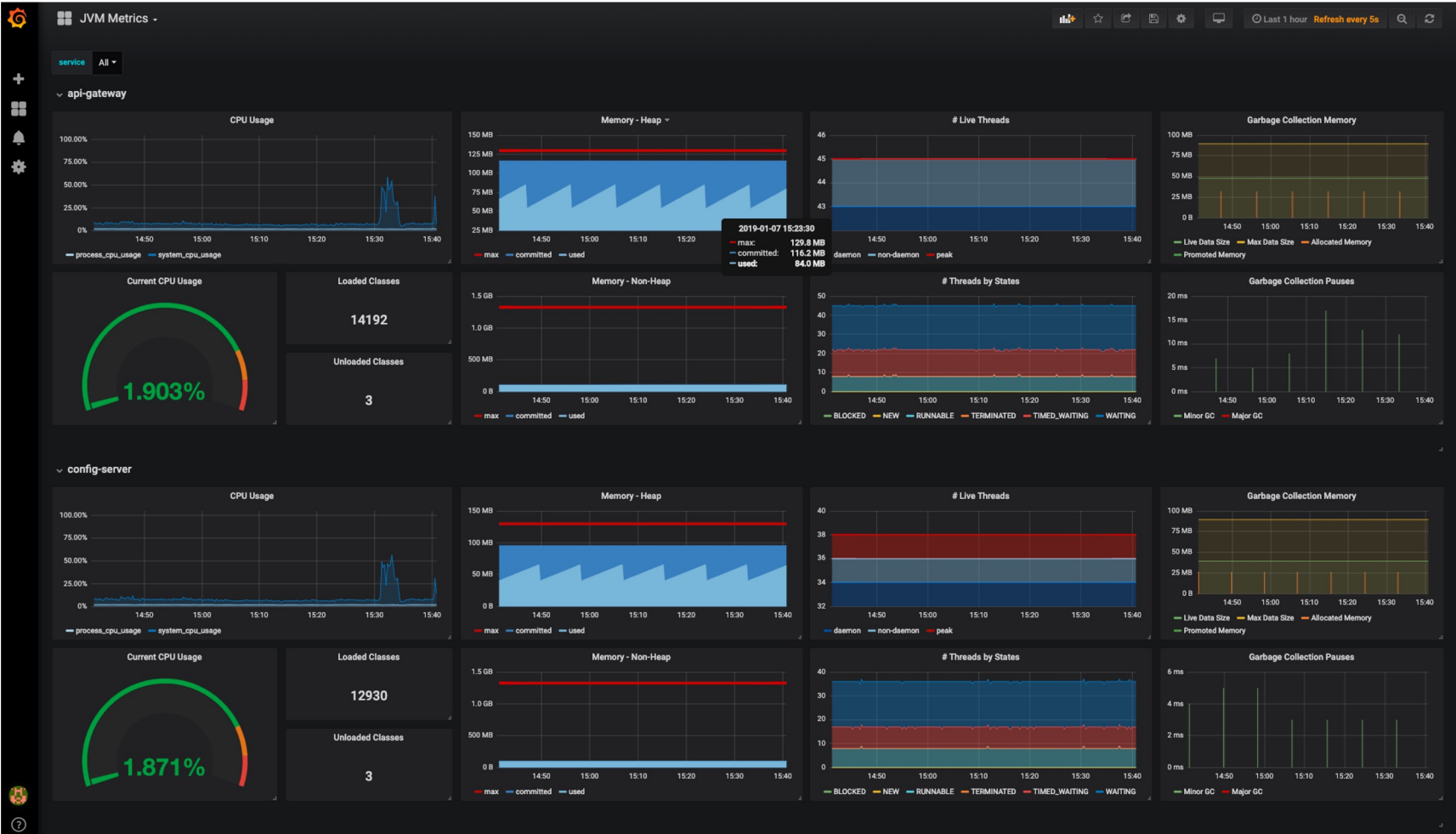
Observability Frontends & APIs

Column Stores

Managed DBs

APIs

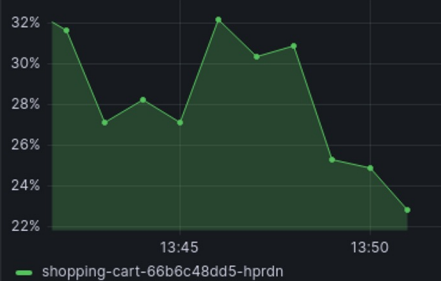
Client Instrumentation



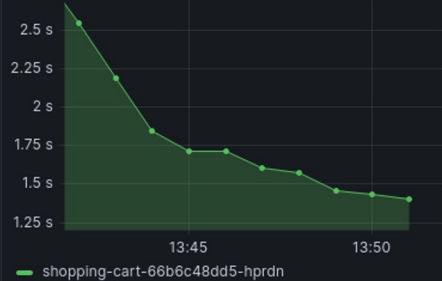
Rate



Error %



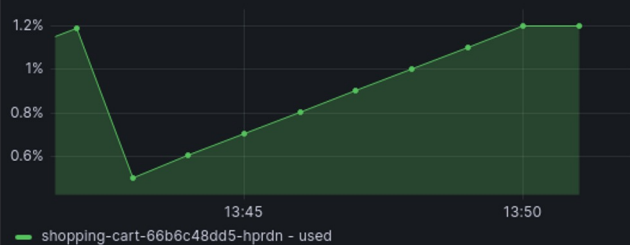
Duration



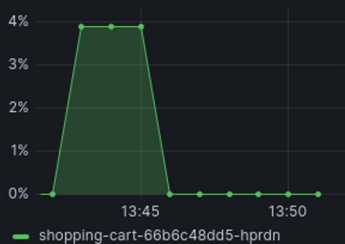
CPU utilization



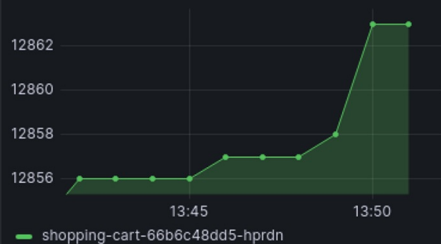
Heap Memory utilization



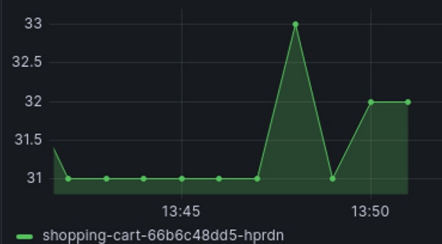
Garbage Collection



Classes



Threads

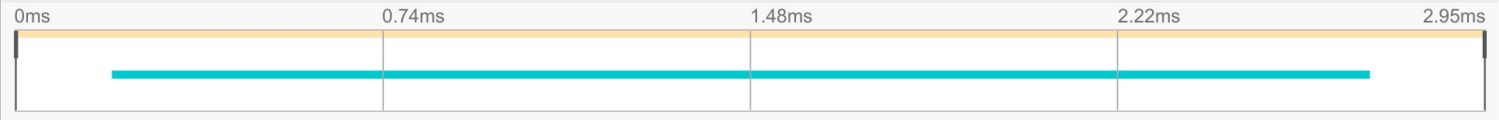




JAEGER

← **todoui.default: todobackend.default.svc.cluster.local:8080/*** 7d518f6 🔍

Trace Start **September 10 2020, 09:40:06.080** | Duration **2.95ms** | Services **2** | Depth **2** | Total Spans **2**



Service & Operation ▾ > ⌵ >> || 0ms 0.74ms 1.48ms 2.22ms 2.95ms

Service & Operation	0ms	0.74ms	1.48ms	2.22ms	2.95ms
todoui.default todobackend.default.svc...	[Timeline bar]				
todobackend.default.svc.cluster.local:8080/*	[Timeline bar]				
<p>Service: todoui.default Duration: 2.95ms Start Time: 0ms</p> <p>> Tags: component = proxy downstream_cluster = - guid:x-request-id = 9f049e5f-5c33-9294-8d08-a95453762cda h...</p> <p>> Process: ip = 10.244.0.34</p> <p>SpanID: e95a8a63a3f0ff03 🔗</p>					
todobackend.default todobacke...	[Timeline bar]				
todobackend.default.svc.cluster.local:8080/*	[Timeline bar]				
<p>Service: todobackend.default Duration: 2.53ms Start Time: 0.2ms</p> <p>> Tags: component = proxy downstream_cluster = - guid:x-request-id = 9f049e5f-5c33-9294-8d08-a95453762cda h...</p> <p>> Process: ip = 10.244.0.30</p> <p>SpanID: d0ae89ed674ea9f6 🔗</p>					



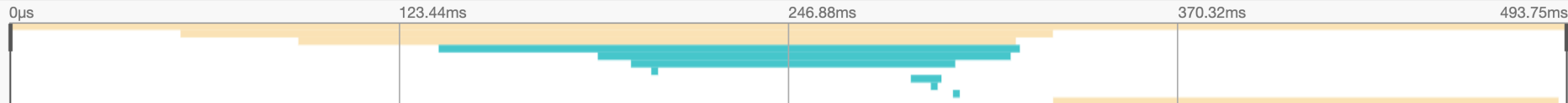
todoui: GET / e83583d

Find...

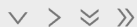


Trace Timeline

Trace Start **October 18 2023, 17:12:25.330** | Duration **493.75ms** | Services **2** | Depth **8** | Total Spans **11**



Service & Operation



0µs 123.44ms 246.88ms 370.32ms 493.75ms

todoui GET /

todoui TodouiApplication.getItems

todoui GET

todobackend GET /todos/

todobackend TodobackendApplication.getTodos

todobackend TodoRepository.findAll

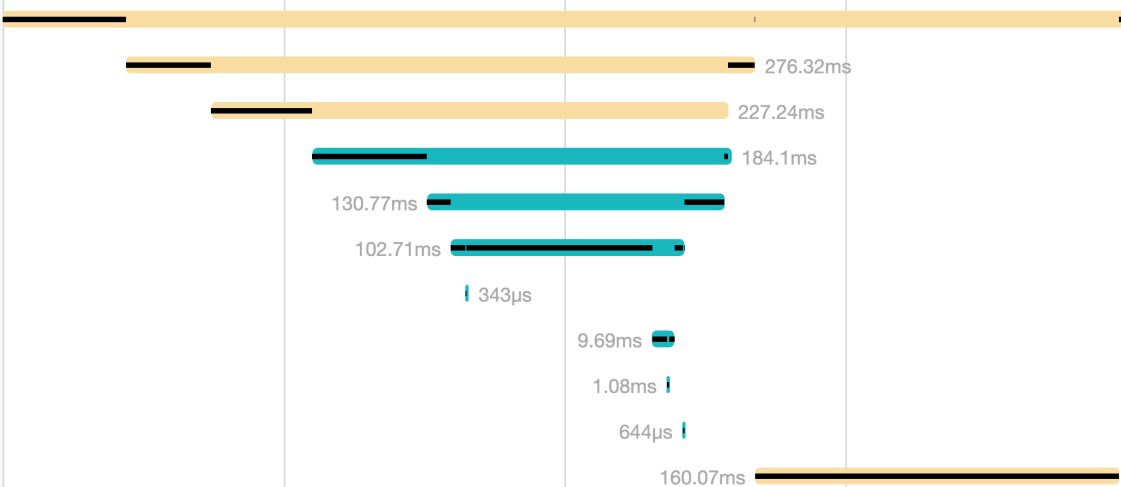
todobackend mydb

todobackend SELECT Todo

todobackend SELECT mydb.todo

todobackend Transaction.commit

todoui Render items

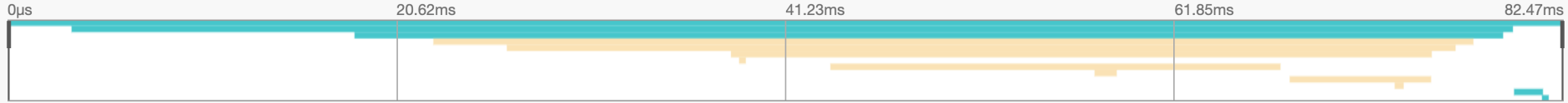




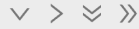
todoui: POST / ac4a333

Trace Timeline ▾

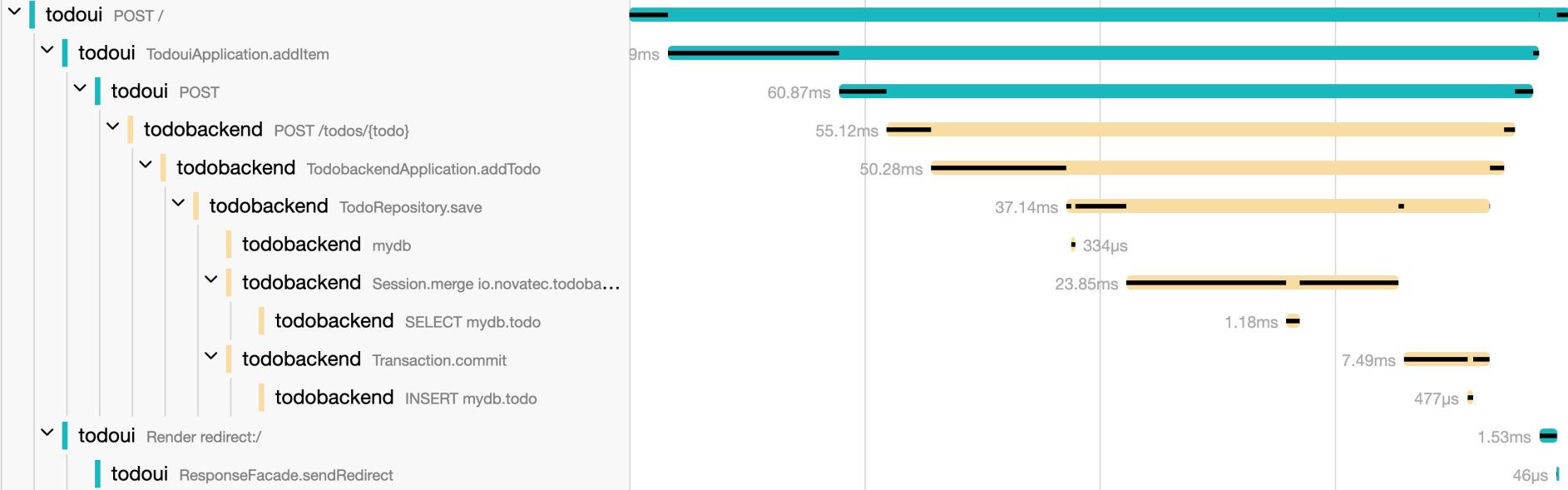
Trace Start **October 18 2023, 17:34:12.561** | Duration **82.47ms** | Services **2** | Depth **8** | Total Spans **13**



Service & Operation



0µs 20.62ms 41.23ms 61.85ms 82.47ms



```
@PostMapping("/todos/{todo}")
String addTodo(@PathVariable String todo){

    this.someUselessMethod(todo);
    //todoRepository.save(new Todo(todo));
    return todo;

}
```

```
@WithSpan
String someUselessMethod(@SpanAttribute String todo){

    todoRepository.save(new Todo(todo));
    if(todo.equals(anObject:"slow")){
        try {
            Thread.sleep(millis:1000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
    if(todo.equals(anObject:"fail")){
        System.out.println(x:"Failing ...");
        System.exit(status:1);
    }
    return todo;

}
```

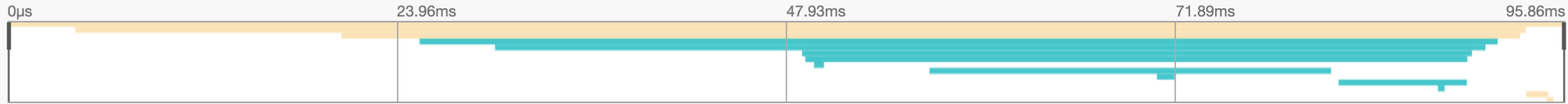
```
@PostMapping("/todos/{todo}")
String addTodo(@PathVariable String todo){

    this.someUselessMethod(todo);
    //todoRepository.save(new Todo(todo));
    return todo;
}
```

```
@WithSpan
String someUselessMethod(@SpanAttribute String todo){

    todoRepository.save(new Todo(todo));
    if(todo.equals(anObject:"slow")){
        try {
            Thread.sleep(millis:1000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
    if(todo.equals(anObject:"fail")){
        System.out.println(x:"Failing ...");
        System.exit(status:1);
    }
    return todo;
}
```

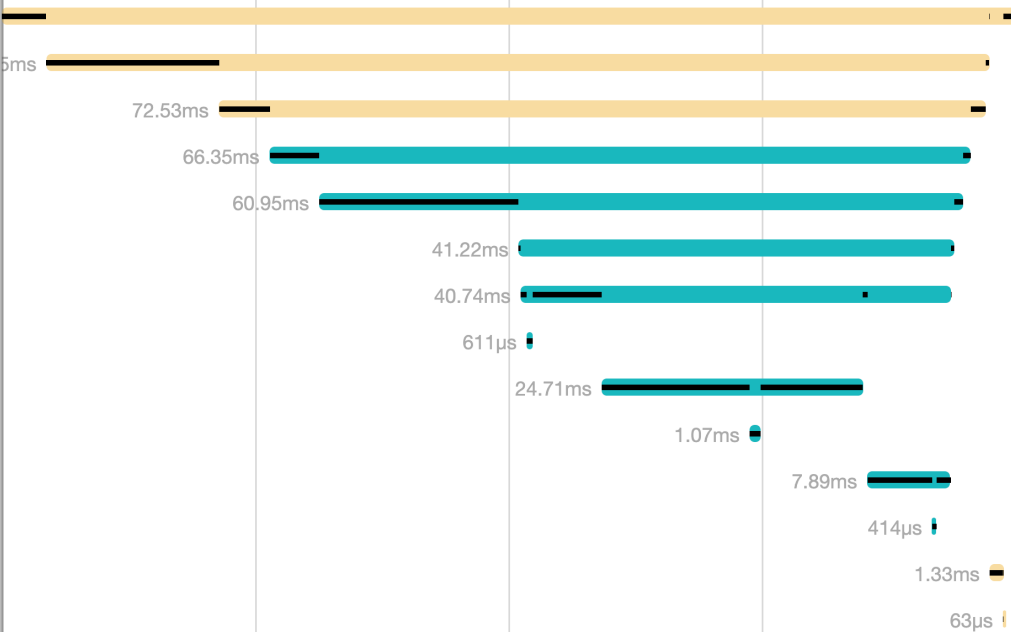

Trace Start **October 18 2023, 17:49:45.016** | Duration **95.86ms** | Services **2** | Depth **9** | Total Spans **14**



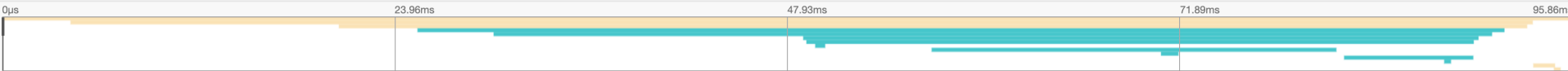
Service & Operation

0µs 23.96ms 47.93ms 71.89ms 95.86ms

- todoui POST /
 - todoui TodouiApplication.addItem
 - todoui POST
 - todobackend POST /todos/{todo}
 - todobackend TodobackendApplication.addItem
 - todobackend TodobackendApplication.someUselessMethod**
 - todobackend TodoRepository.save
 - todobackend mydb
 - todobackend Session.merge io.novatec.todobackend.TODO
 - todobackend SELECT mydb.todo
 - todobackend Transaction.commit
 - todobackend INSERT mydb.todo

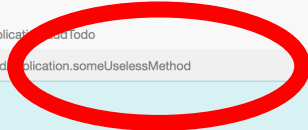
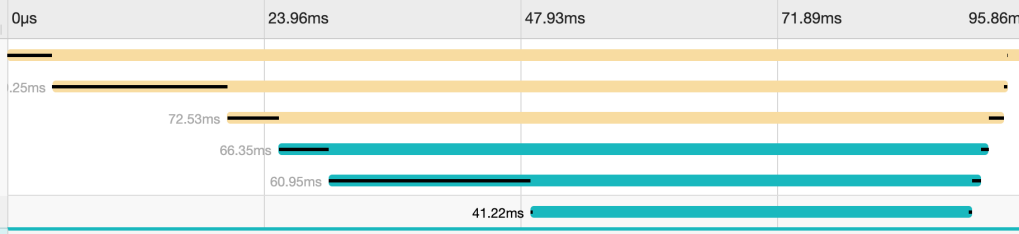


Trace Start **October 18 2023, 17:49:45.016** | Duration **95.86ms** | Services **2** | Depth **9** | Total Spans **14**



Service & Operation

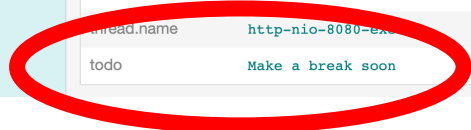
- ▼ todoui POST /
 - ▼ todoui TodouiApplication.addItem
 - ▼ todoui POST
 - ▼ todobackend POST /todos/{todo}
 - ▼ todobackend TodobackendApplication.addItem
 - ▼ todobackend TodobackendApplication.someUselessMethod



TodobackendApplication.someUselessMethod Service: todobackend | Duration: 41.22ms | Start Time: 48.89ms

▼ Tags

code.function	someUselessMethod
code.namespace	io.novatec.todobackend.TodobackendApplication
internal.span.format	proto
otel.library.name	io.opentelemetry.opentelemetry-instrumentation-annotations-1.16
otel.library.version	1.30.0-alpha
span.kind	internal
thread.id	143
thread.name	http-nio-8080-exec-1
todo	Make a break soon



How-To

Hide commercial tools

Show formats on edges



Your OpenAPM Landscape

Nothing has been selected. Start creating your custom OpenAPM landscape!

or

Learn how to use the landscape

Select filter ...

Agent In traditional APM architectures, the agents are responsible for collecting the performance data from... show more

AppDynamics Agent

Boomerang

Checkmk Agent

DD Language Agents

EaseAgent

Elastic APM Agent

Elastic Beats

Icinga 2 Agent

Instana AutoTrace™

Kamon Kanela

Kieker Agent

Nagios
Nagios Cross-Platform Agent

New Relic Agents

Performance Co-Pilot

PINPOINT
Pinpoint Agent

Prometheus Exporter

Promitor

Promtail

Scouter Agent

Sensu Client

Skywalking
SkyWalking

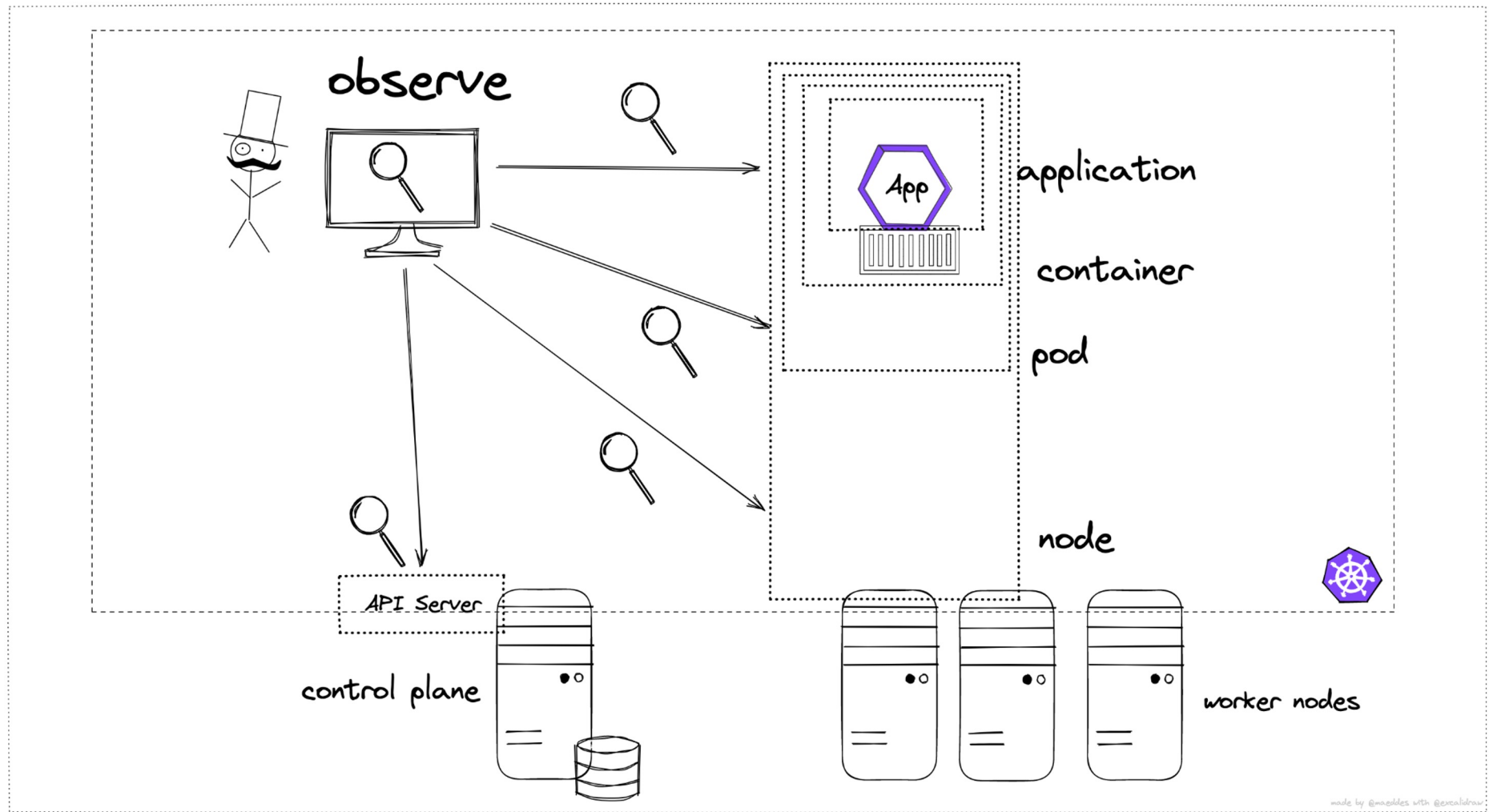
Telegraf

Wavefront J-Agent

collectd

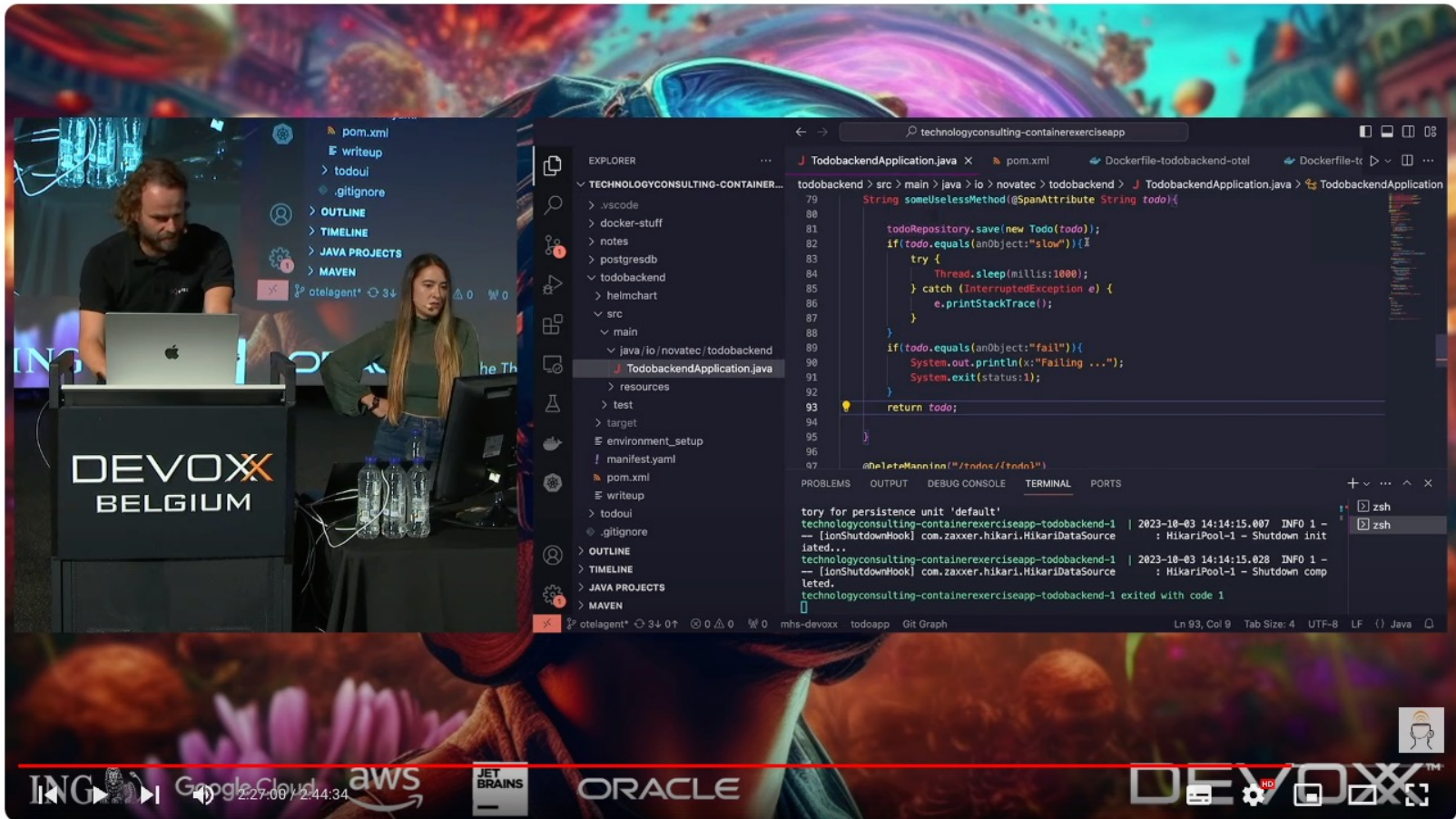
Characteristics

- Provides application level metrics and enables root cause analysis
- Changes to application or container required
- Agents specific to programming language and frameworks





<https://speakerdeck.com/maeddes/whats-going-on-in-my-cluster>



A hitchhikers guide to observe Java applications in Kubernetes By Tiffany Jernigan, Matthias Haeussl

Thanks!



matthiashaessler



@maeddes