



Observability von Java Apps in Kubernetes

Matthias Häußler, Novatec

#whoami



matthiashaessler



@maeddes



NOVATEC

Chief Technologist

Hochschule
für Technik
Stuttgart

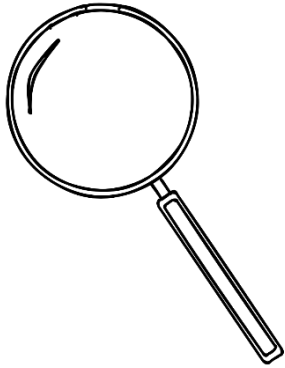
ESSLINGEN
UNIVERSITY

For people and technology.

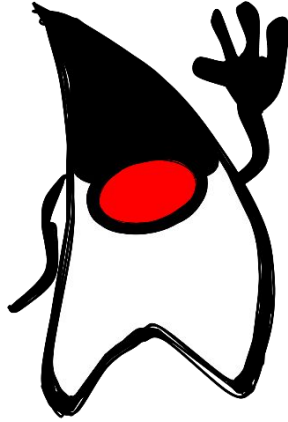
Distributed Systems



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



observability

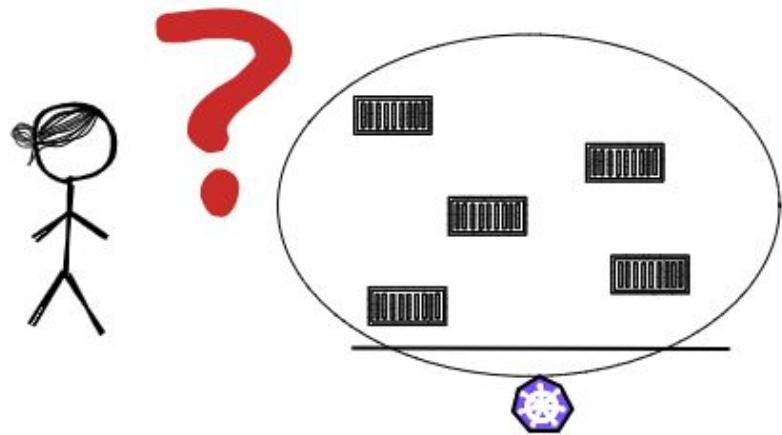
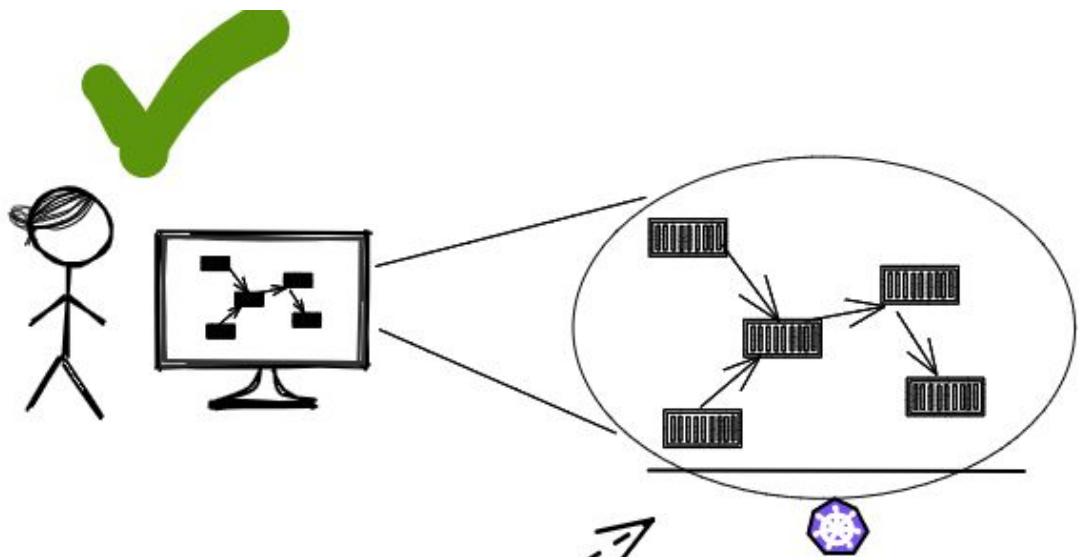


java

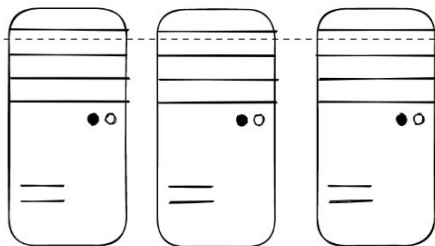
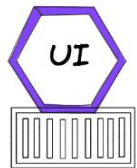


kubernetes

Why?

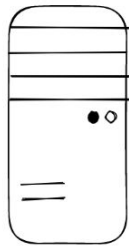


application/workloads

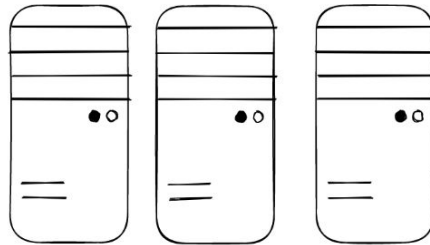


nodes

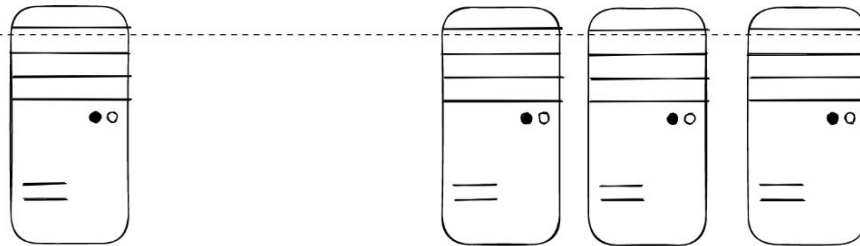
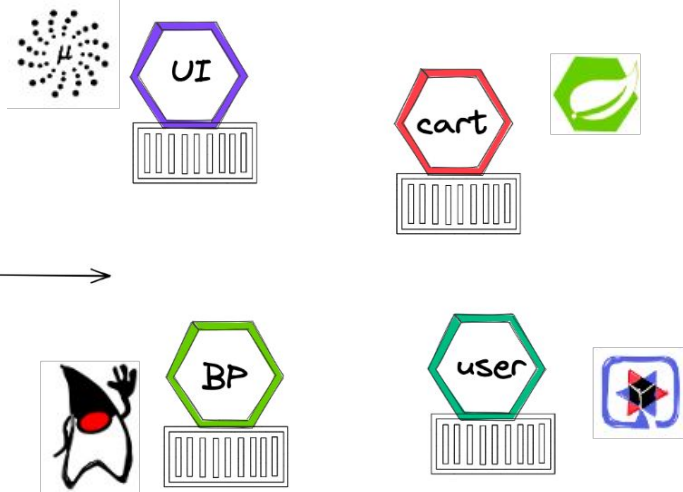
control plane



worker nodes



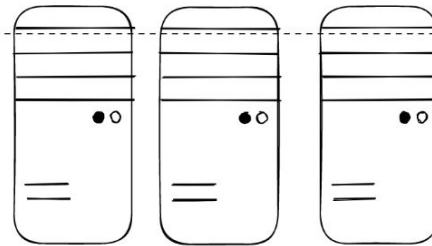
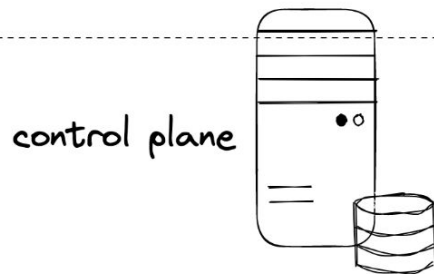
application/workloads



worker nodes

replicaset
pod
ingress
service
deployment

API Objects



worker nodes

replicaset

pod

ingress

service

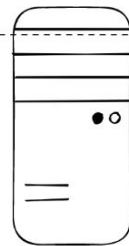
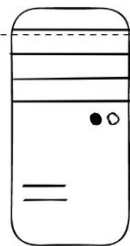
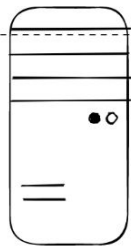
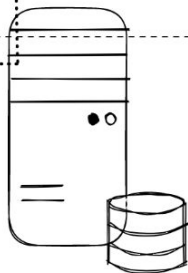
deployment

API Server

API Server

control plane

worker nodes



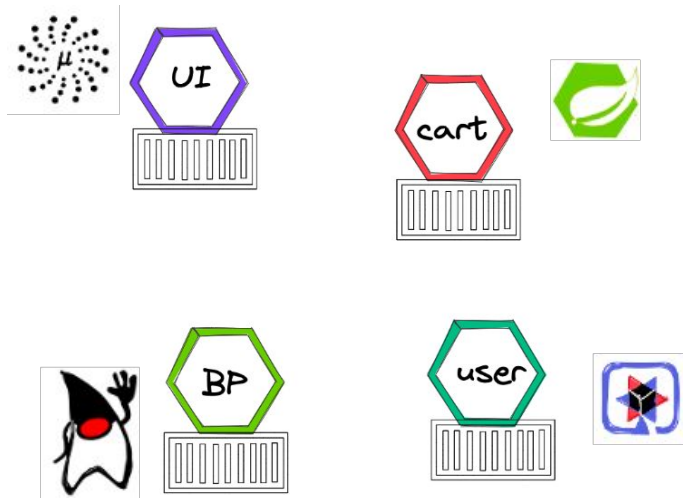
replicaset

pod

ingress

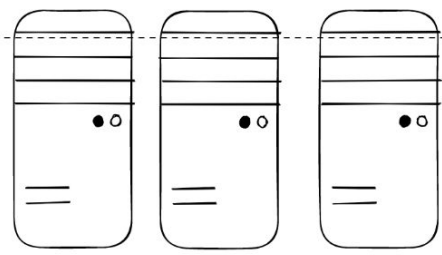
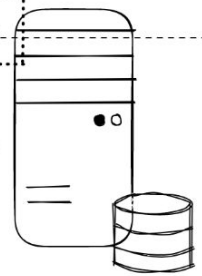
service

deployment



API Server

control plane



worker nodes



replicaset

pod

ingress

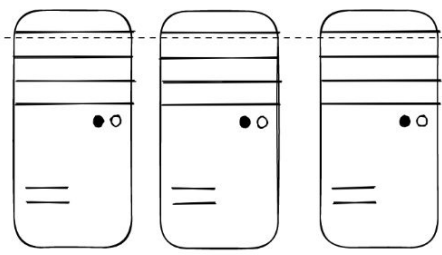
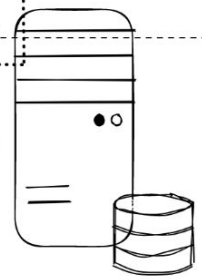
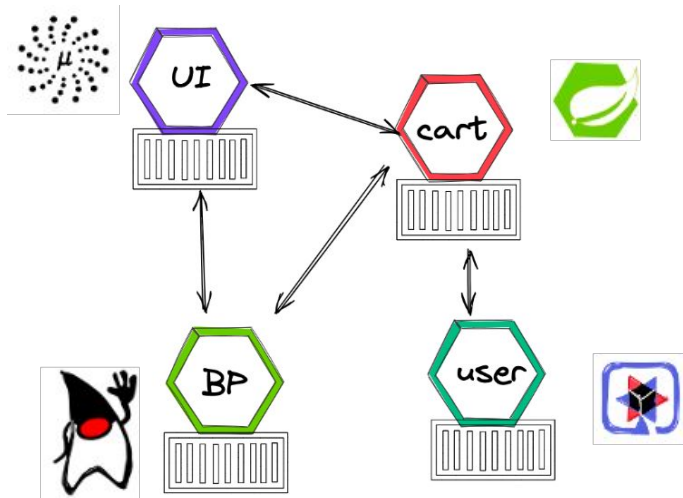
service

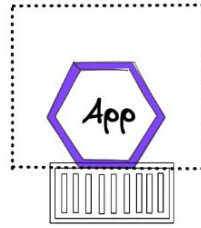
deployment

API Server

control plane

worker nodes

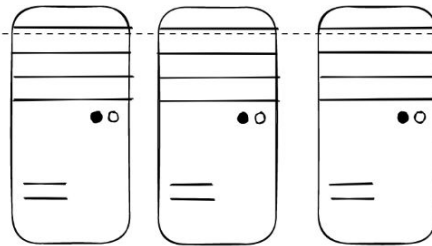
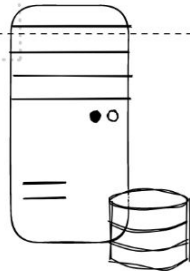




application

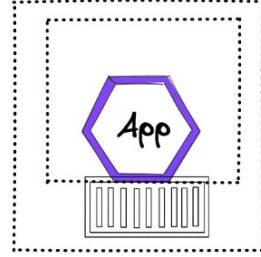


control plane

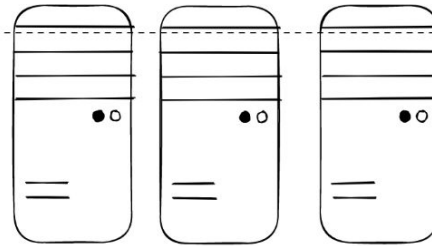
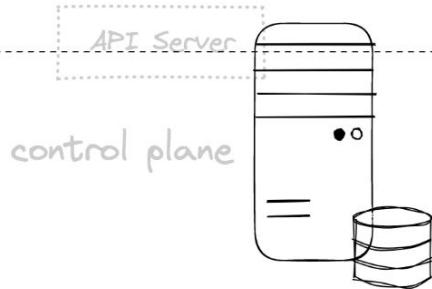


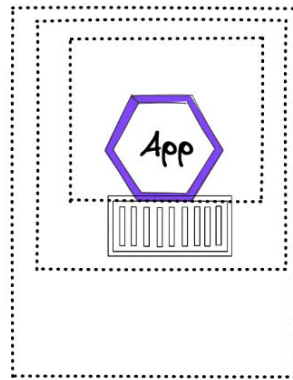
worker nodes



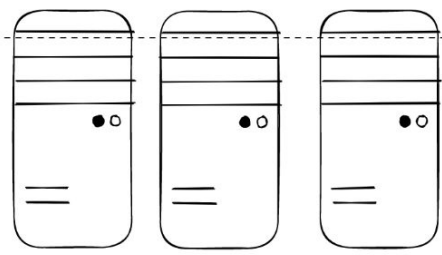
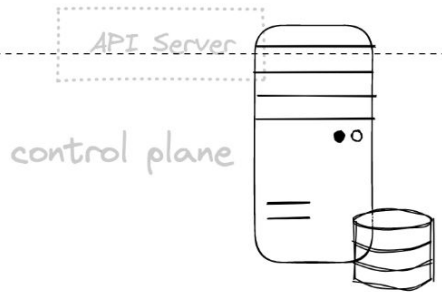


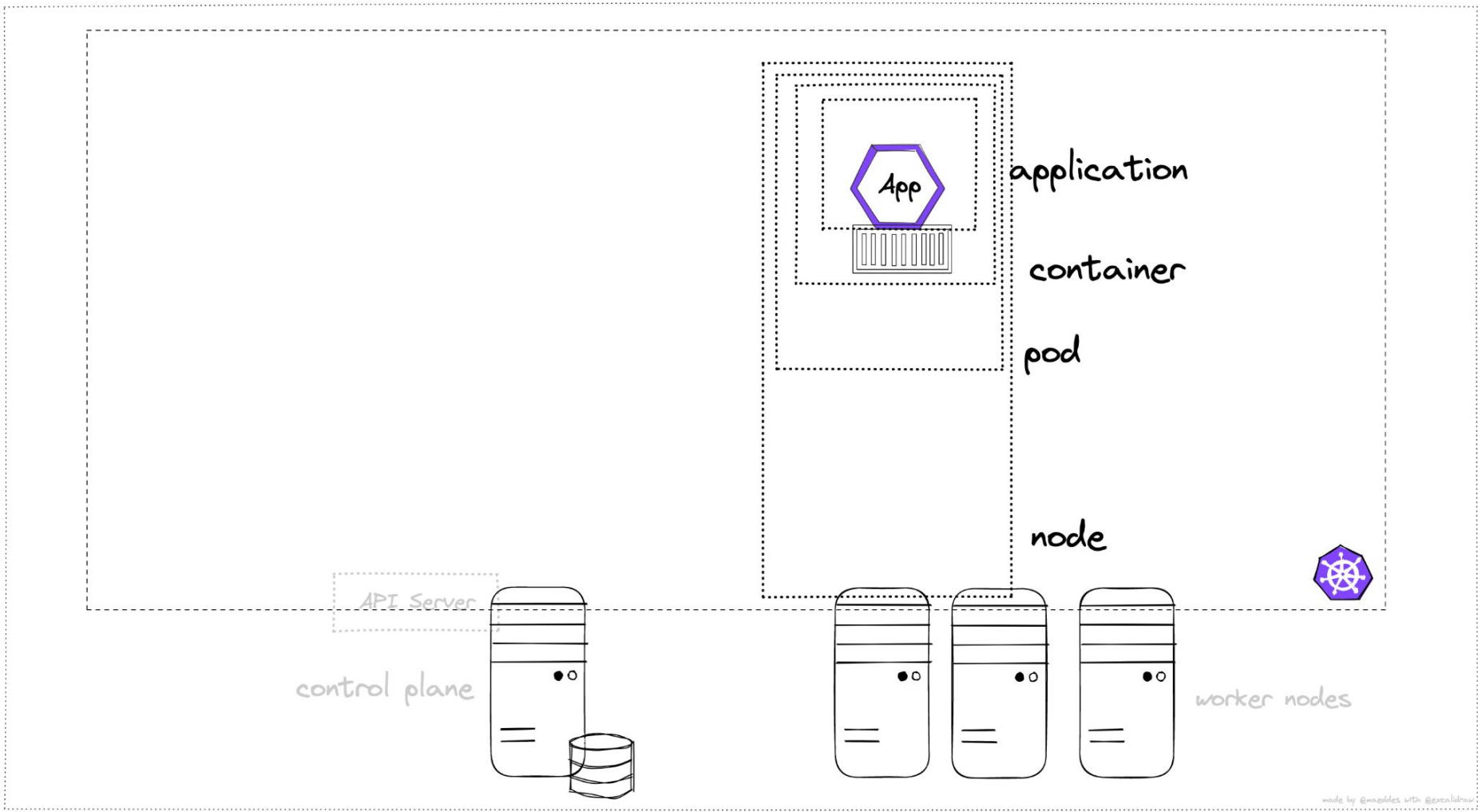
application
container





application
container
pod





observe



application

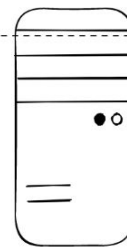
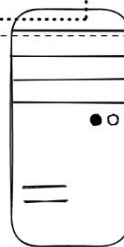
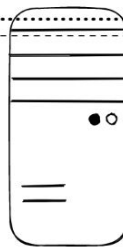
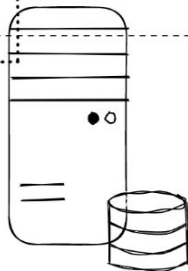
container

pod

node

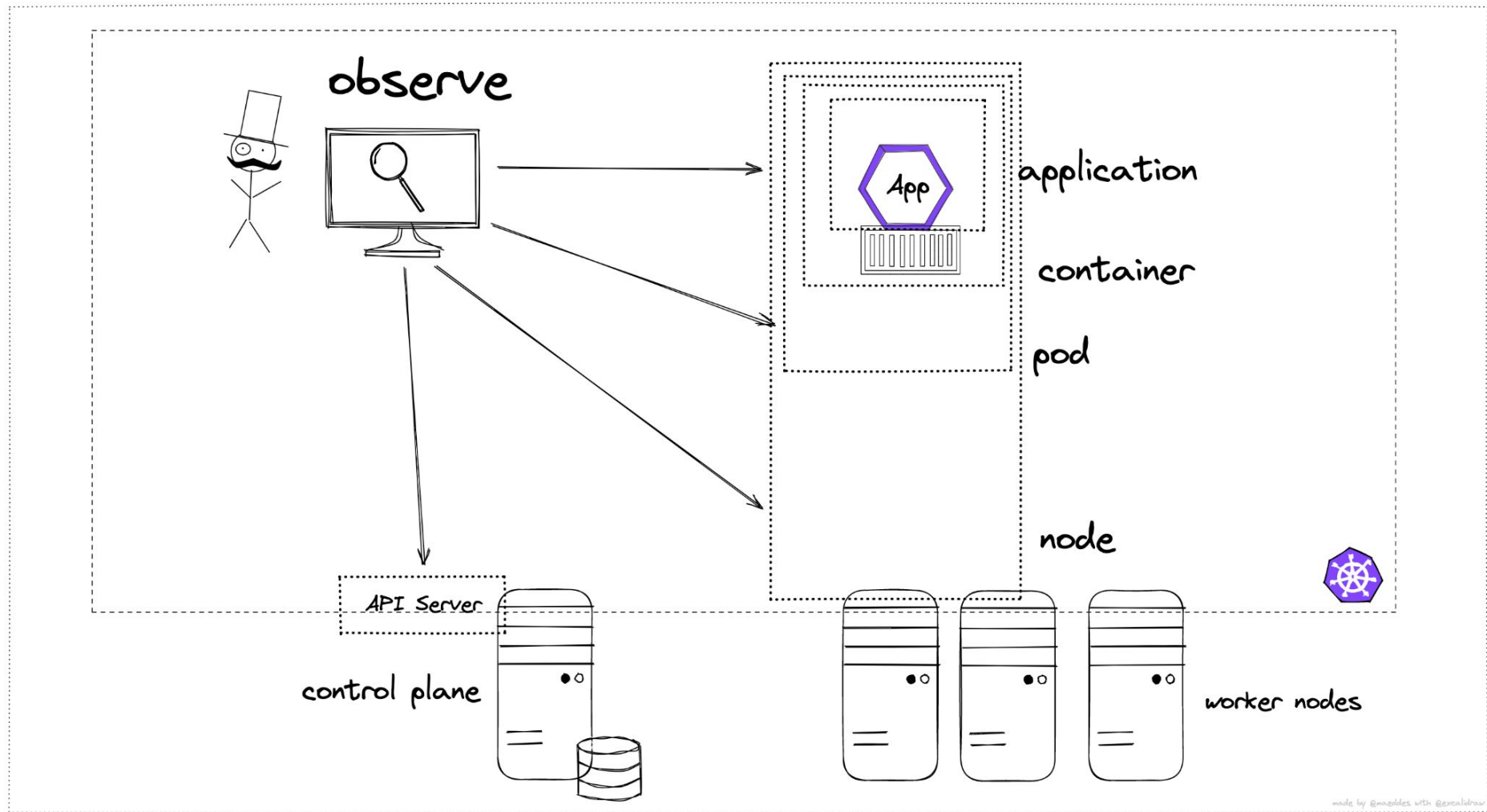
API Server

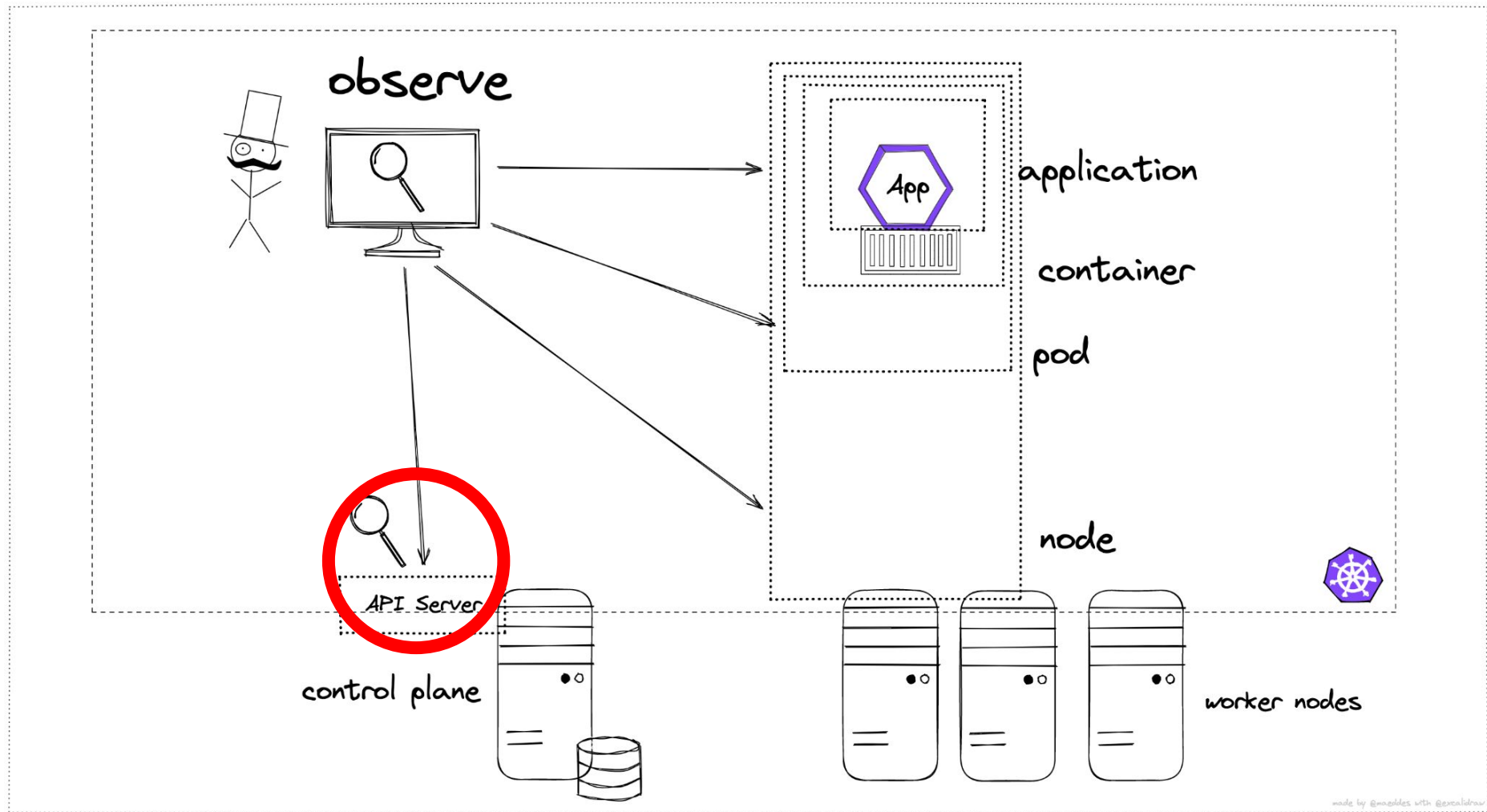
control plane

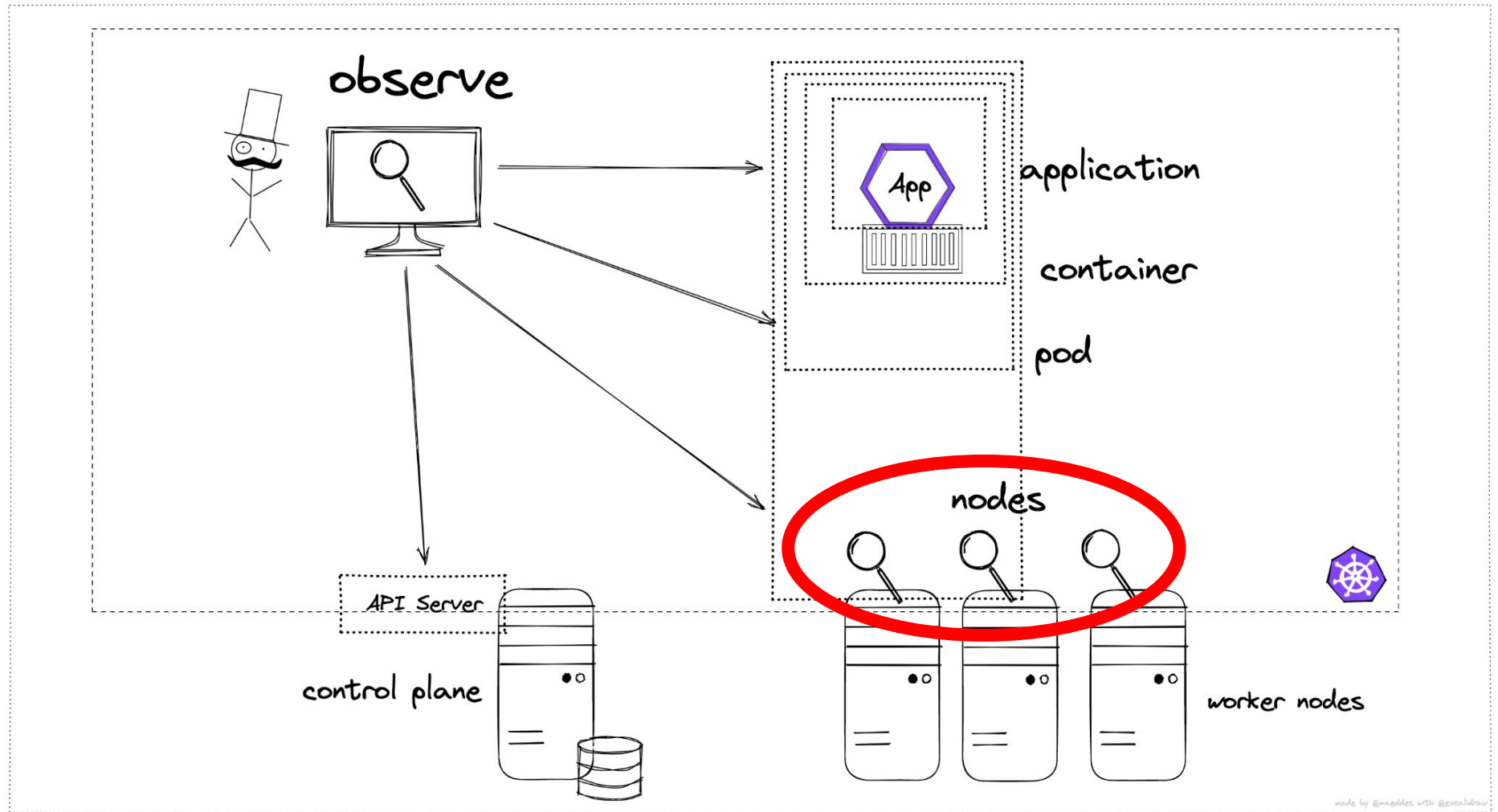


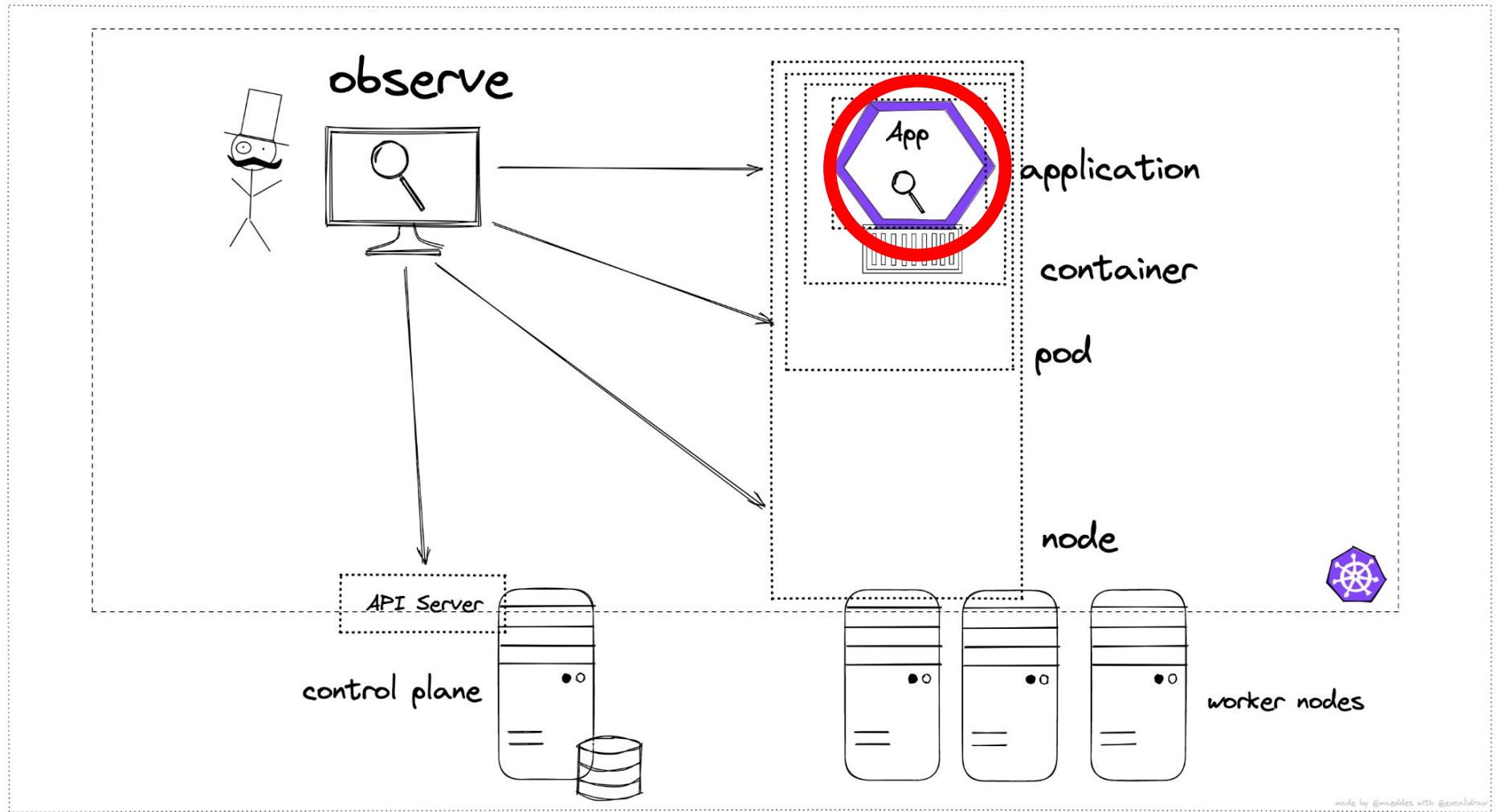
worker nodes



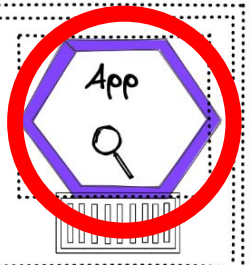








observe



application

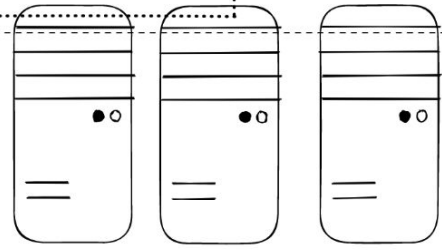
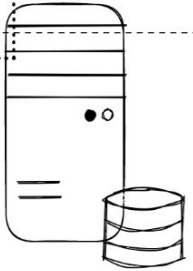
container

pod

node

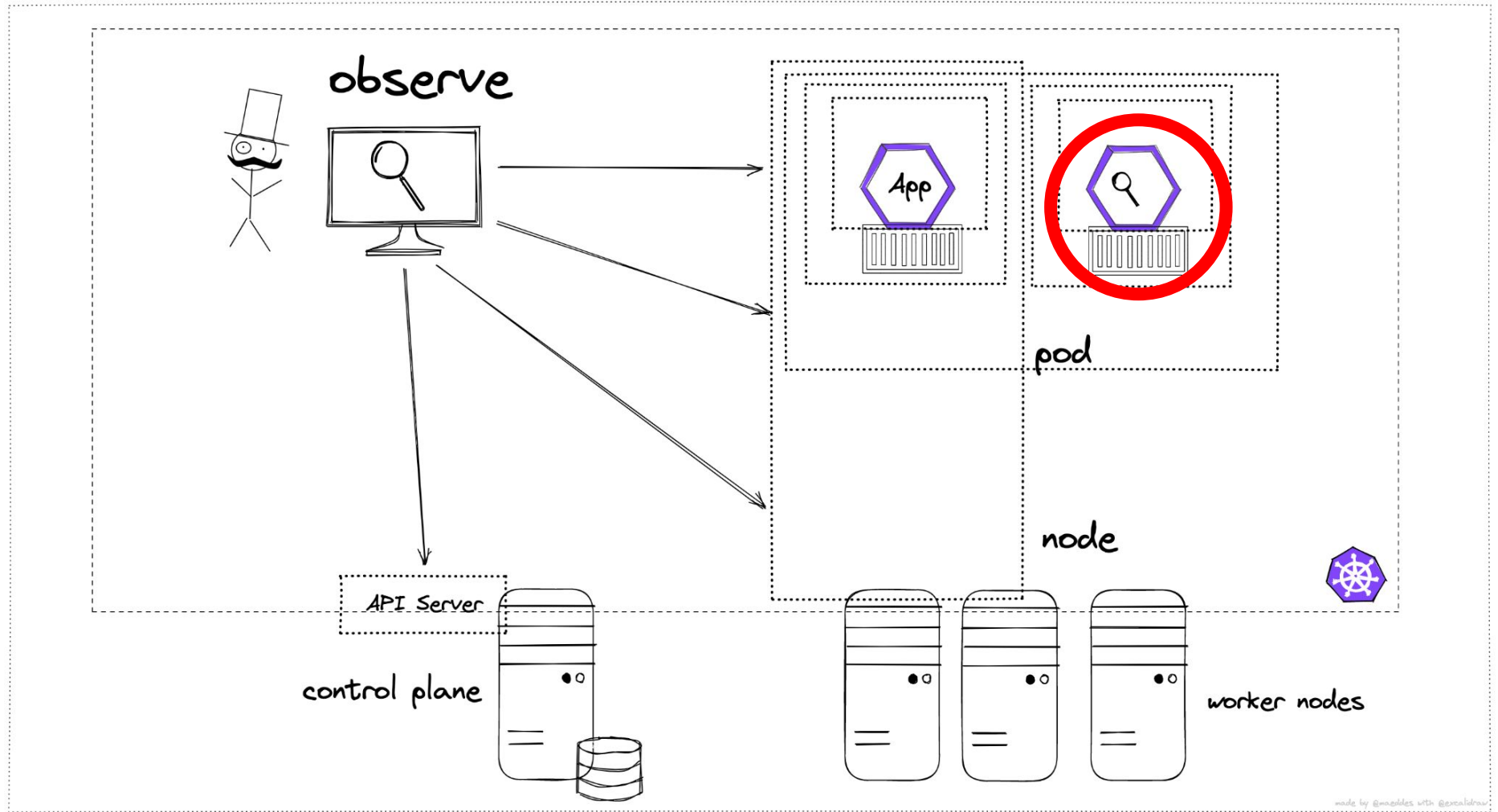
API Server

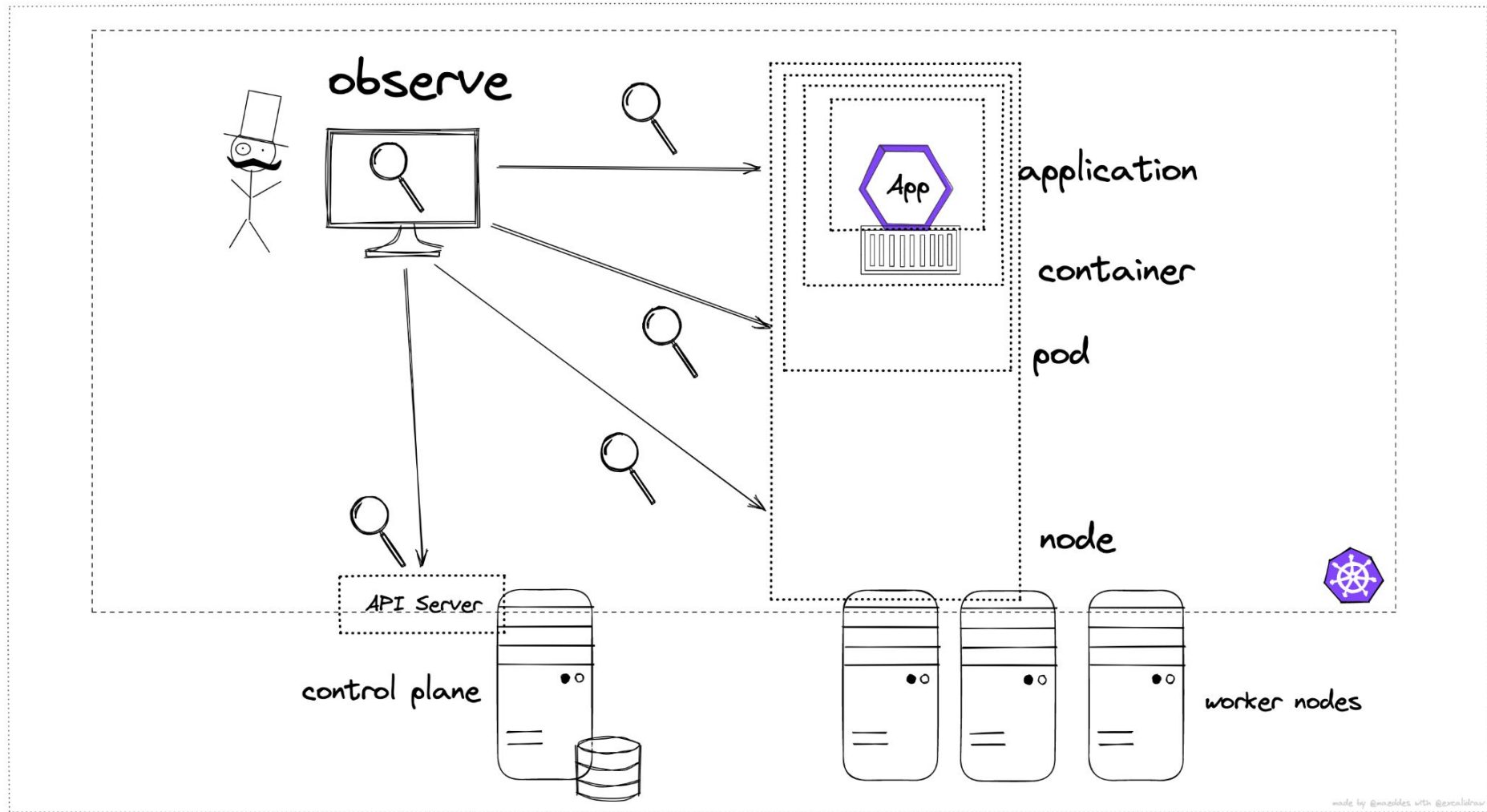
control plane

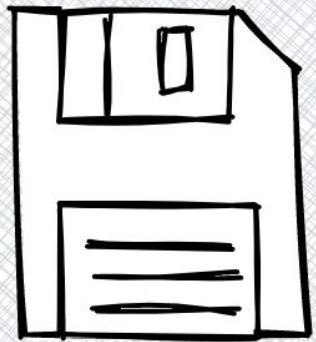


worker nodes









tools?

1 **App Definition and Development**

Database	Streaming & Messaging	Application Definition & Image Build	Continuous Integration & Delivery

2 **Orchestration & Management**

Scheduling & Orchestration	Coordination & Service Discovery	Remote Procedure Call	Service Proxy	API Gateway	Service Mesh

3 **Runtime**

Cloud Native Storage	Container Runtime	Cloud Native Network

4 **Provisioning**

Automation & Configuration	Container Registry	Security & Compliance	Key Management

Platform

Certified Kubernetes - Distribution	Certified Kubernetes - Hosted	Certified Kubernetes - Installer	PaaS/Container Service

Serverless

Members

Observability and Analysis

Monitoring	Logging

CD Foundation Landscape

Application Definition & Image Build

Database: KV, V, CockroachDB, Yugabyte, Redis, Aerospike, ScyllaDB, MemSQL, InfluxDB, TimescaleDB, etc.

Streaming & Messaging: CloudEvents, NATS, Apache Kafka, Apache Pulsar, etc.

Application Definition & Image Build: HELM, Backstage, Buildpacks.io, Kubernetes, Docker, etc.

Continuous Integration & Delivery: Argo, Flux, Tekton, OpenShift, etc.

Orchestration & Management

Scheduling & Orchestration: Kubernetes, OpenShift, Volcano, etc.

Coordination & Service Discovery: CoreDNS, etcd, etc.

Remote Procedure Call: gRPC, etc.

Service Proxy: Envoy, etc.

API Gateway: Kong, etc.

Service Mesh: Istio, etc.

Runtime

Cloud Native Storage: MinIO, etc.

Container Runtime: containerd, etc.

Cloud Native Network: Cilium, etc.

Provisioning

Automation & Configuration: Ansible, etc.

Container Registry: Harbor, etc.

Security & Compliance: Falco, etc.

Key Management: etc.

Platform

Certified Kubernetes - Distribution: AWS, etc.

Certified Kubernetes - Hosted: AWS, etc.

Certified Kubernetes - Installer: AWS, etc.

PaaS/Container Service: etc.

Observability and Analysis

Monitoring: Grafana, Prometheus, etc.

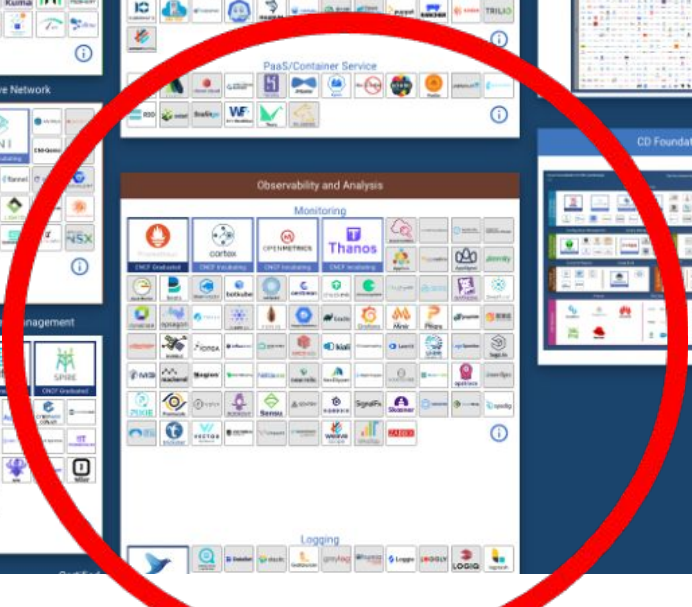
Logging: etc.

Serverless

CD DevOps Landscape

Members

CD Foundation Landscape





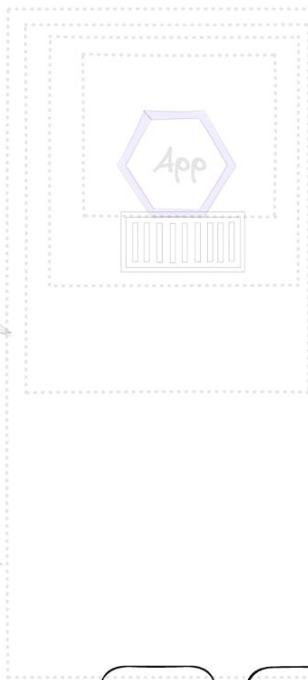
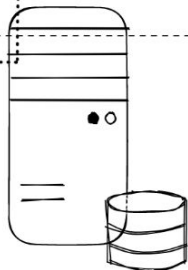
Kubernetes API

observe



API Server

control plane

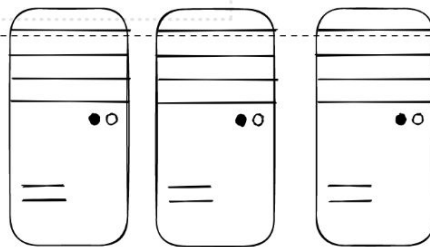


application

container

pod

node



worker nodes

observe

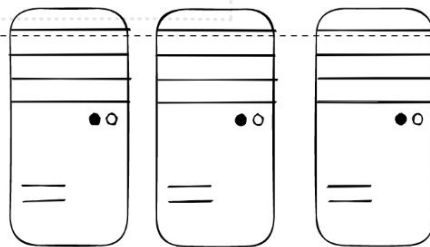
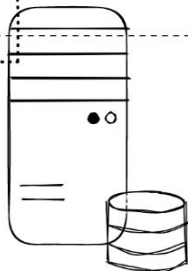


```
kubectl get ...  
kubectl describe ...  
kubectl logs ...  
kubectl debug ...  
kubectl exec ...
```



API Server

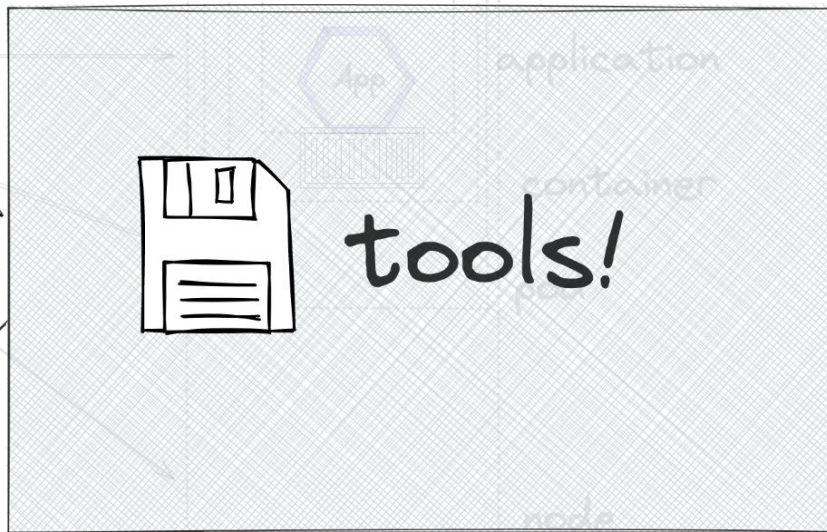
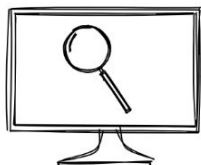
control plane



worker nodes

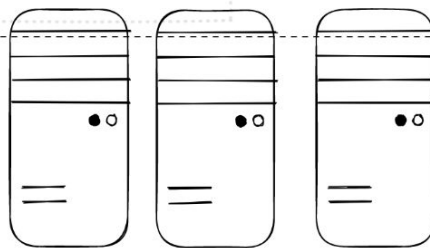
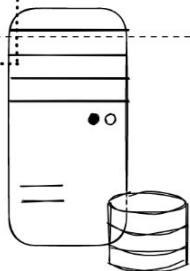


observe



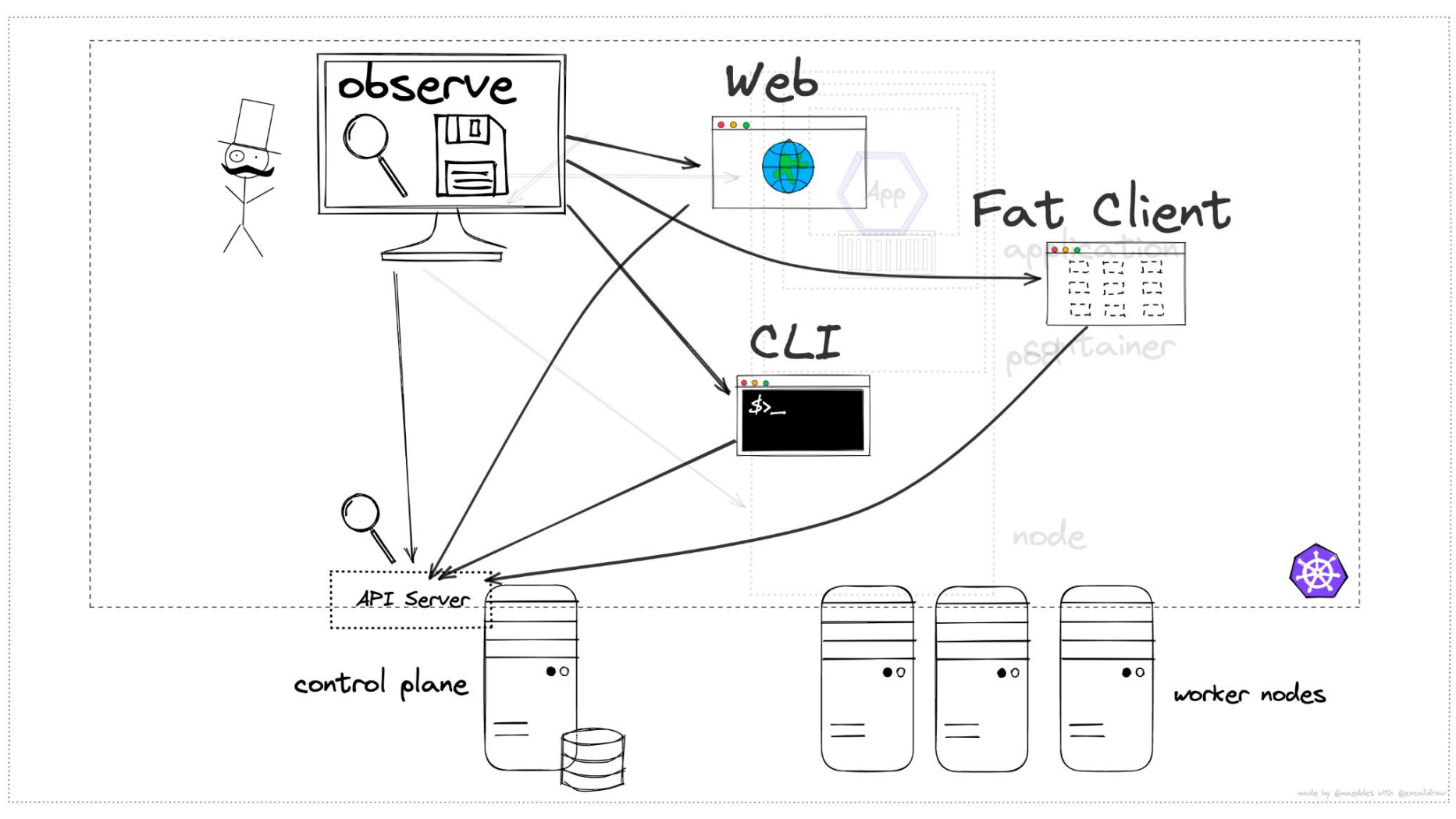
API Server

control plane



worker nodes





observe

Web

Fat Client

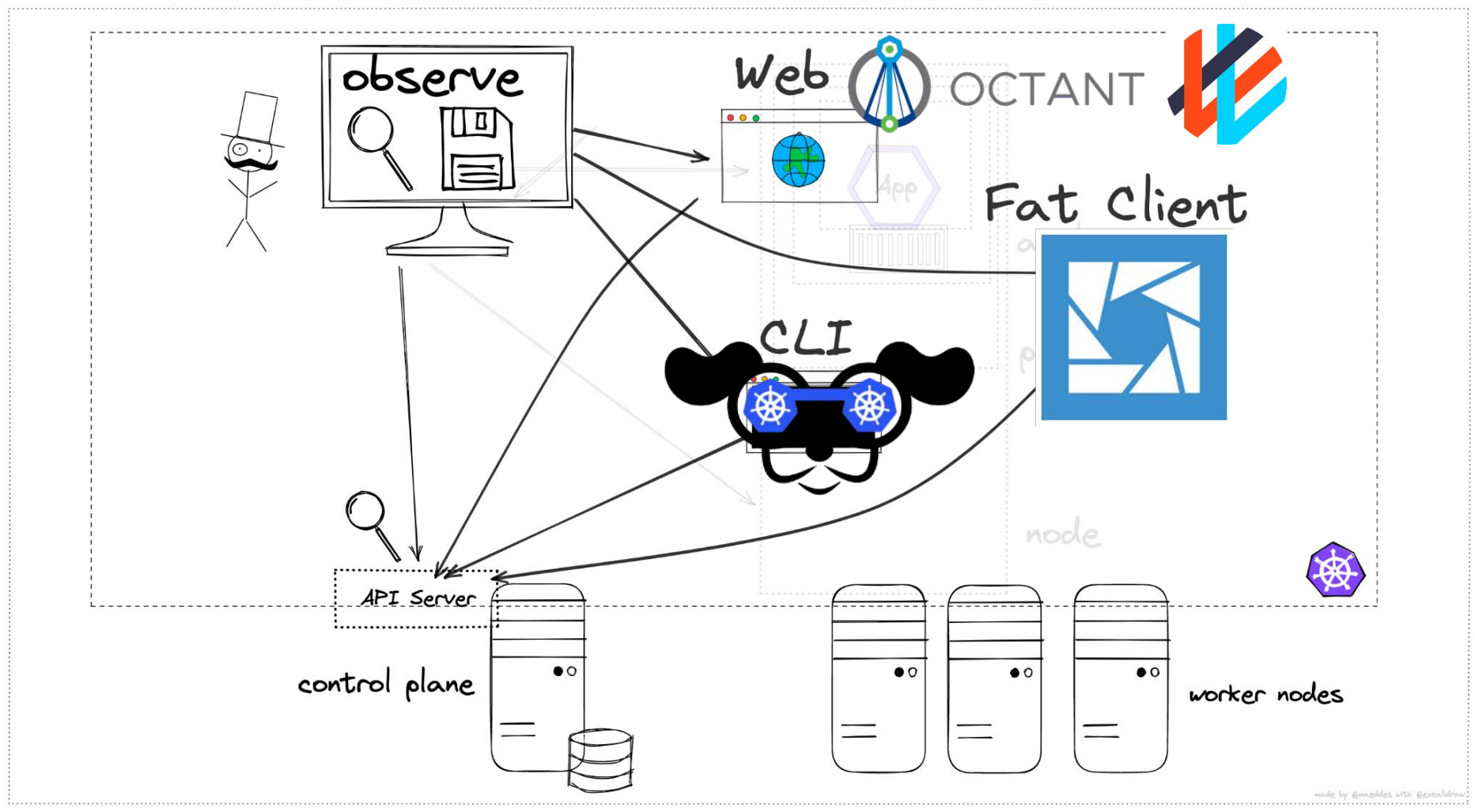
API Server

CLI

control plane

worker nodes





Officially deprecated



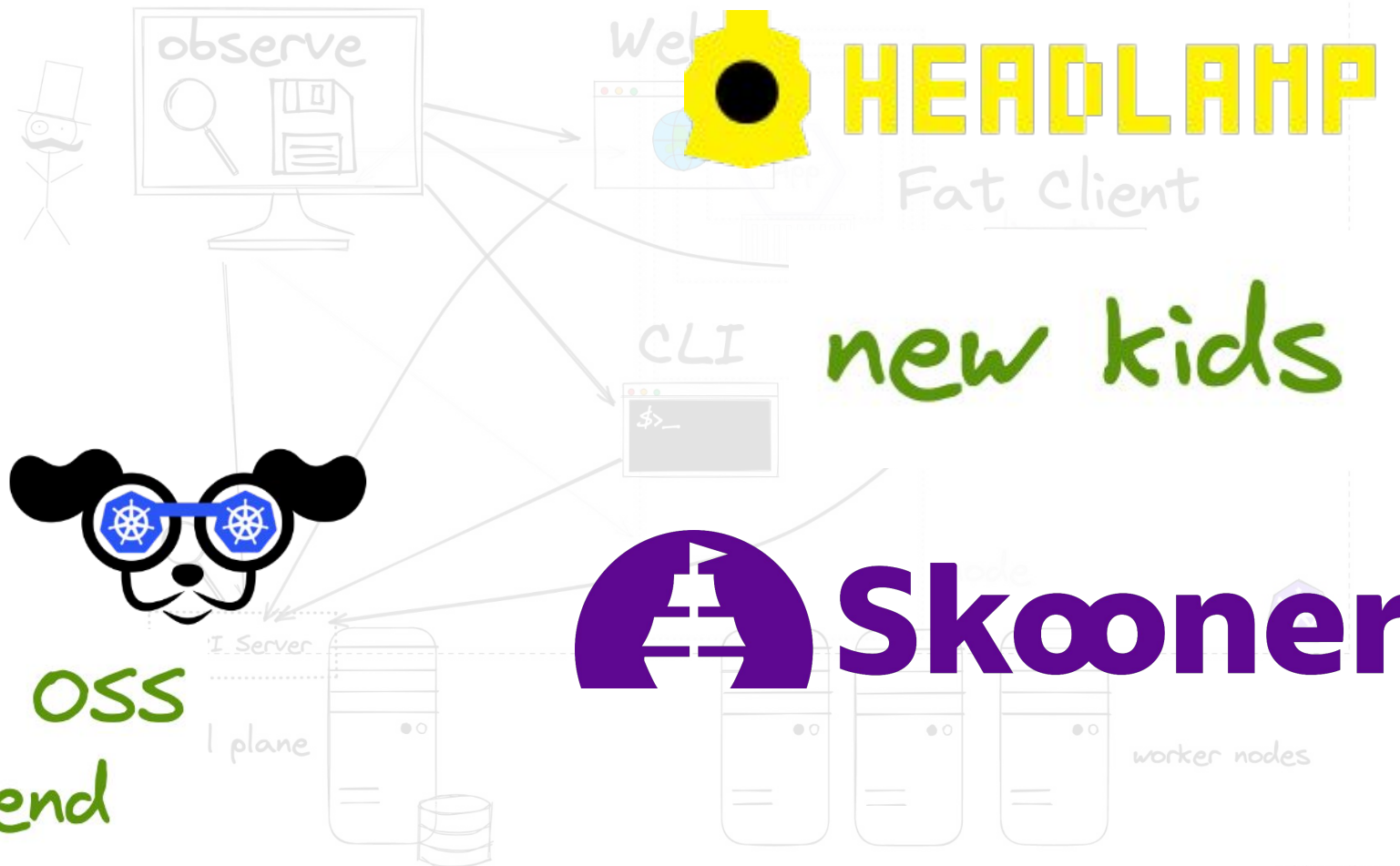
OCTANT

"Less" Activity



good OSS friend





HEADLAMP

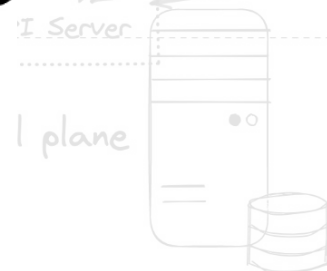
Fat Client



new kids



Skooner



worker nodes

good OSS friend

```

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>



Applications



Namespace Overview



Cluster Overview



Plugins

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

Namespace Overview > Workloads > Pods

Pods

Pods

	Name	Labels	Ready	Phase	Status	Restarts	Node	Age
⋮	api-gateway-79545bcd86-djptt	app:api-gateway security.istio.io/tlsMo...	2/2	Running	Running	0	aks-nodepool1-26169201-vmss00000r	6m
⋮	customers-db-mysql-0	app.kubernetes.io/...	2/2	Running	Running	0	aks-nodepool1-26169201-vmss00000r	6m
⋮	customers-service-7b45c4f5f7-nxhsw	app:customers-ser... security.istio.io/tls...	2/2	Running	Running	1	aks-nodepool1-26169201-vmss00000r	6m
⋮	vets-db-mysql-0	app.kubernetes.io/...	2/2	Running	Running	0	aks-nodepool1-26169201-vmss00000r	6m
⋮	vets-service-79dbb4bd8b-s85dz	app:vets-service security.istio.io/tlsMo...	2/2	Running	Running	1	aks-nodepool1-26169201-vmss00000r	6m
⋮	vets-service-v02-5c4f69749f-ag7dm	app:vets-service security.istio.io/tlsMo...	2/2	Running	Running	1	aks-nodepool1-26169201-vmss00000r	6m
⋮	visits-db-mysql-0	app.kubernetes.io/...	2/2	Running	Running	0	aks-nodepool1-26169201-vmss00000r	6m
⋮	visits-service-6d59cc5f9f-v6pt9	app:visits-service security.istio.io/tlsMo...	2/2	Running	Running	2	aks-nodepool1-26169201-vmss00000r	6m
⋮	wavefront-proxy-6b58d6f557-64bcm	app:wavefront-pro... security.istio.io/tls...	1/2	Running	CrashLoopBackOff	5	aks-nodepool1-26169201-vmss00000r	6m

Items per page 10 1 - 9 of 9 Items



Preferences

Lens < mhsdemo

Cluster Nodes Workloads

Overview Pods Deployments DaemonSets StatefulSets ReplicaSets Jobs CronJobs

Pods 12 items Namespaces: default, spring- Search...

Name	Namespace	Containers	Restarts	Controlled By	Node	QoS	Age	Status
aks-helloworld-one-56c7b8d79d-tvm5s	default	1	0	ReplicaSet	aks-nodepool1-2616f	BestEffort	59d	Running
aks-helloworld-two-58bbb47f58-hftb8	default	1	0	ReplicaSet	aks-nodepool1-2616f	BestEffort	59d	Running
api-gateway-79545bcd86-djptt	spring-petclinic	3	0	ReplicaSet	aks-nodepool1-2616f	Burstable	9m31s	Running
customers-db-mysql-0	spring-petclinic	3	0	StatefulSet	aks-nodepool1-2616f	Burstable	9m33s	Running
customers-service-7b45c4f5f7-nxhsw	spring-petclinic	3	1	ReplicaSet	aks-nodepool1-2616f	Burstable	9m31s	Running
ingress-nginx-controller-756f546d89-6xb...	default	1	0	ReplicaSet	aks-nodepool1-2616f	Burstable	59d	Running
vets-db-mysql-0	spring-petclinic	3	0	StatefulSet	aks-nodepool1-2616f	Burstable	9m33s	Running
vets-service-79dbb4bd8b-s85dz	spring-petclinic	3	1	ReplicaSet	aks-nodepool1-2616f	Burstable	9m32s	Running
vets-service-v02-5c4f69749f-dg7dm	spring-petclinic	3	1	ReplicaSet	aks-nodepool1-2616f	Burstable	9m32s	Running
visits-db-mysql-0	spring-petclinic	3	0	StatefulSet	aks-nodepool1-2616f	Burstable	9m33s	Running
visits-service-6d59cc5f9f-v6pt9	spring-petclinic	3	2	ReplicaSet	aks-nodepool1-2616f	Burstable	9m31s	Running
wavefront-proxy-6b58d6f557-64bcm	spring-petclinic	3	6	ReplicaSet	aks-nodepool1-2616f	Burstable	9m32s	CrashLoopB...

CLUSTER

Namespaces

Nodes

CRDs

WORKLOADS

STORAGE

NETWORK

SECURITY

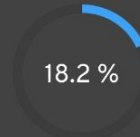
CONFIGURATION

CPU Usage



0.63 / 16 units

Memory Usage



22.82 / 125.61 GB

Pods



Events Warnings



Type	Name	Namespace	Reason	Age
Pod	oteldemo-featureflagservice-84b8575847-mp87z	otel	Pulled	7m
Ingress	otel	otel	Sync	46m
Pod	ingress-nginx-controller-6b8bfd7f69-fngth	ingress-nginx	RELOAD	1h
Pod	oteldemo-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



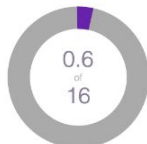
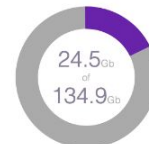
ACCOUNTS

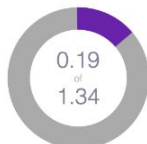


PROFILE








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-6b8bfd7f69-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

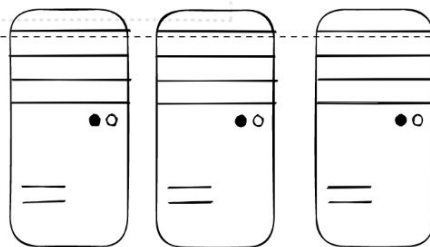
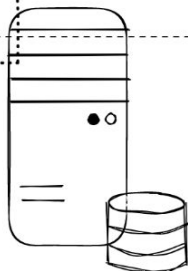


```
kubectl get ...  
kubectl describe ...  
kubectl logs ...  
kubectl debug ...  
kubectl exec ...
```



API Server

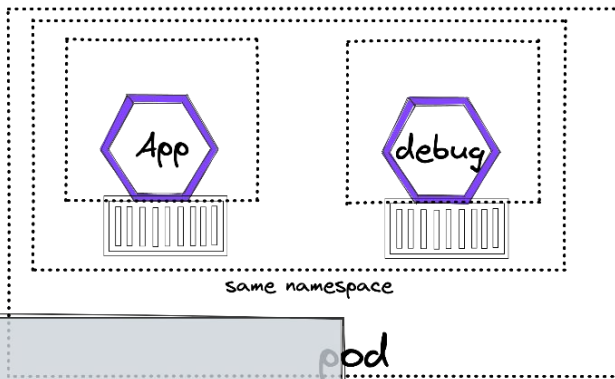
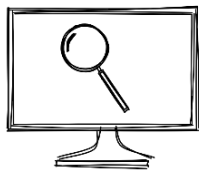
control plane



worker nodes



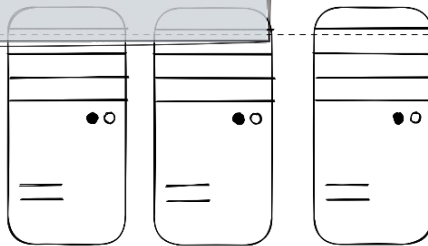
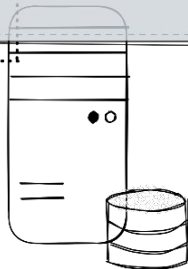
observe



```
kubectl debug  
-it pod/name --image=maeddes/universal  
--target=container-name
```

API Server

control plane



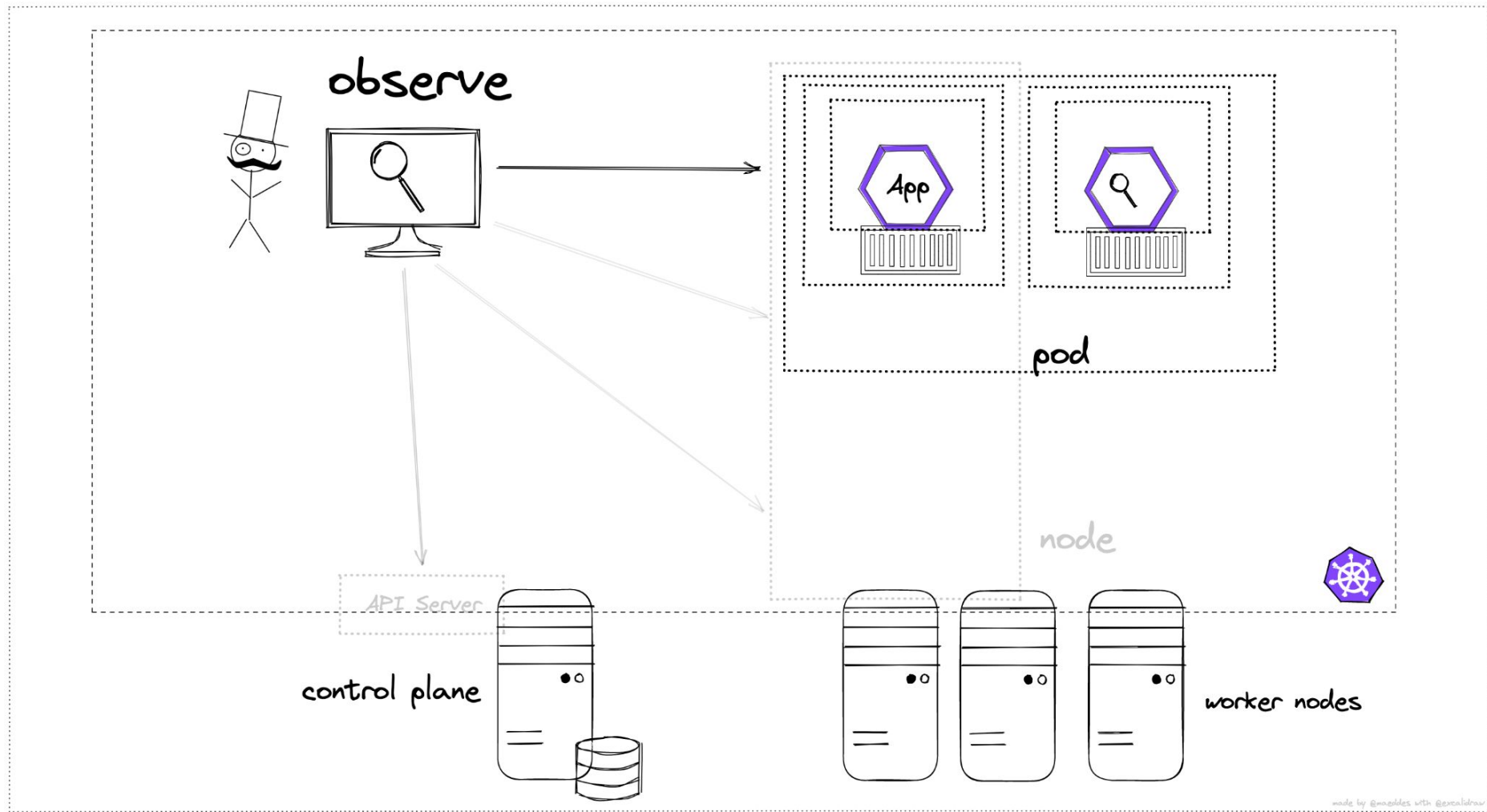
worker nodes



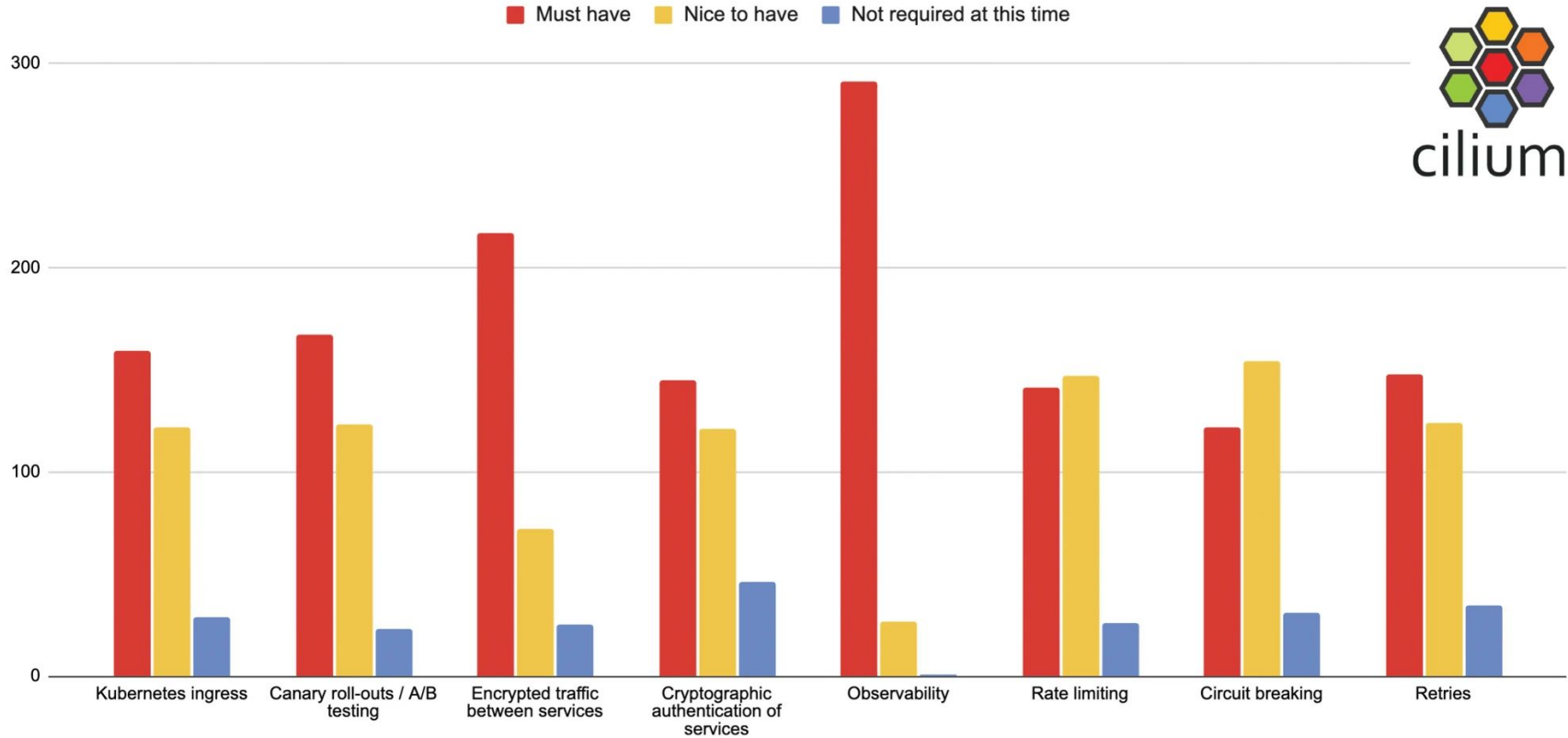
Characteristics:

- no change to cluster or apps required
- non-intrusive
- helpful for understanding K8s
and getting overview
- no network insights

pod-based
(Service Mesh)



What features of a Service Mesh interest you most?



replica/stateful/daemon set



pod



replica/stateful/daemon set



pod

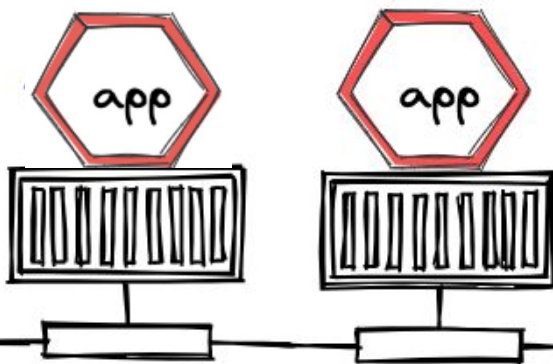


network traffic

replica/stateful/daemon set



pod



network traffic

multiple containers in pod share network

replica/stateful/daemon set



pod

proxy



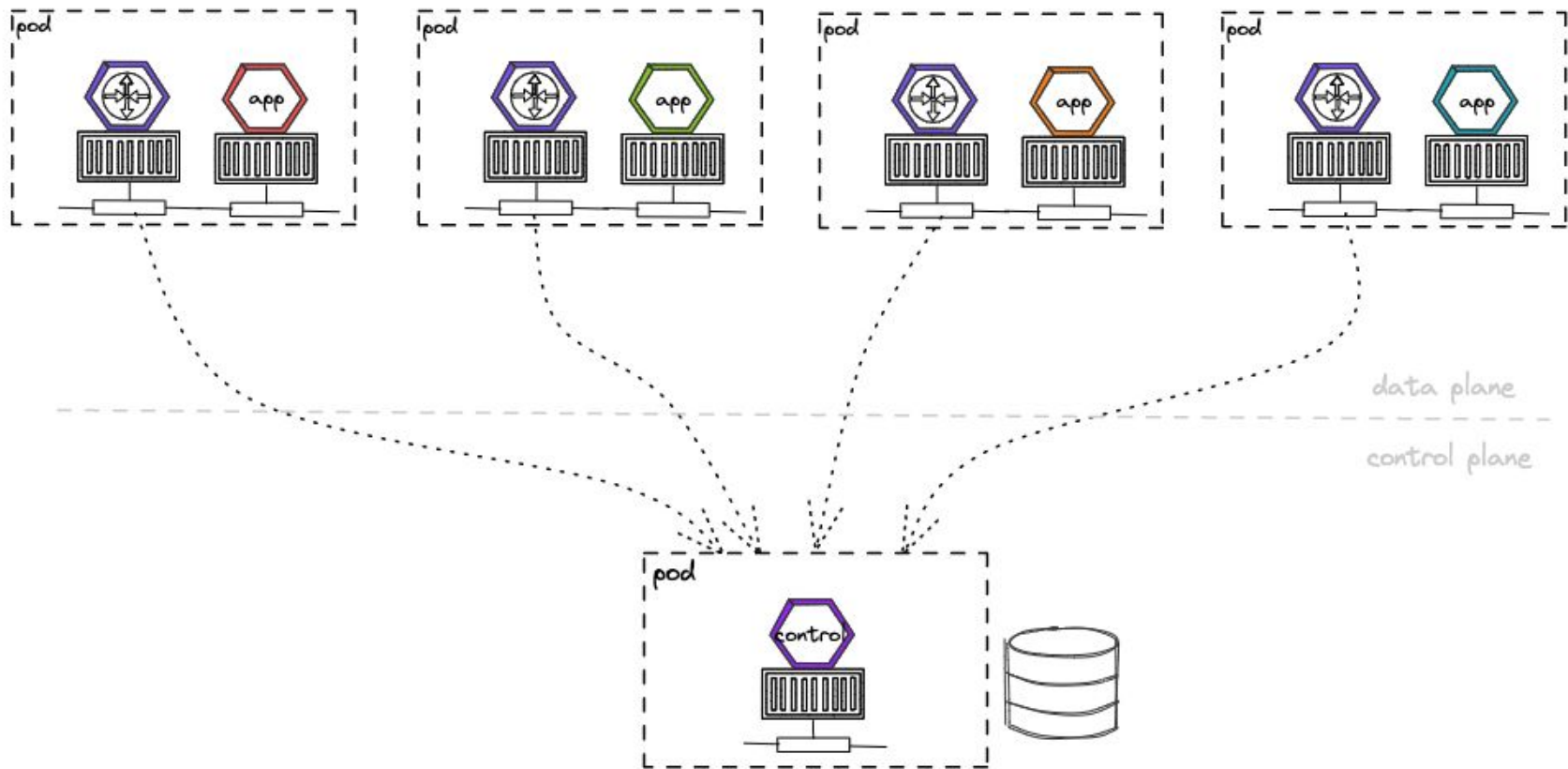
app

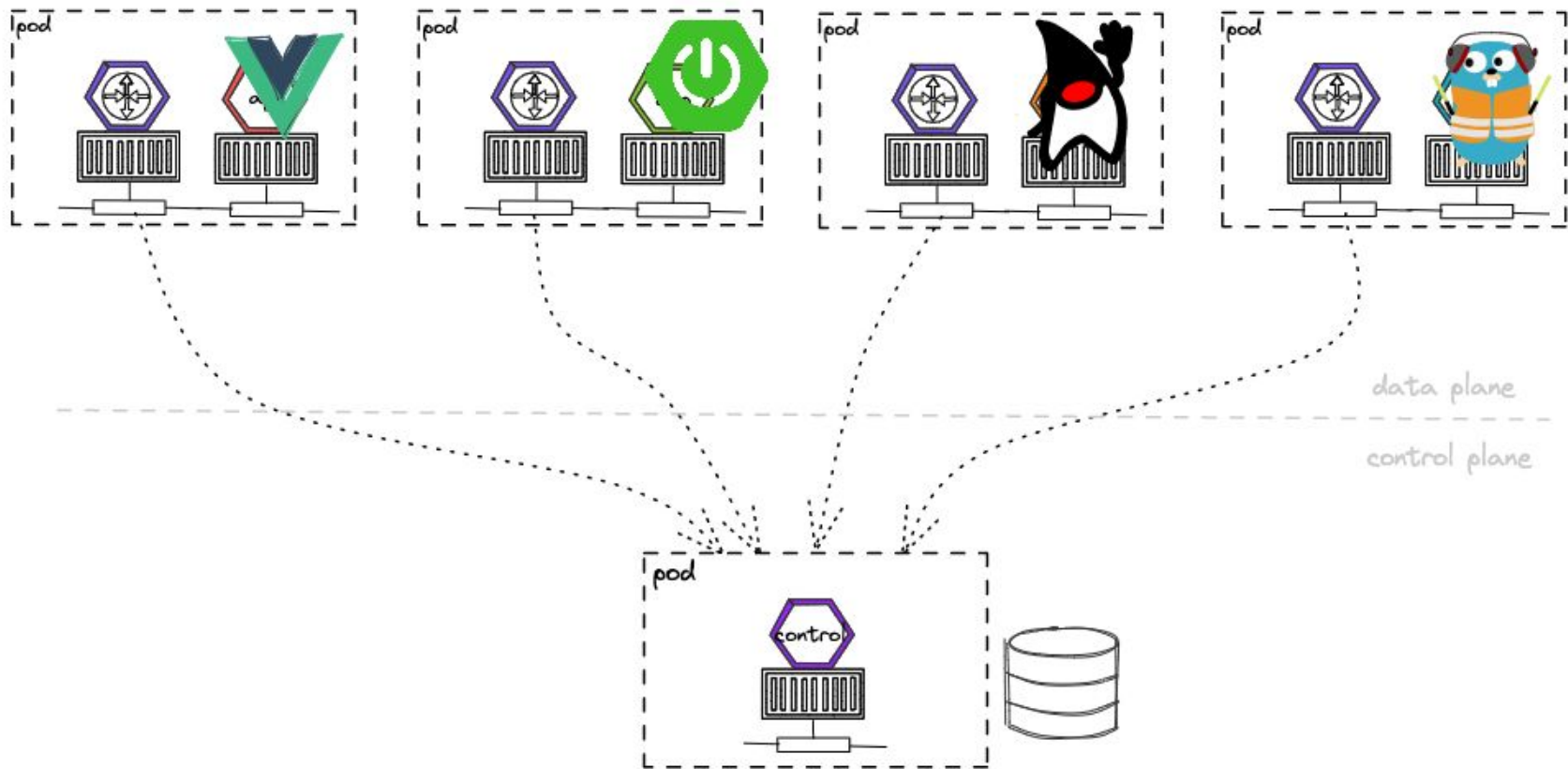


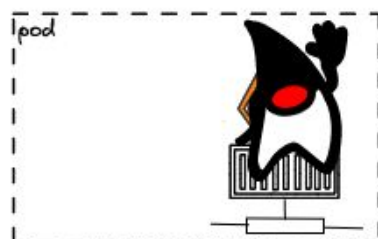
network traffic



multiple containers in pod share network

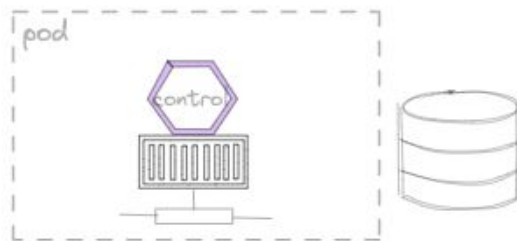


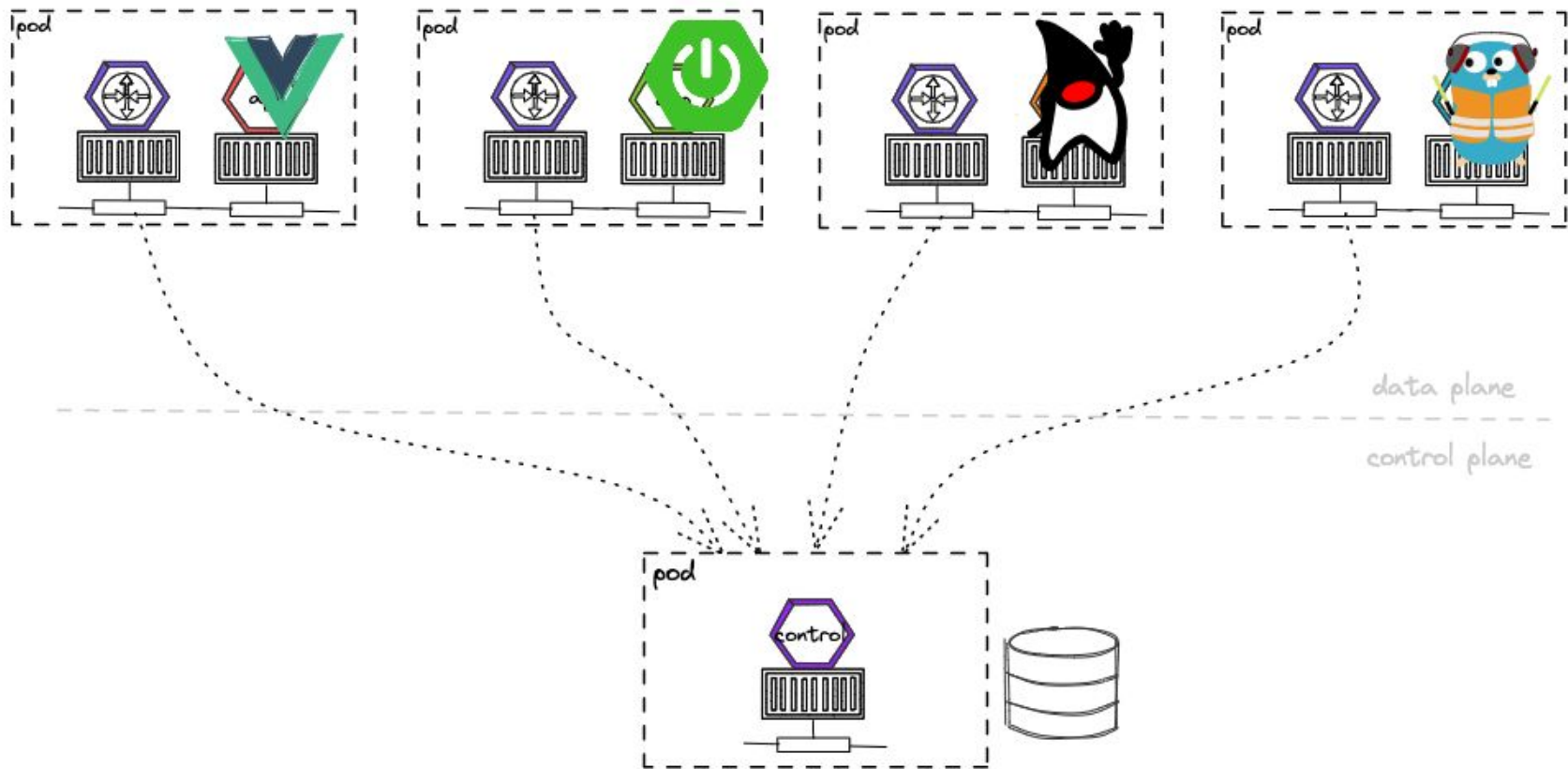


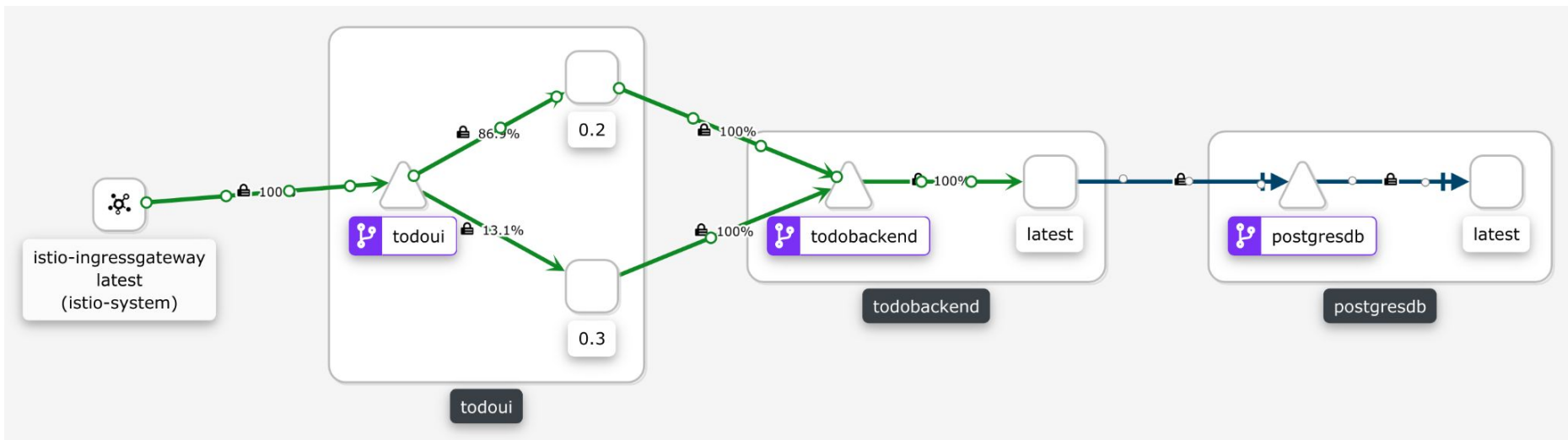


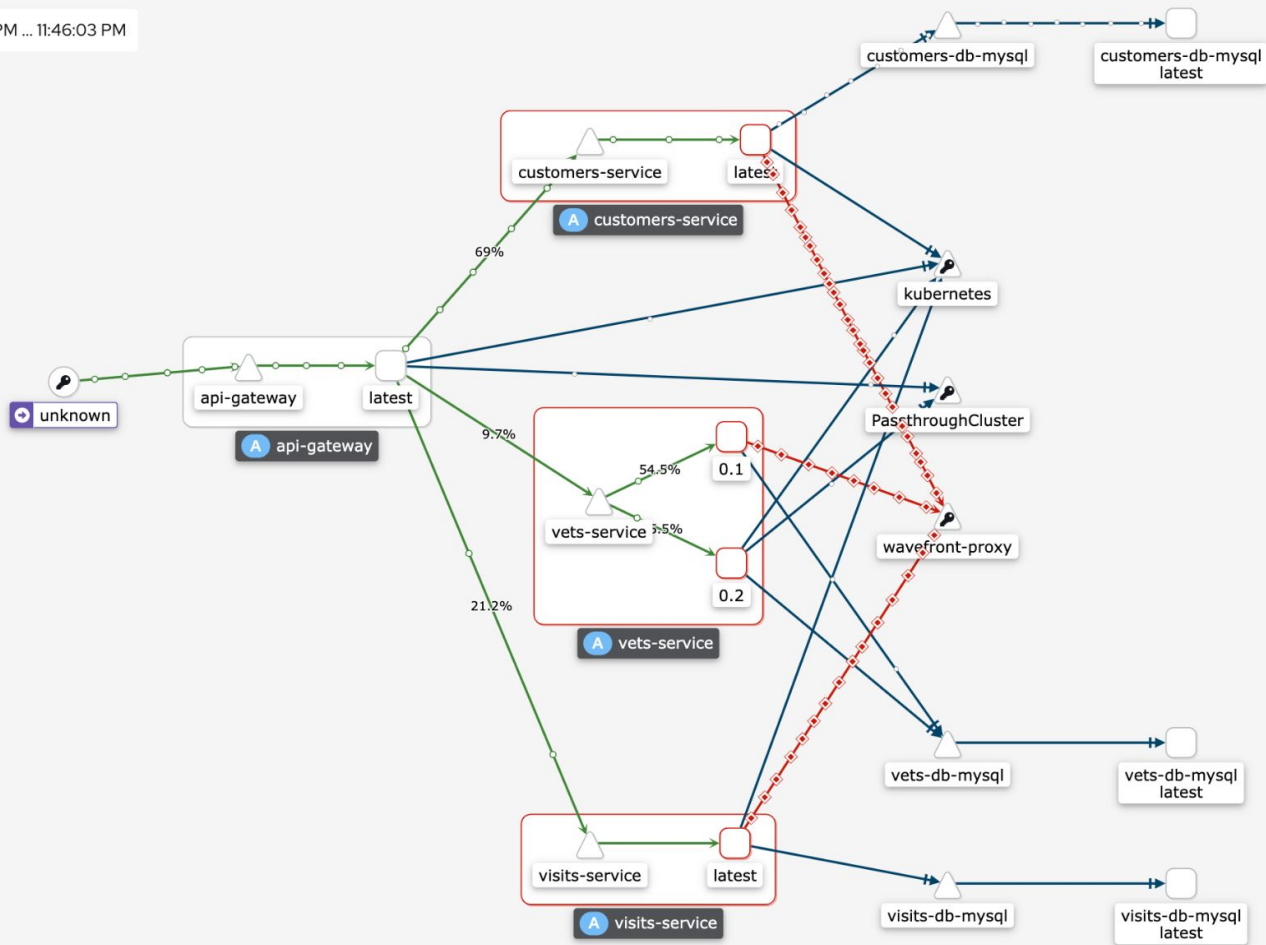
data plane

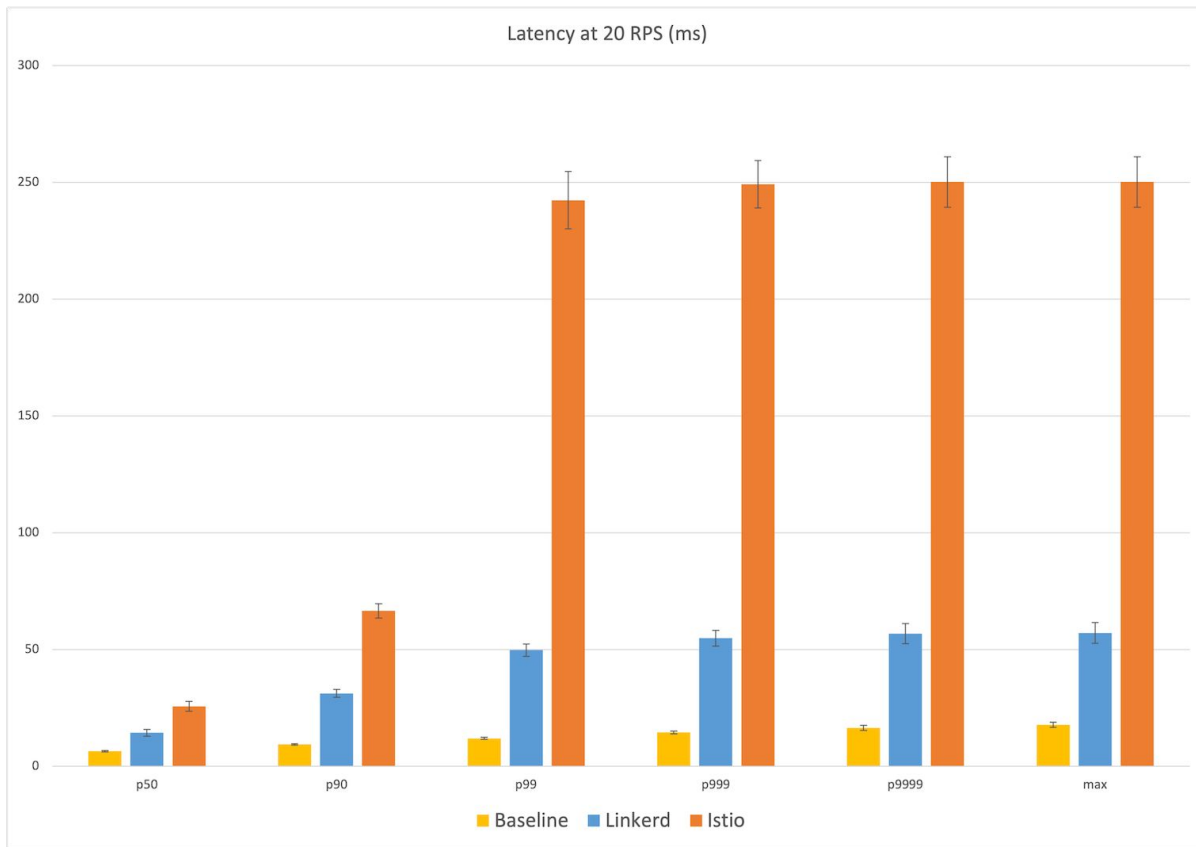
control plane











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

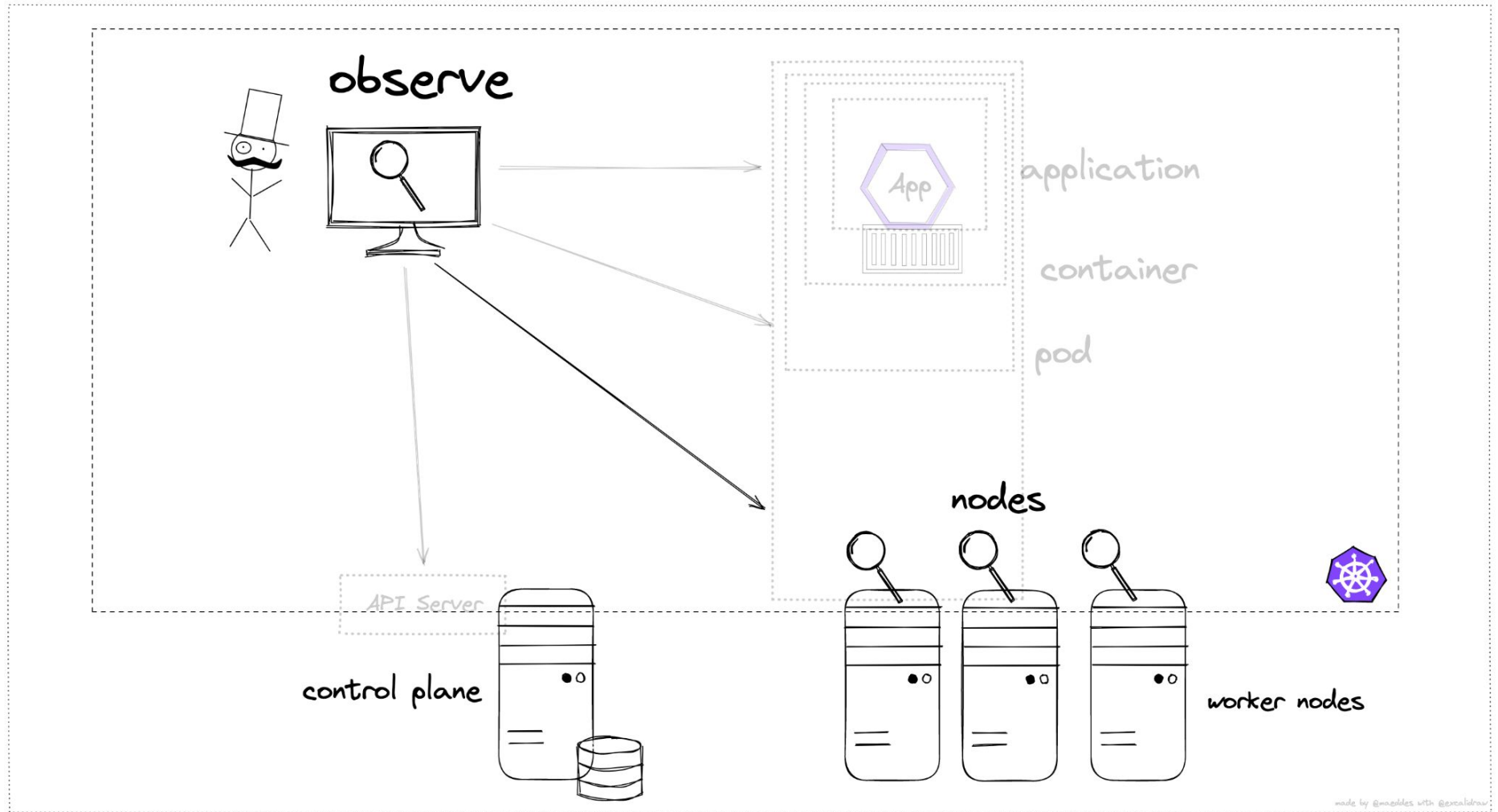
Characteristics:

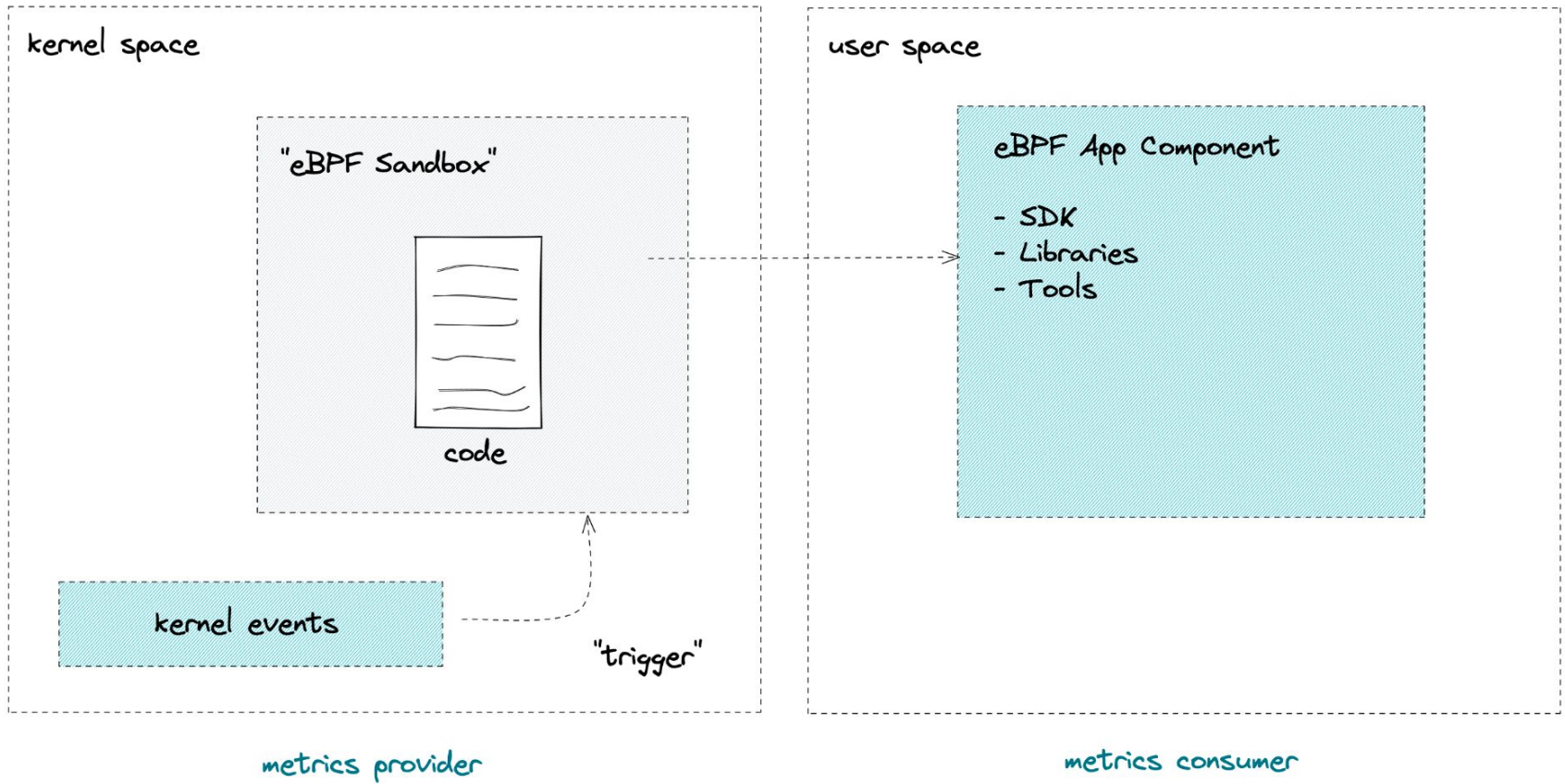
- extends Kubernetes for limitations in traffic awareness and shaping capabilities
- concept of injecting sidecar-proxy into each pod to have control over entire network flow
- no application level metrics
- configuration changes at runtime

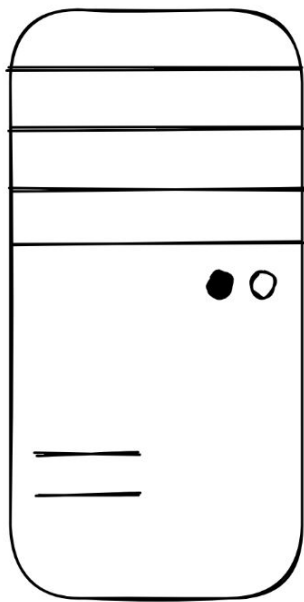
node-based
(eBPF)



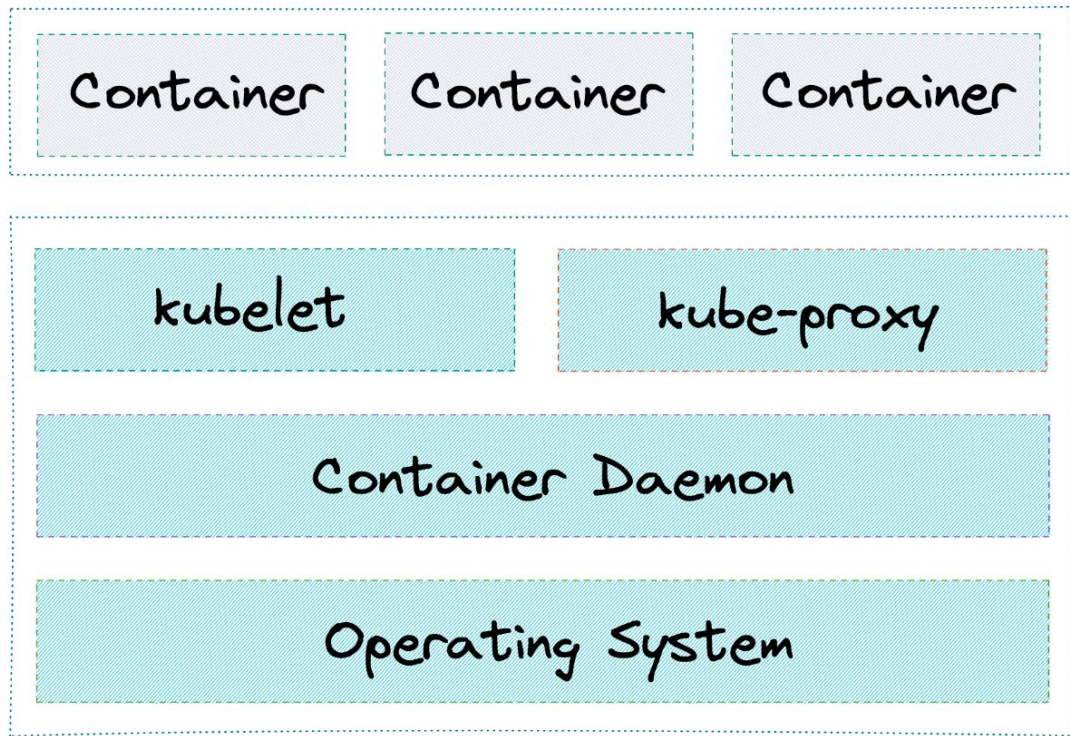
extended Berkeley Packet Filter

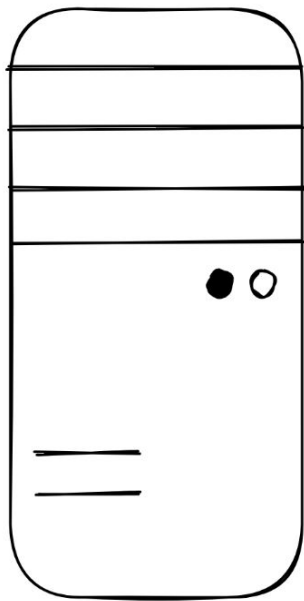




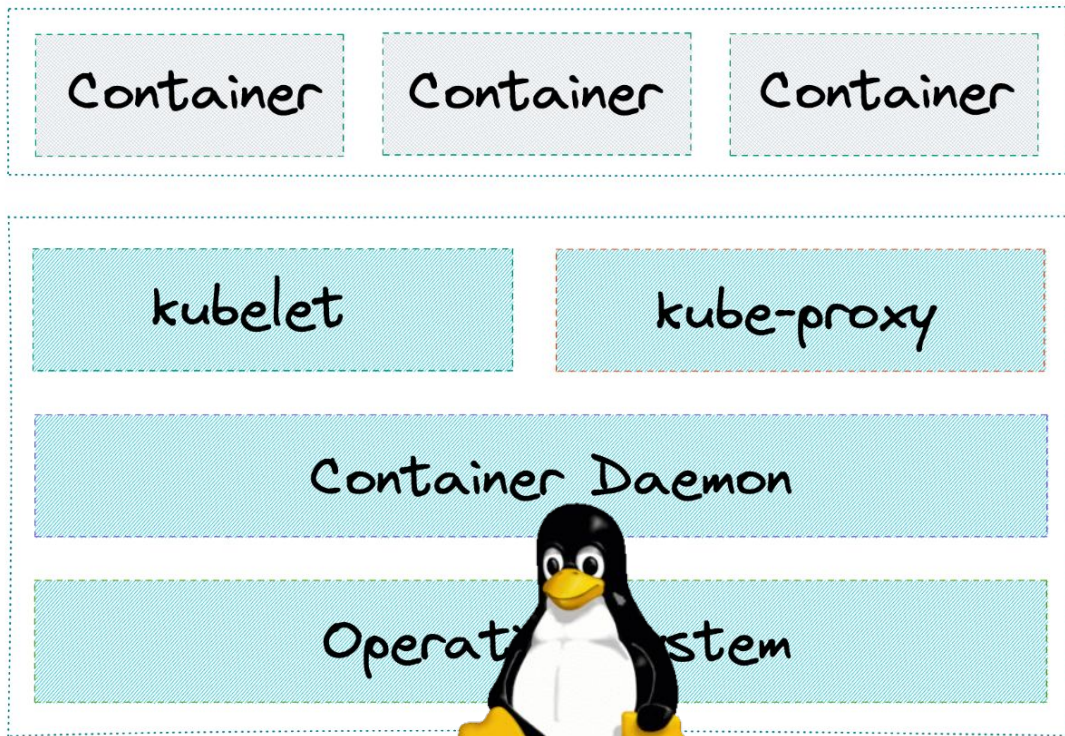


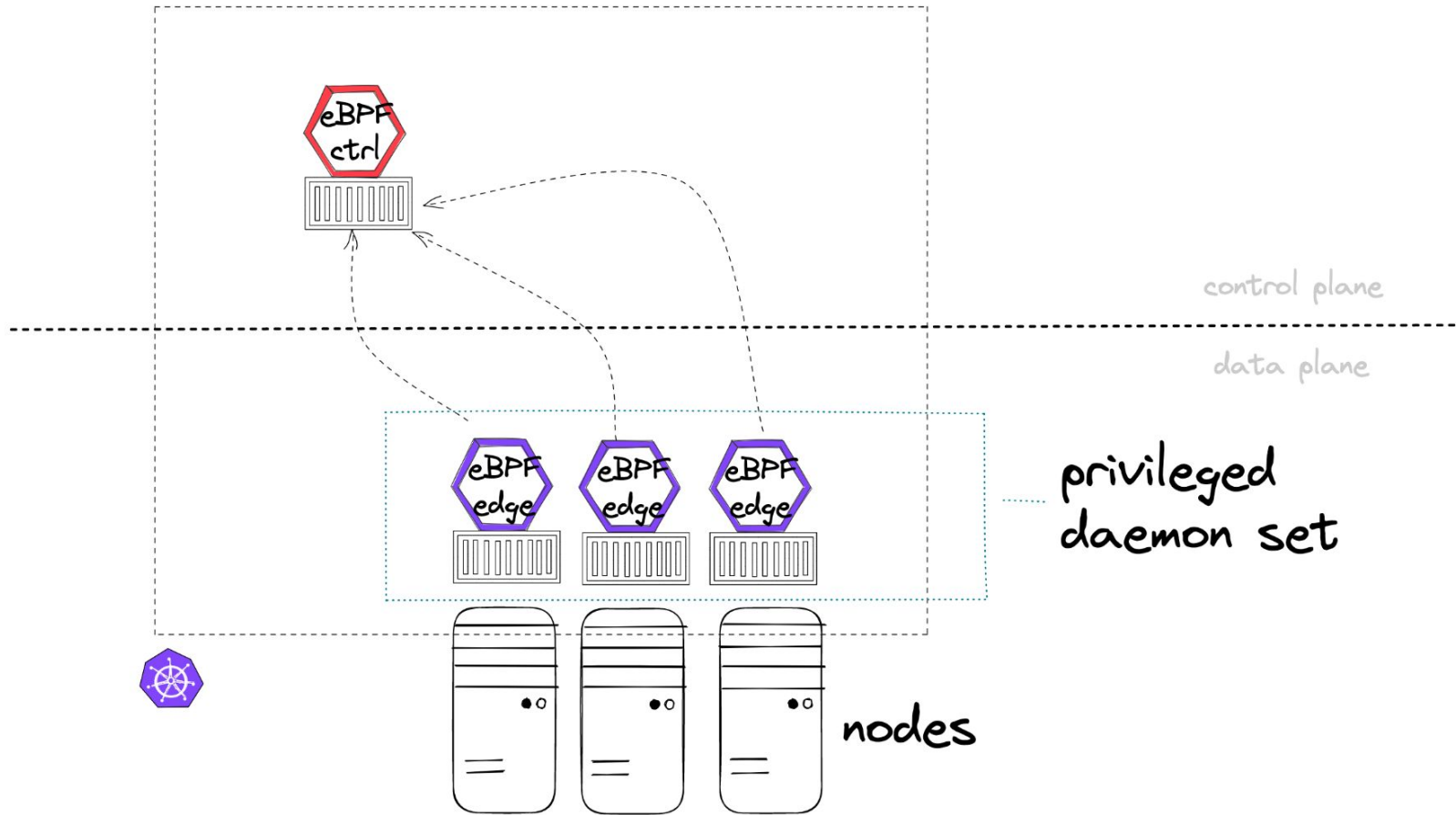
Kubernetes Worker Node

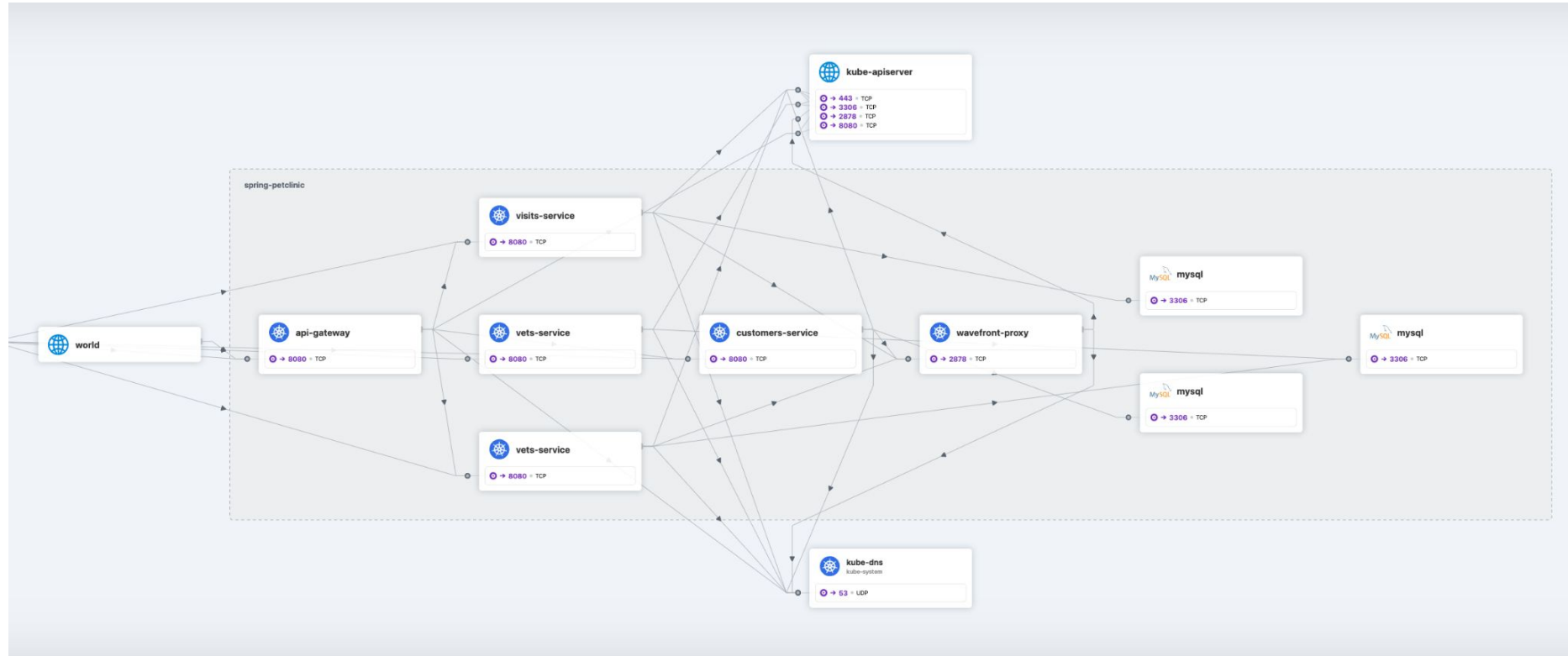


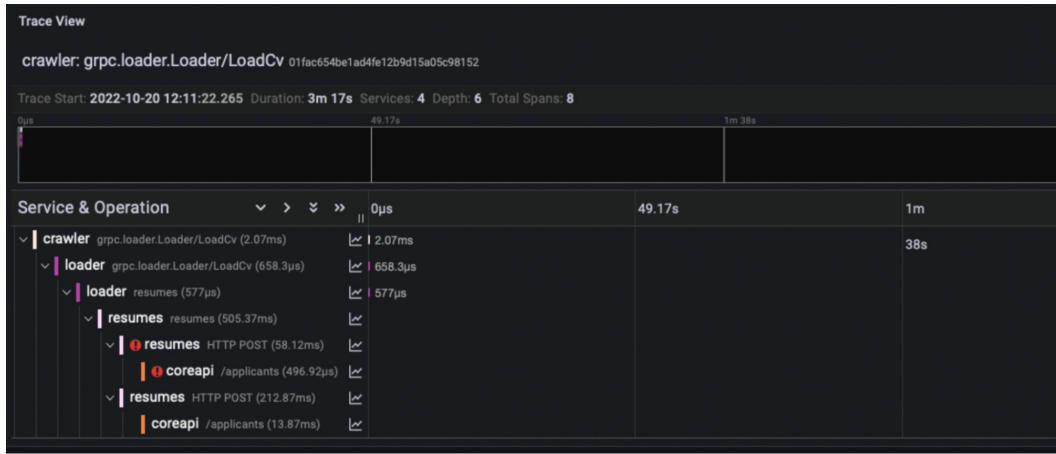
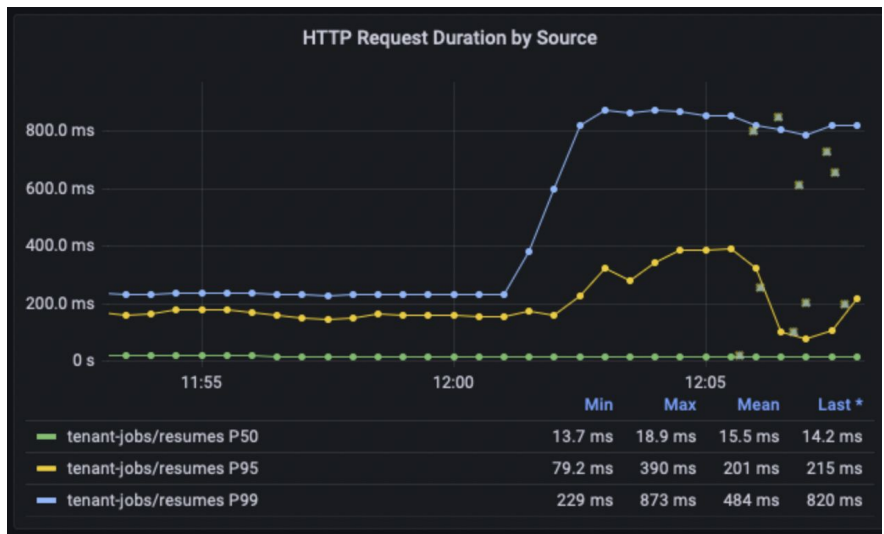


Kubernetes Worker Node





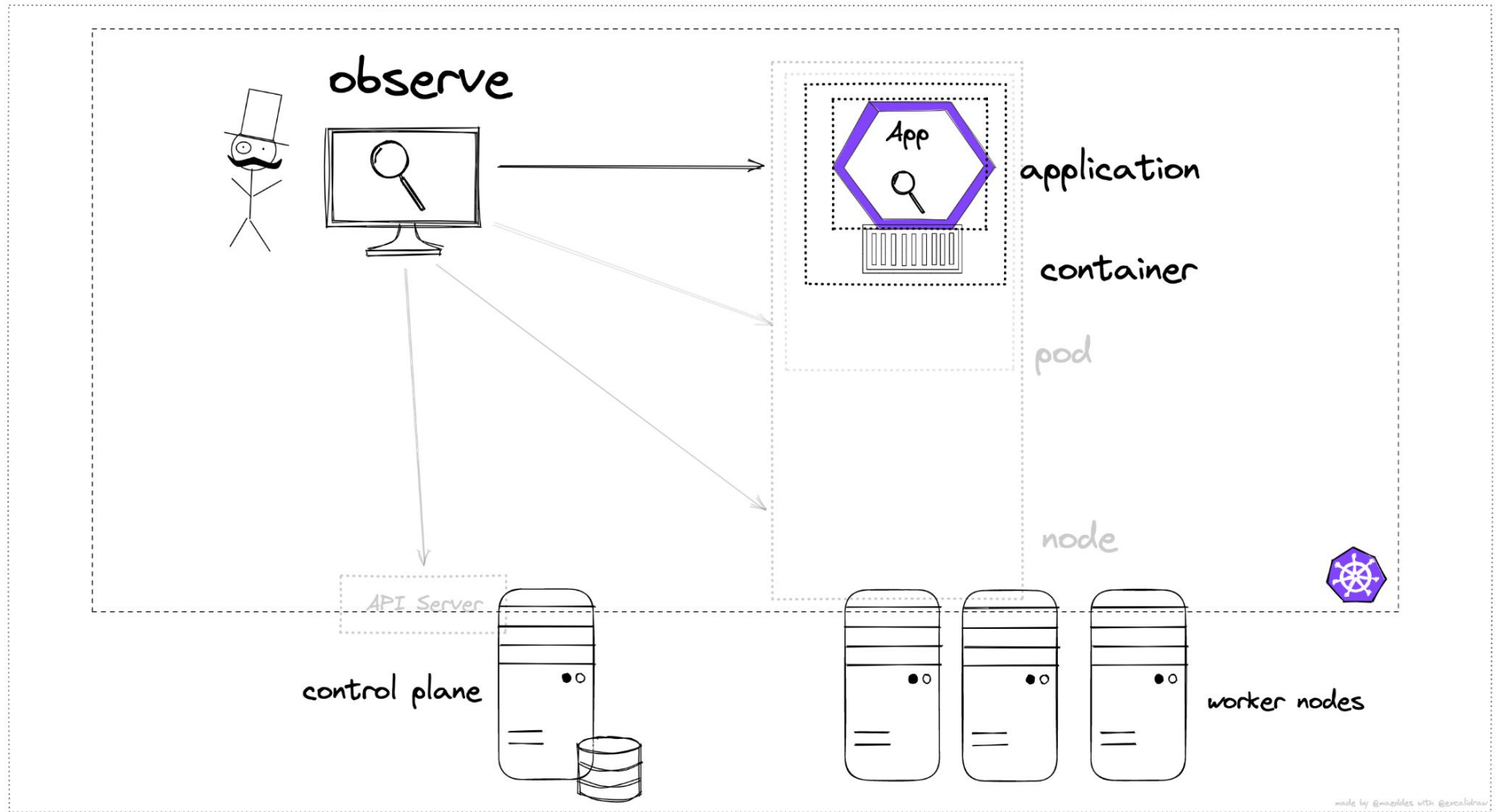




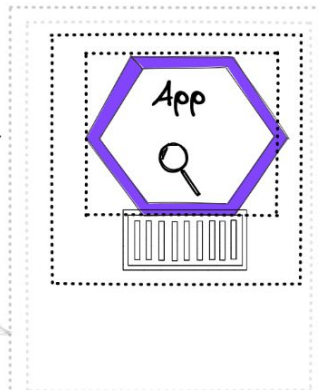
Characteristics:

- injection of proxy component on node instead of pod level
- Linux low-level functionality leveraged for Kubernetes
- early stage of development
- Service Mesh functionality in beta program level by Cilium
- lower latency with less hops as opposed to pod injection

application-based



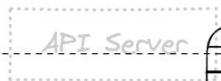
observe



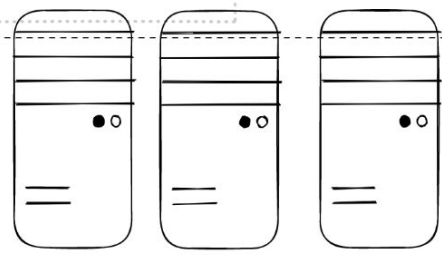
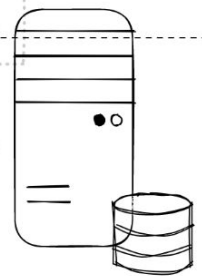
application
container

pod

node

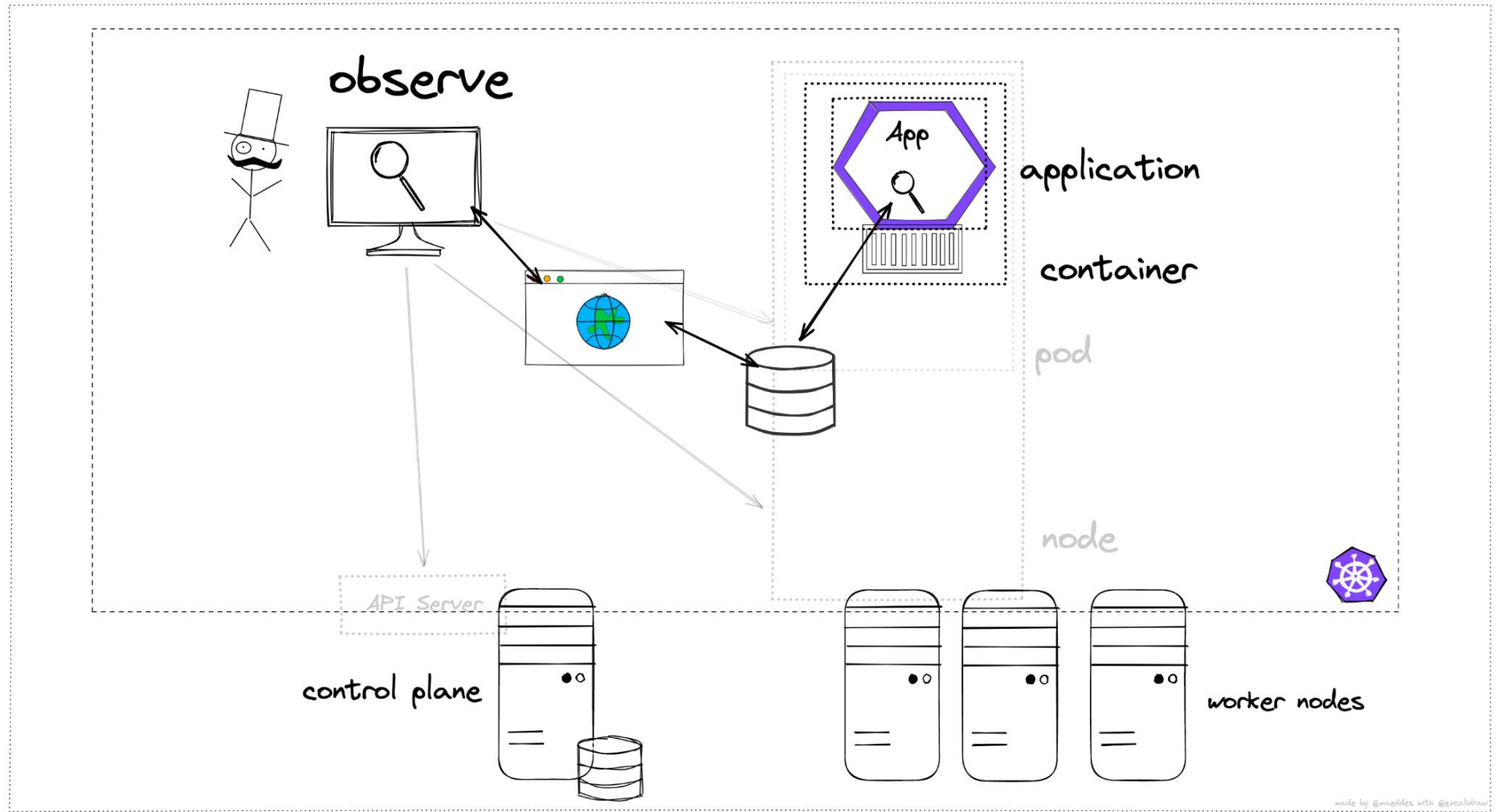


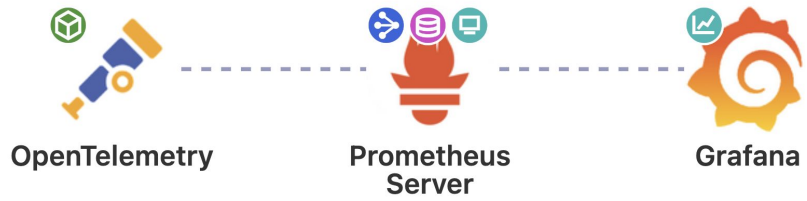
control plane

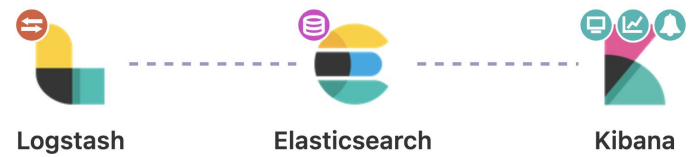
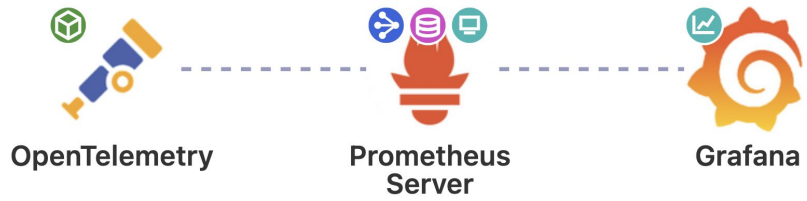


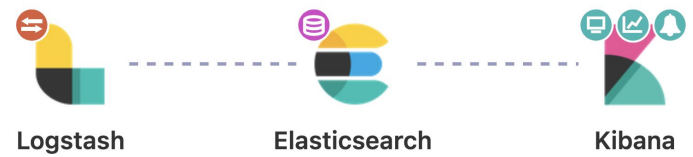
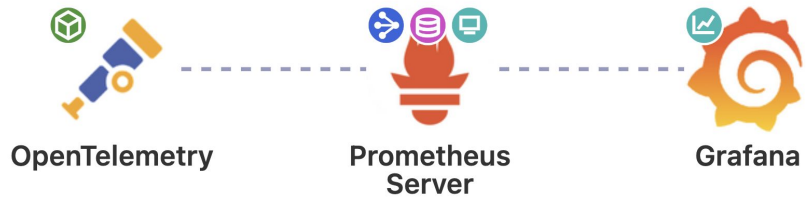
worker nodes

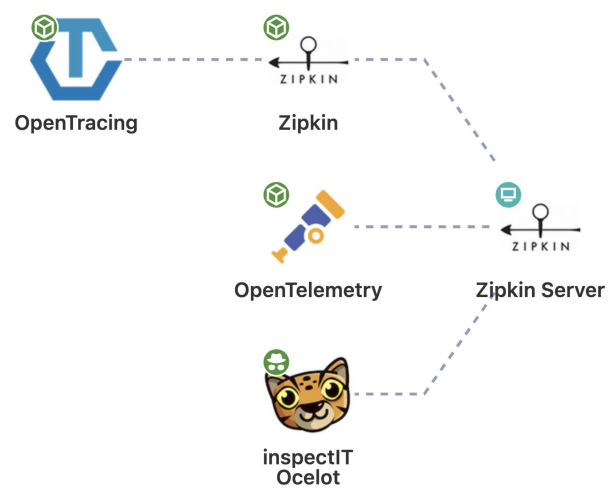
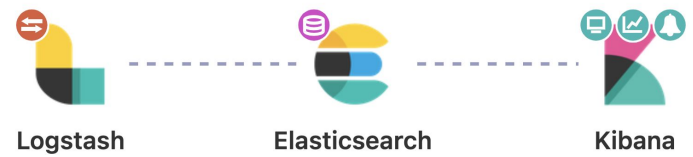
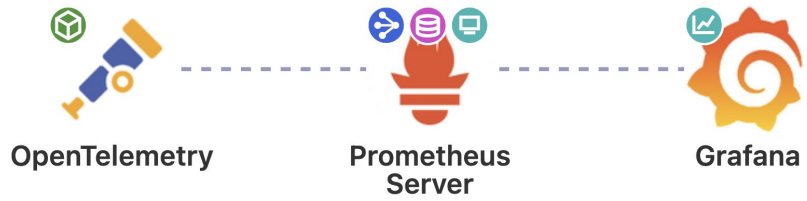














OpenTelemetry

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

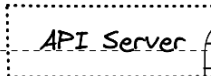
Managed DBs

APIs

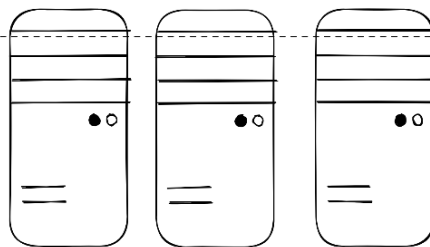
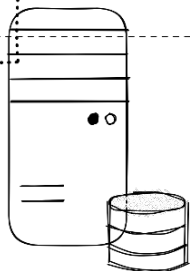
Client Instrumentation



observe



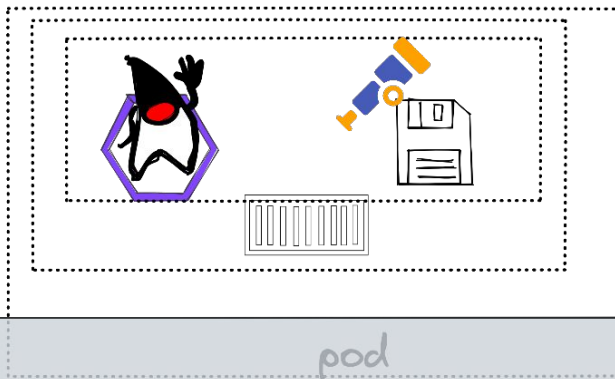
control plane



worker nodes

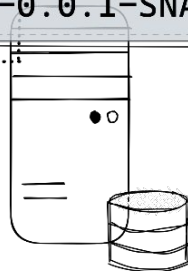


observe

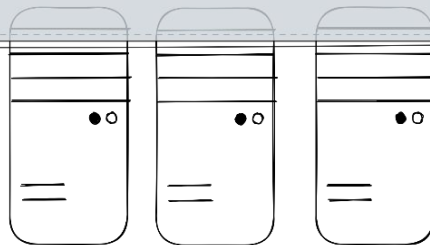


```
FROM eclipse-temurin:17
RUN mkdir -p /opt/todoui
WORKDIR /opt/todoui
ADD https://github.com/open-telemetry/opentelemetry-java-instrumentation/releases/latest/download/opentelemetry-javaagent.jar .
COPY target/todoui-0.0.1-SNAPSHOT.jar /opt/todoui
CMD ["java", "-javaagent:/opt/todoui/opentelemetry-javaagent.jar",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

control plane

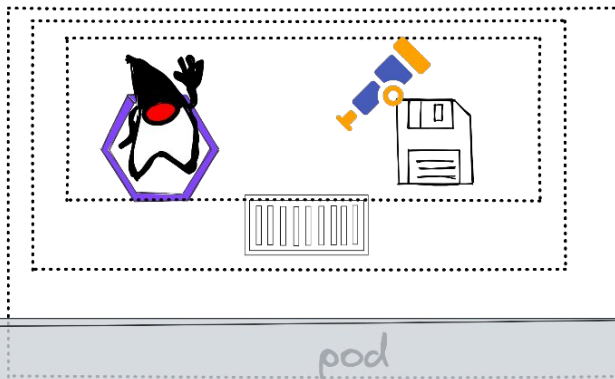


API Server



worker nodes

observe

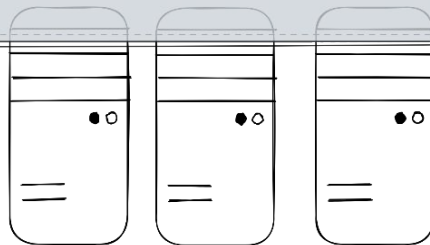
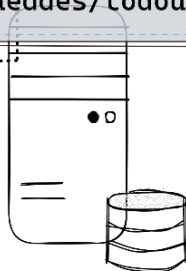


containers:

- 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

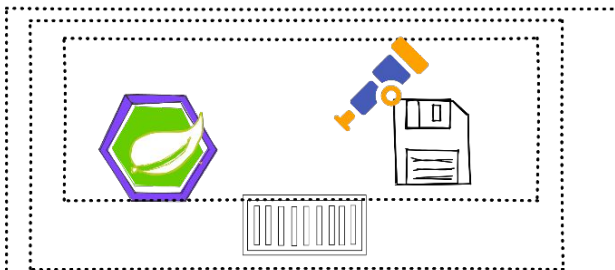
API Server

control plane



worker nodes

observe

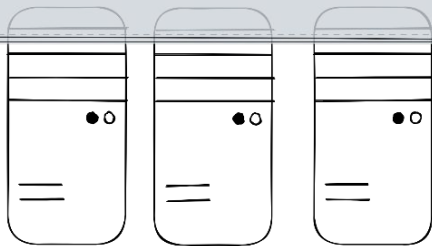
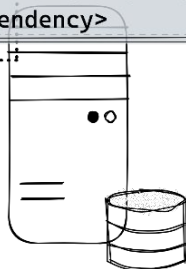


```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-sleuth</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-sleuth-otel-autoconfigure</artifactId>
</dependency>
<dependency>
  <groupId>io.opentelemetry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

pod

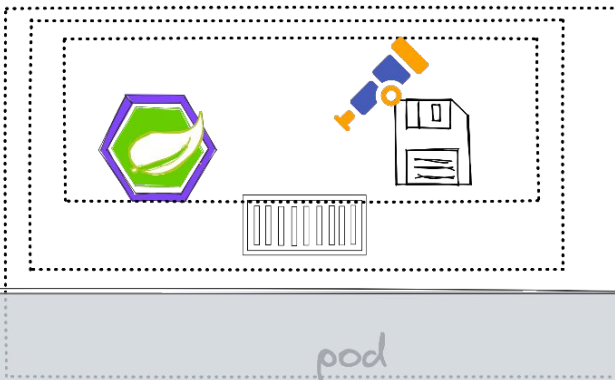
API S...

control plane



worker nodes

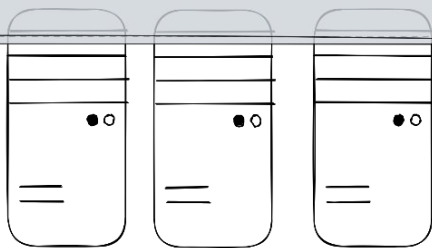
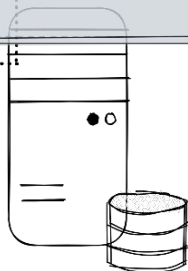
observe



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

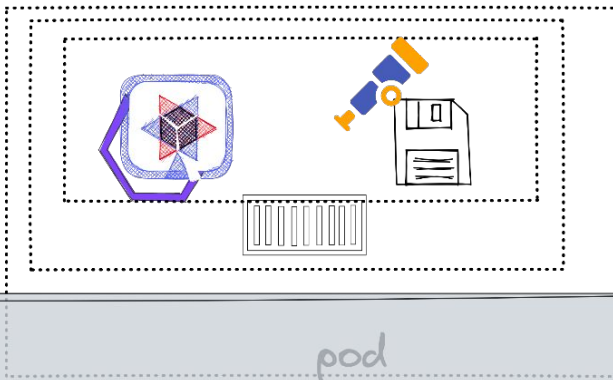
API Server

control plane



worker nodes

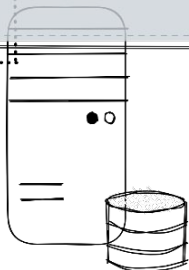
observe



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

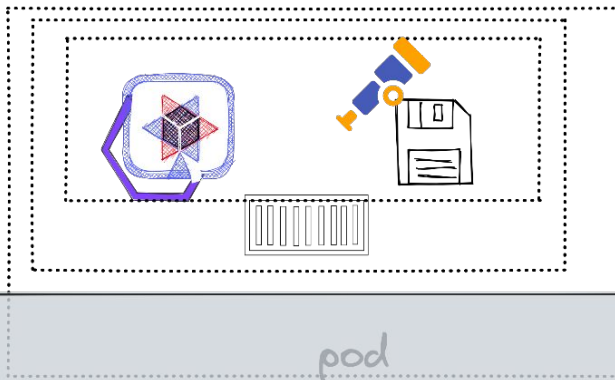
API Server

control plane



worker nodes

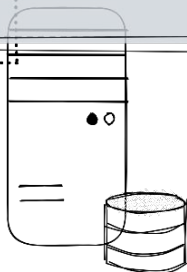
observe



```
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

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

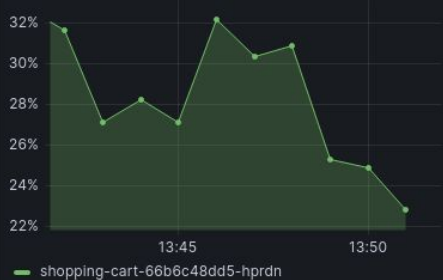
Client Instrumentation



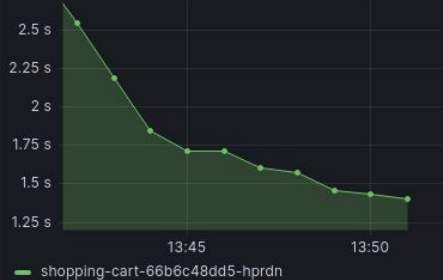
Rate



Error %



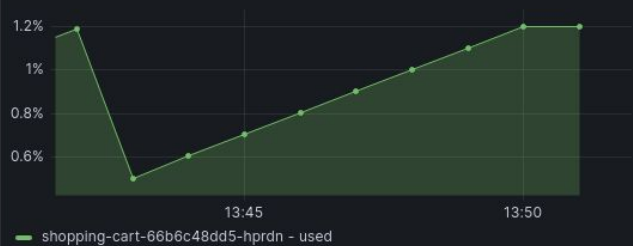
Duration



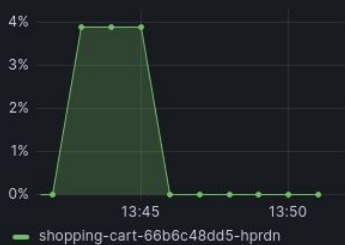
CPU utilization



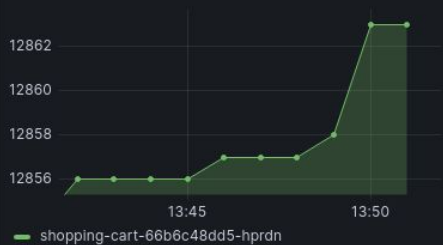
Heap Memory utilization



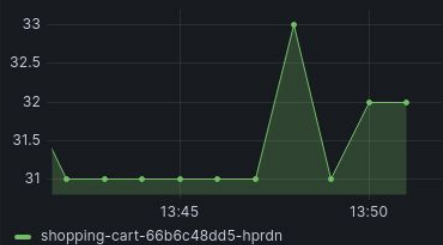
Garbage Collection



Classes

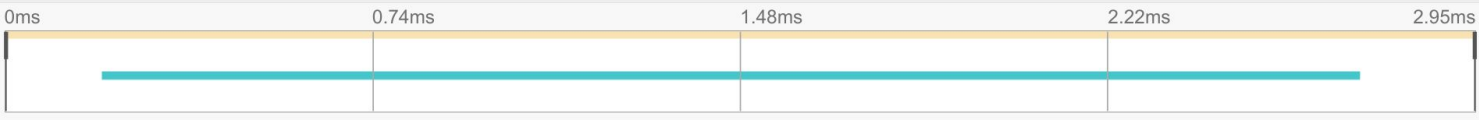


Threads



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

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	Duration
todoui.default todobackend.default.svc.cluster.local:8080/*	2.95ms
todobackend.default todobackend.default.svc.cluster.local:8080/*	2.53ms

todobackend.default.svc.cluster.local:8080/* Service: **todoui.default** | Duration: **2.95ms** | Start Time: **0ms**

> **Tags:** component = proxy | downstream_cluster = - | guid:x-request-id = 9f049e5f-5c33-9294-8d08-a95453762cda | h...

> **Process:** ip = 10.244.0.34

SpanID: e95a8a63a3f0ff03 🔗

todobackend.default.svc.cluster.local:8080/* Service: **todobackend.default** | Duration: **2.53ms** | Start Time: **0.2ms**

> **Tags:** component = proxy | downstream_cluster = - | guid:x-request-id = 9f049e5f-5c33-9294-8d08-a95453762cda | h...

> **Process:** ip = 10.244.0.30

SpanID: d0ae89ed674ea9f6 🔗

Jaeger UI | Lookup by Trace ID... | Search | Compare | System Architecture | About Jaeger

← | todoui.default: GET / 3e6d859 | Find... | Trace Timeline

Trace Start **January 11, 2021, 16:43:22.986** | Duration **5.34ms** | Services **2** | Depth **4** | Total Spans **4**

Service & Operation

- todoui.default GET /
 - todoui.default GET /todos/
 - todobackend.default GET /todos/
 - HikariProxyPreparedStatement.executeQuery** HikariProxyPrep... (129µs)

HikariProxyPreparedStatement.executeQuery | Service: **todobackend.default** | Duration: **129µs** | Start Time: **2.35ms**

Tags

db.type	sql
db.url	postgresql://postgresdb:5432/mydb
internal.span.format	jaeger
java.fqn	com.zaxxer.hikari.pool.HikariProxyPreparedStatement.executeQuery()

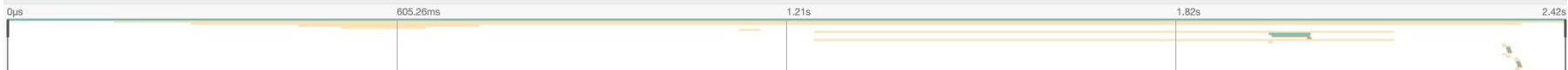
Process:

SpanID: [bd3a5eae7e70936](#)

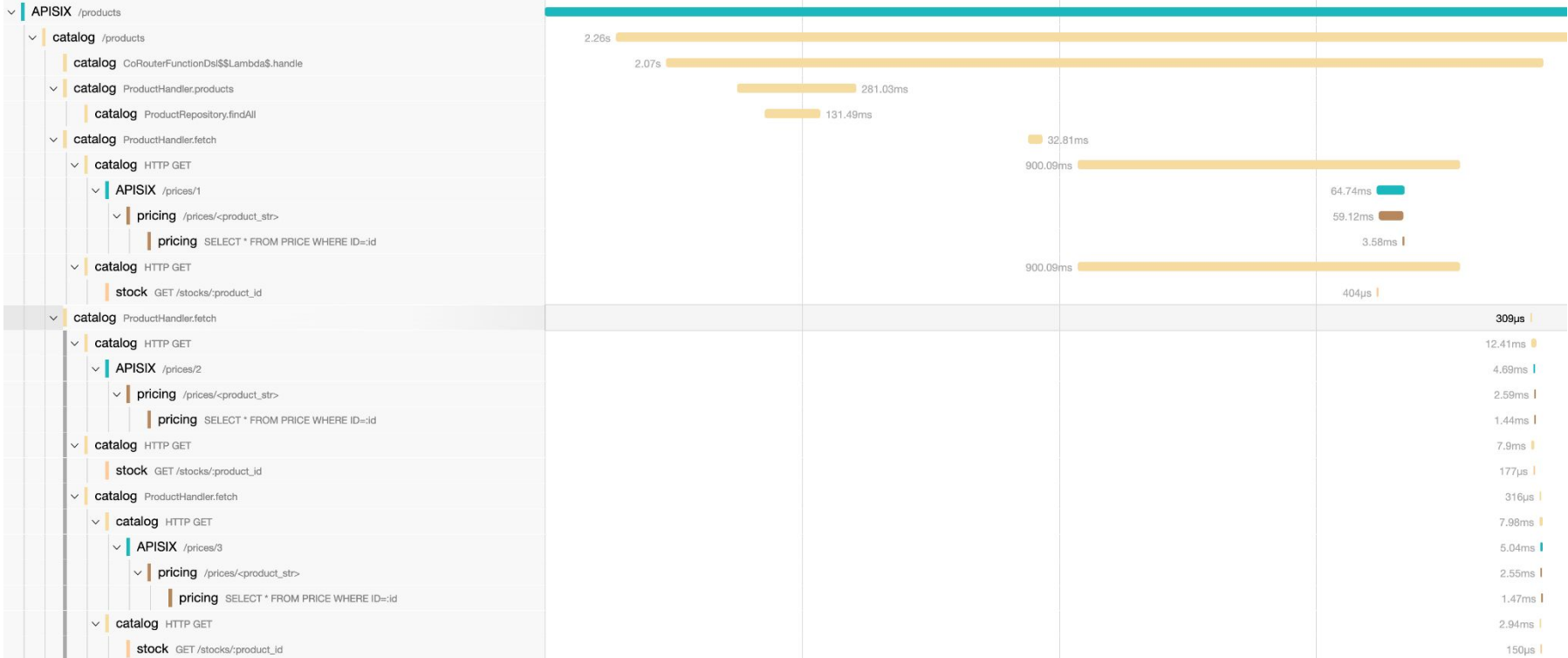

```
@WithSpan("ProductHandler.fetch")
private suspend fun fetchProductDetails(@SpanAttribute("id") id: Long, product: Product) = coroutineS
    val price = async(Dispatchers.IO) {
        | client.get().uri("${props.pricingEndpoint}/${product.id}").retrieve().bodyToMono<Price>().awa
    }
    val stocks = async(Dispatchers.IO) {
        | client.get().uri("${props.stockEndpoint}/${product.id}").retrieve().bodyToMono<Array<InStockL
        |     .awaitSingle()
    }
    product.withDetails(price.await(), stocks.await())
}
```

```
@WithSpan("ProductHandler.fetch")
private suspend fun fetchProductDetails(@SpanAttribute("id") id: Long, product: Product) = coroutineScope {
    val price = async(Dispatchers.IO) {
        client.get().uri("${props.pricingEndpoint}/${product.id}").retrieve().bodyToMono<Price>().await()
    }
    val stocks = async(Dispatchers.IO) {
        client.get().uri("${props.stockEndpoint}/${product.id}").retrieve().bodyToMono<Array<InStockList>>()
            .awaitSingle()
    }
    product.withDetails(price.await(), stocks.await())
}
```

Trace Start **May 10 2023, 12:17:29.196** Duration **2.42s** Services **4** Depth **8** Total Spans **26**



Service & Operation



stock GET /stocks/:product_id

catalog ProductHandler.fetch

catalog HTTP GET

APISIX /prices/2

ProductHandler.fetch

> **Tags:** code.function = fetchProductDetails | code.namespace = ch.frank | catalog.ProductHandler | id = 2 | internal.span.form

> **Process:** container.id = 9d97133752655fbd9b3824cb4e2a792016d3b71c1145 | 9e8ed2c3cc03ed18d | host.arch = amd64 | host

demo kudos!



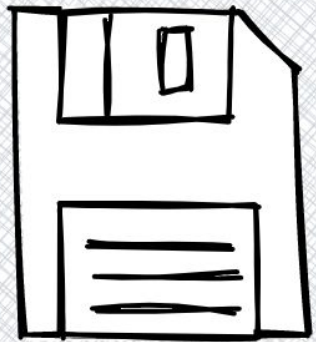
https://twitter.com/nicolas_frinkel



<https://www.baeldung.com/>

<https://github.com/nfrankel/opentelemetry-tracing>

<https://www.baeldung.com/spring-boot-opentelemetry-setup>



tools?

This image displays a vast collection of logos for cloud-native technologies, organized into several main categories:

- App Definition and Development:** Database, Streaming & Messaging, Application Definition & Image Build, Continuous Integration & Delivery.
- Orchestration & Management:** Scheduling & Orchestration, Coordination & Service Discovery, Remote Procedure Call, Service Proxy, API Gateway, Service Mesh.
- Runtime:** Cloud Native Storage, Container Runtime, Cloud Native Network.
- Provisioning:** Automation & Configuration, Container Registry, Security & Compliance, Key Management.
- Platform:** Certified Kubernetes - Distribution, Certified Kubernetes - Hosted, Certified Kubernetes - Installer, Paas/Container Service.
- Observability and Analysis:** Monitoring.
- Serverless**
- Members**
- CD Foundation Landscape**

A red circle highlights the **Platform** section, specifically the **Paas/Container Service** area, which includes logos for various managed Kubernetes and container services.

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 Agent



Prometheus Exporter



Promitor



Promtail



Scouter Agent



Sensu Client



SkyWalking



Telegraf



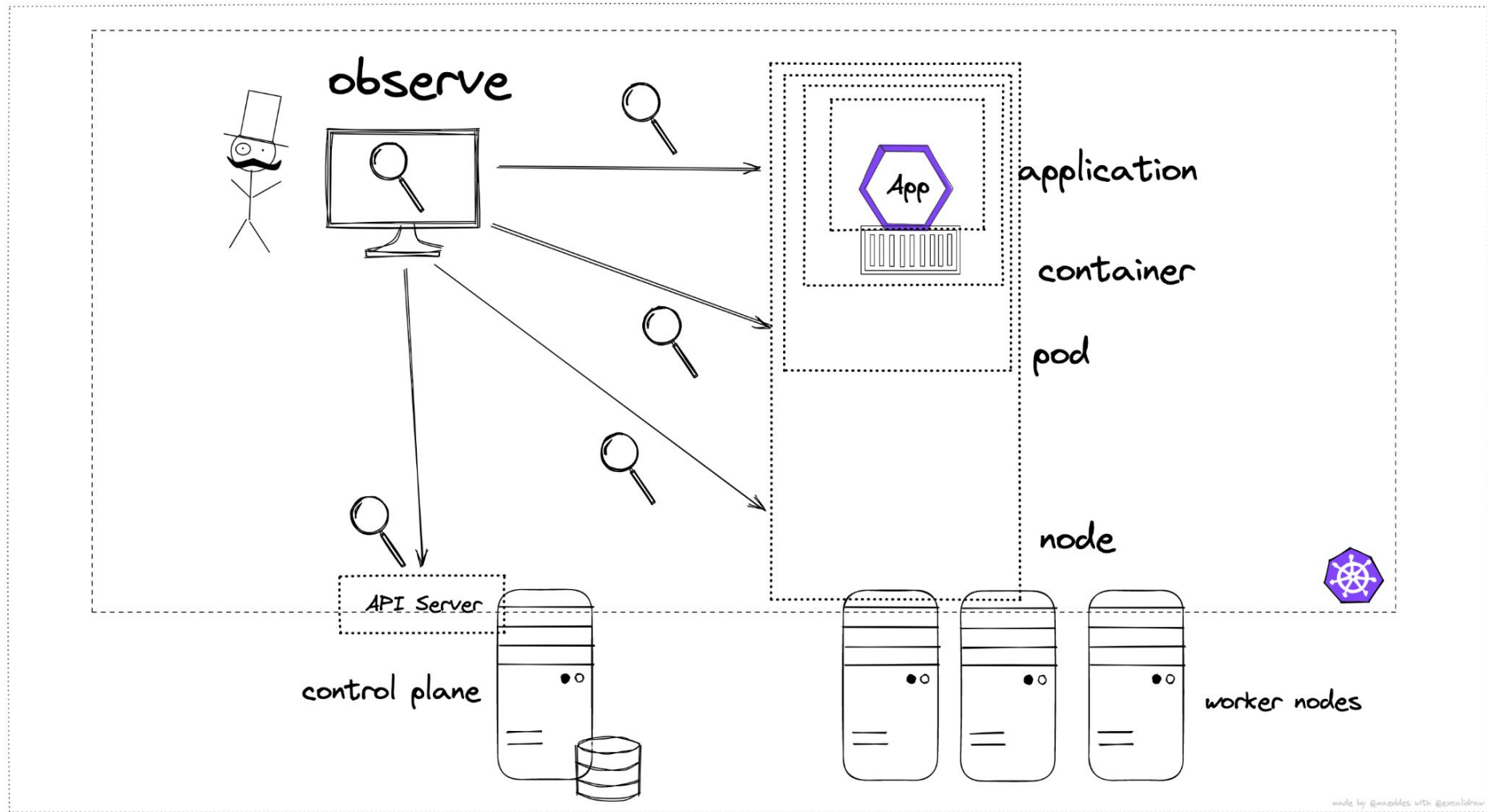
Wavefront J-Agent



collectd

Characteristics:

- agents specific to programming languages and frameworks
- provide application level metrics
- often part of the application process or base container
- enable root-cause analysis





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

Thanks!



matthiashaessler



@maeddes