



# Scale your deployment using Custom Metrics

**Rajat Khanna**  
**SDE III @ CommercelQ**



legalimpurity



rajatkhan08



legalimpurity





iStock™  
Credit: Gurzzza

1218409189





1218409164

iStock™  
Credit: Gurzzza





1218409135

iStock™  
Credit: Gurzzza









iStock™  
Credit: zubada



# Why Autoscaling?

# Why Autoscaling?

- Static configuration is not optimal.
- When demand increases the app should scale up to stay responsive.
- When demand decreases the app should scale down to not waste resource and save cost.

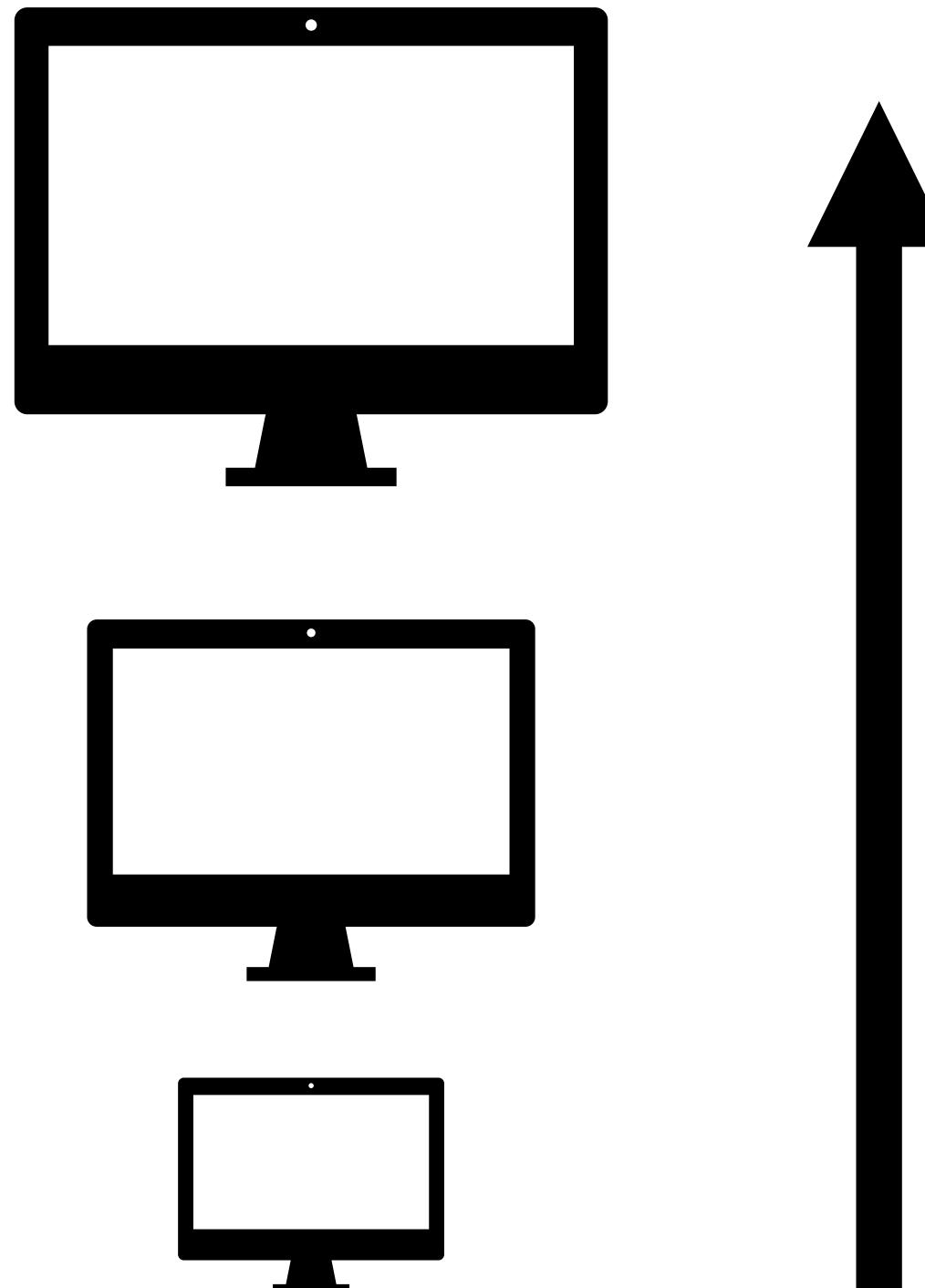


# Different types of Autoscaling

# Different types of Autoscaling

## Vertical Scaling

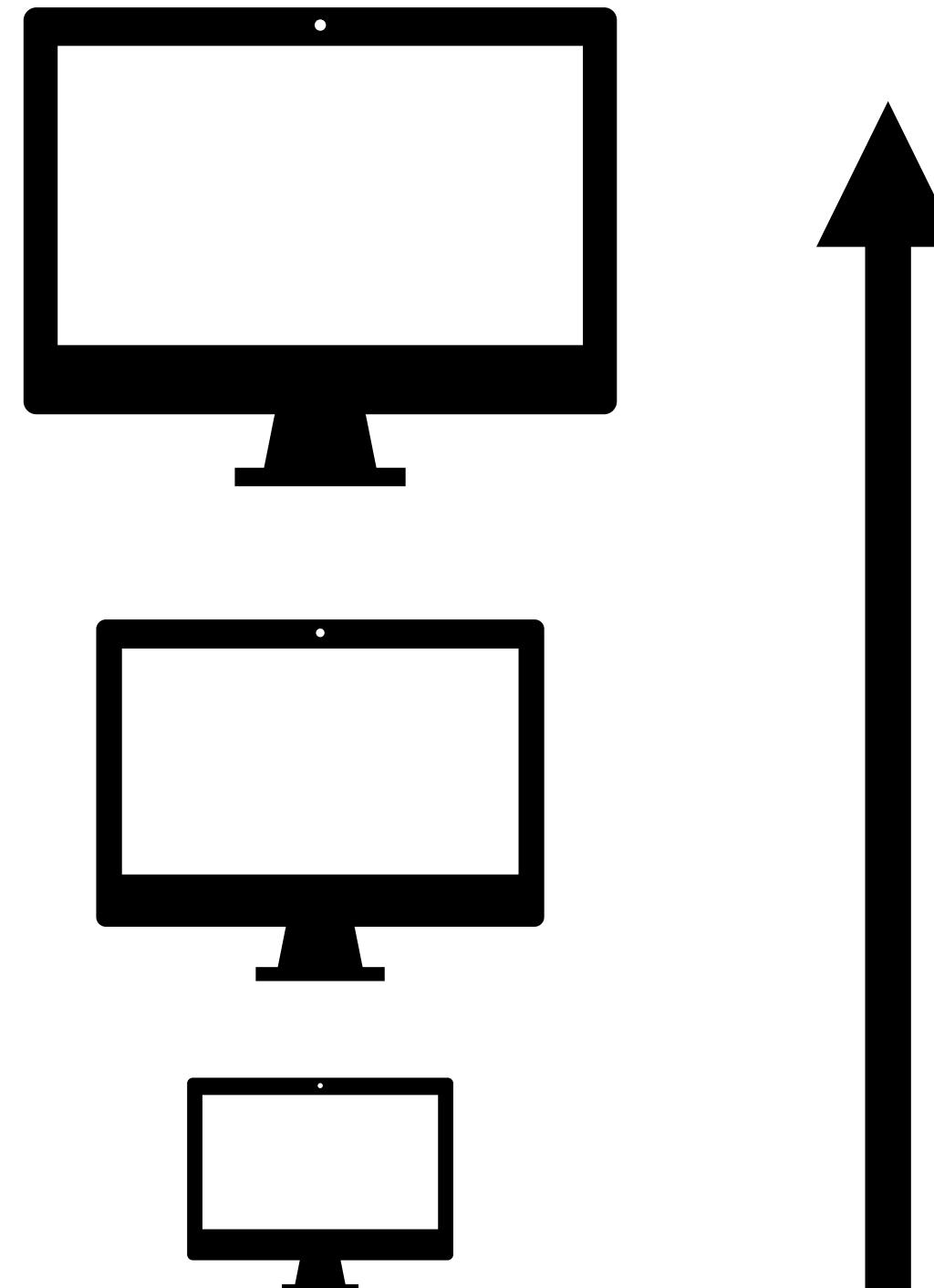
Increase size of  
the Instance (RAM, CPU etc.)



# Different types of Autoscaling

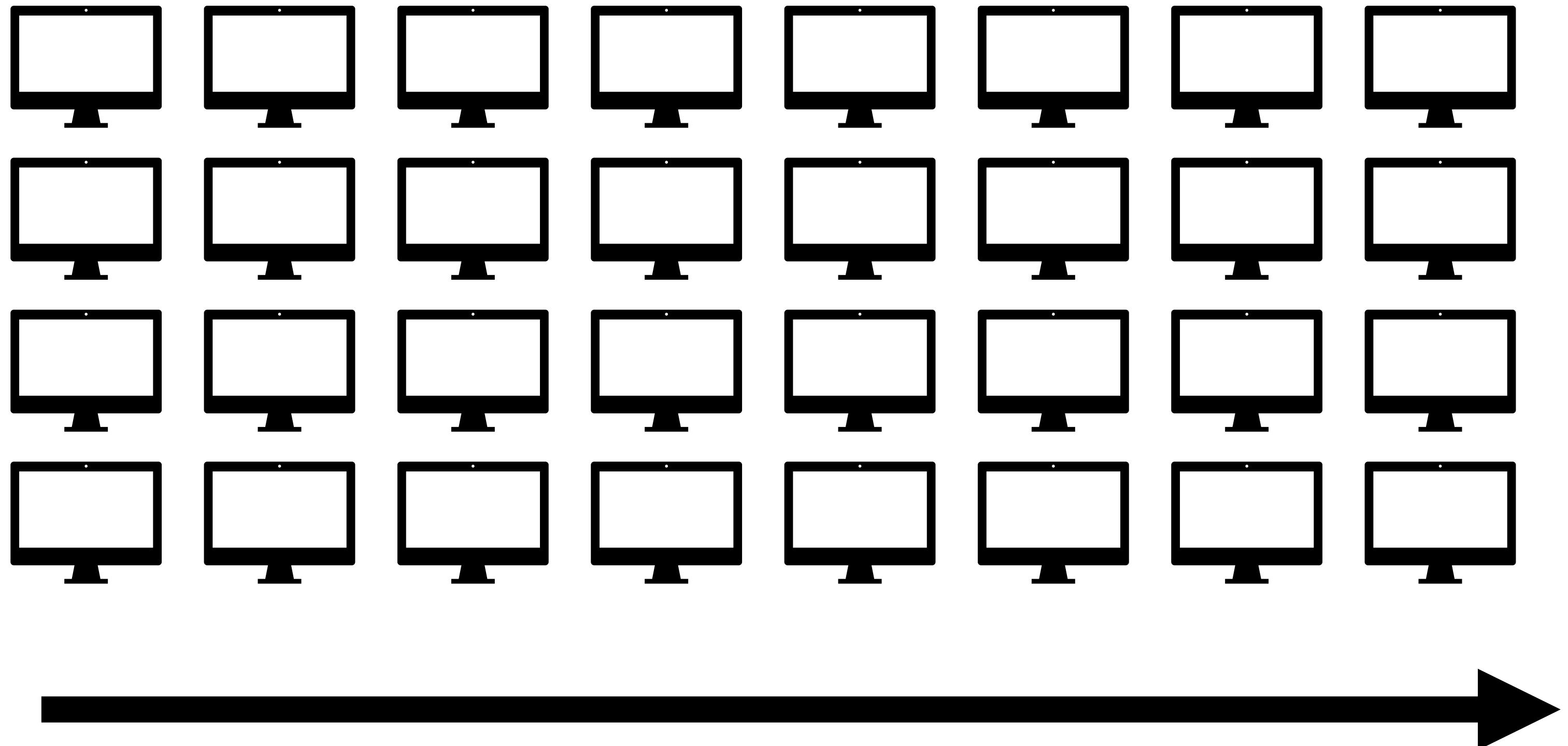
## Vertical Scaling

Increase size of  
the Instance (RAM, CPU etc.)



## Horizontal Scaling

Add more instances

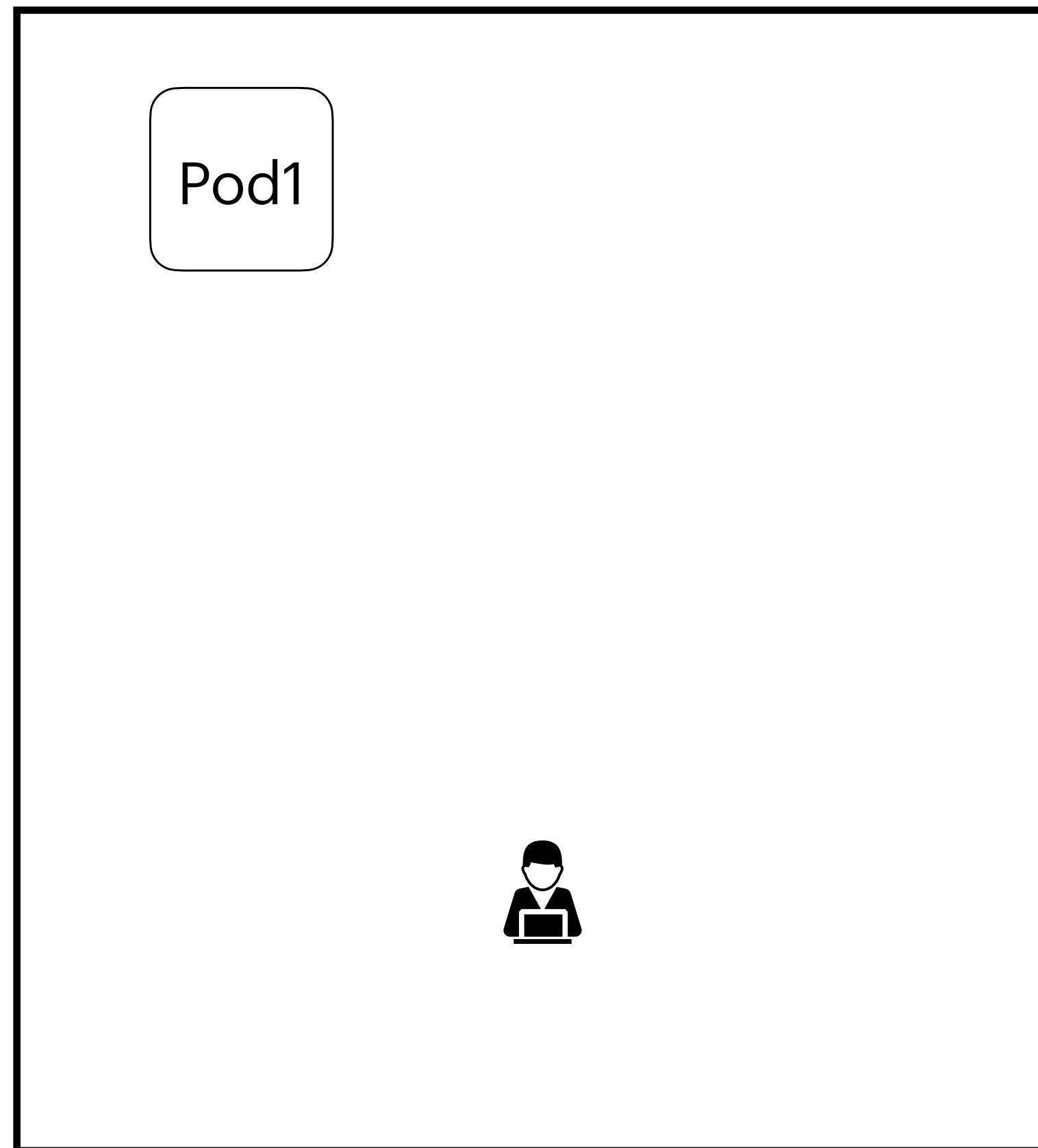




# Different types of Autoscalers in Kubernetes

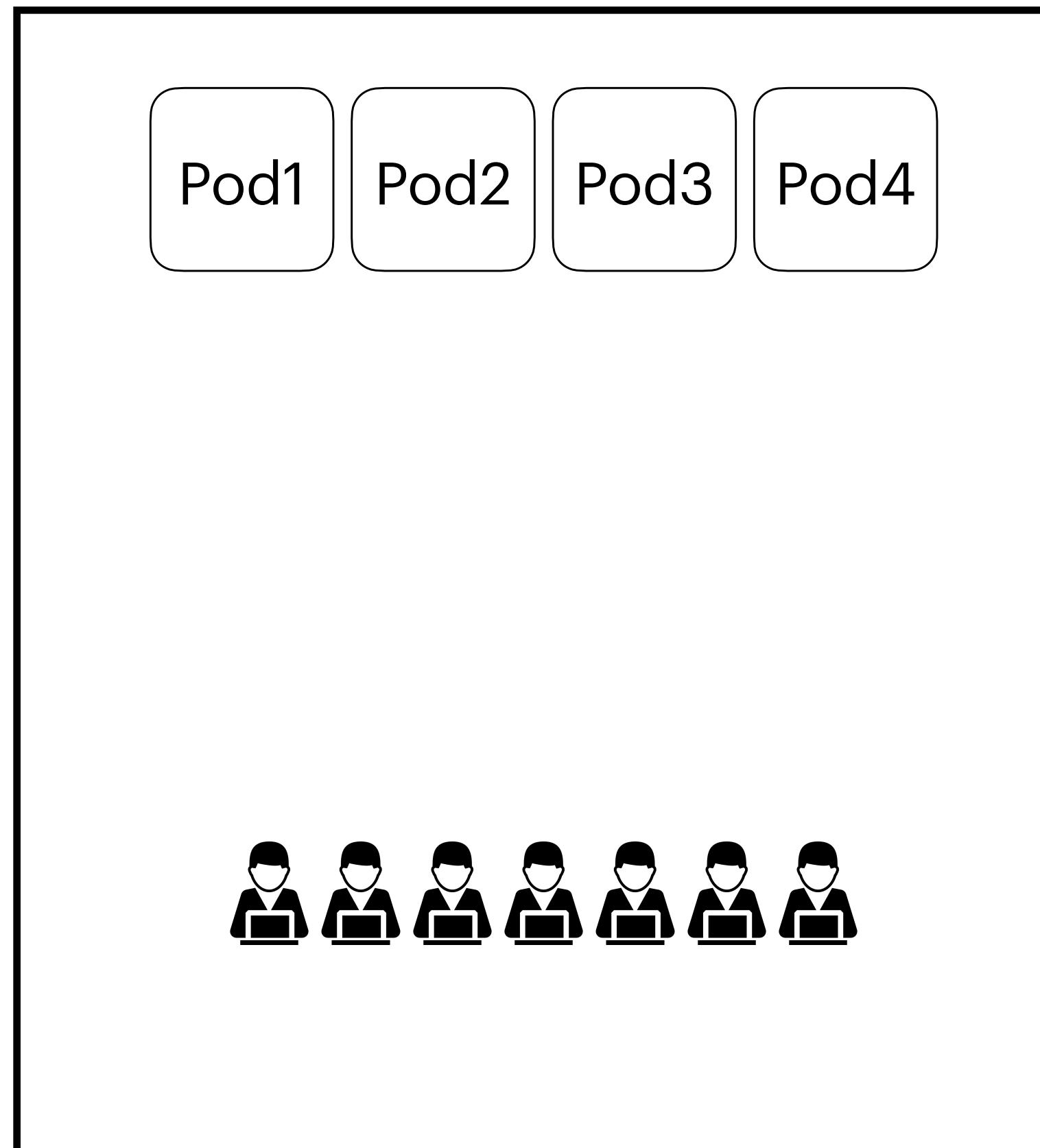
# Different types of Autoscalers in Kubernetes

## Horizontal Pod Autoscaler



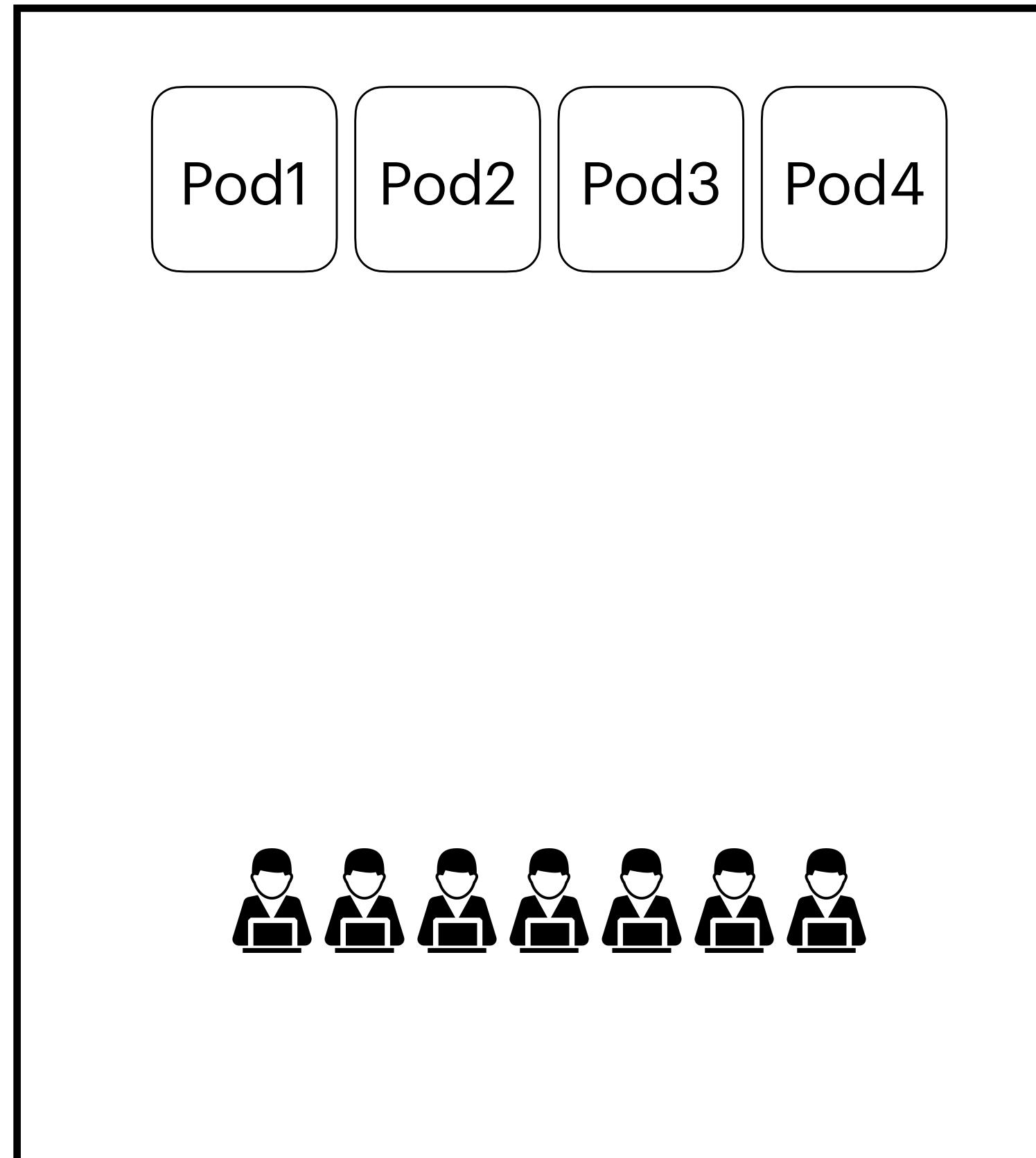
# Different types of Autoscalers in Kubernetes

## Horizontal Pod Autoscaler

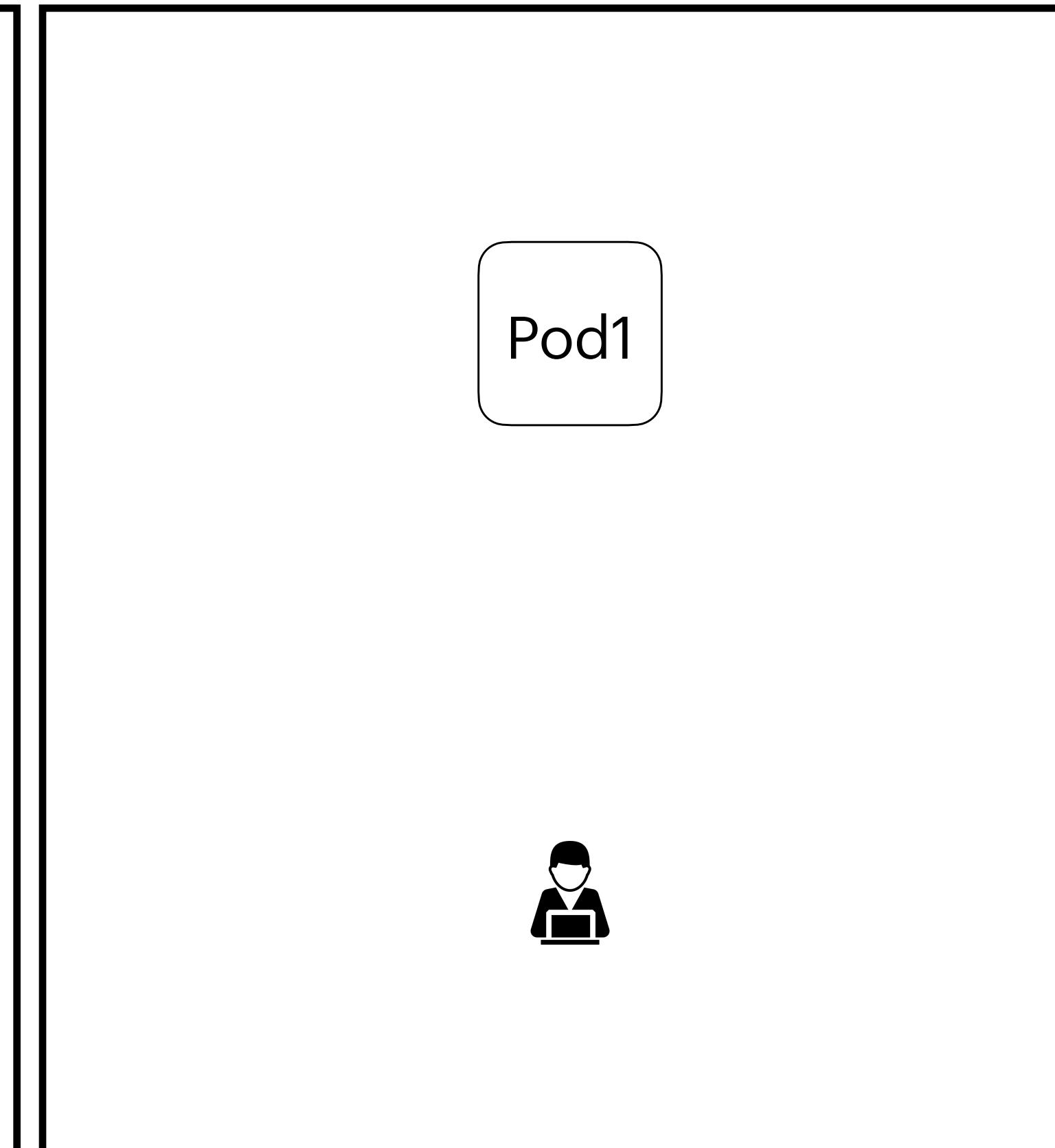


# Different types of Autoscalers in Kubernetes

Horizontal Pod Autoscaler

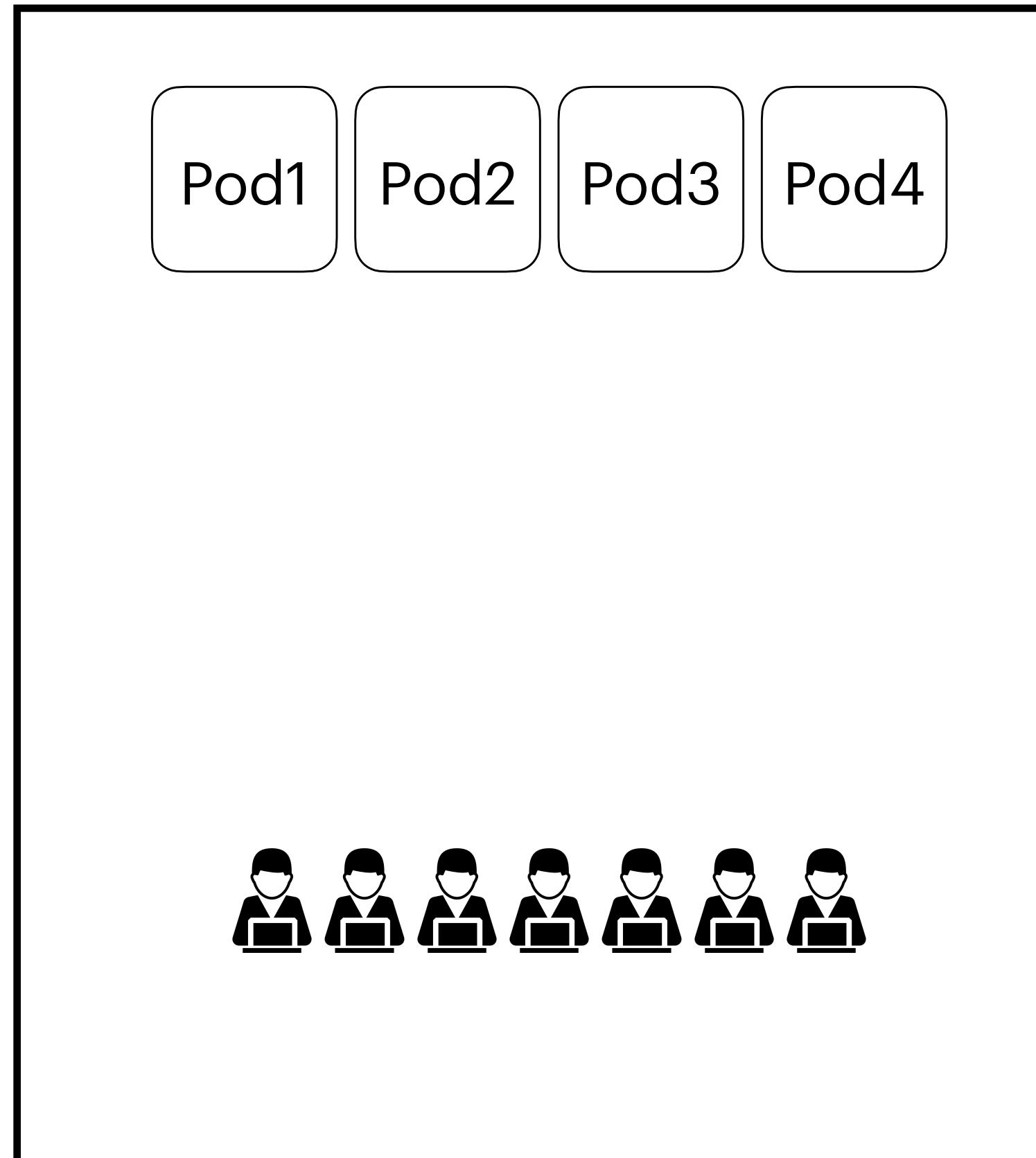


Vertical Pod Autoscaler

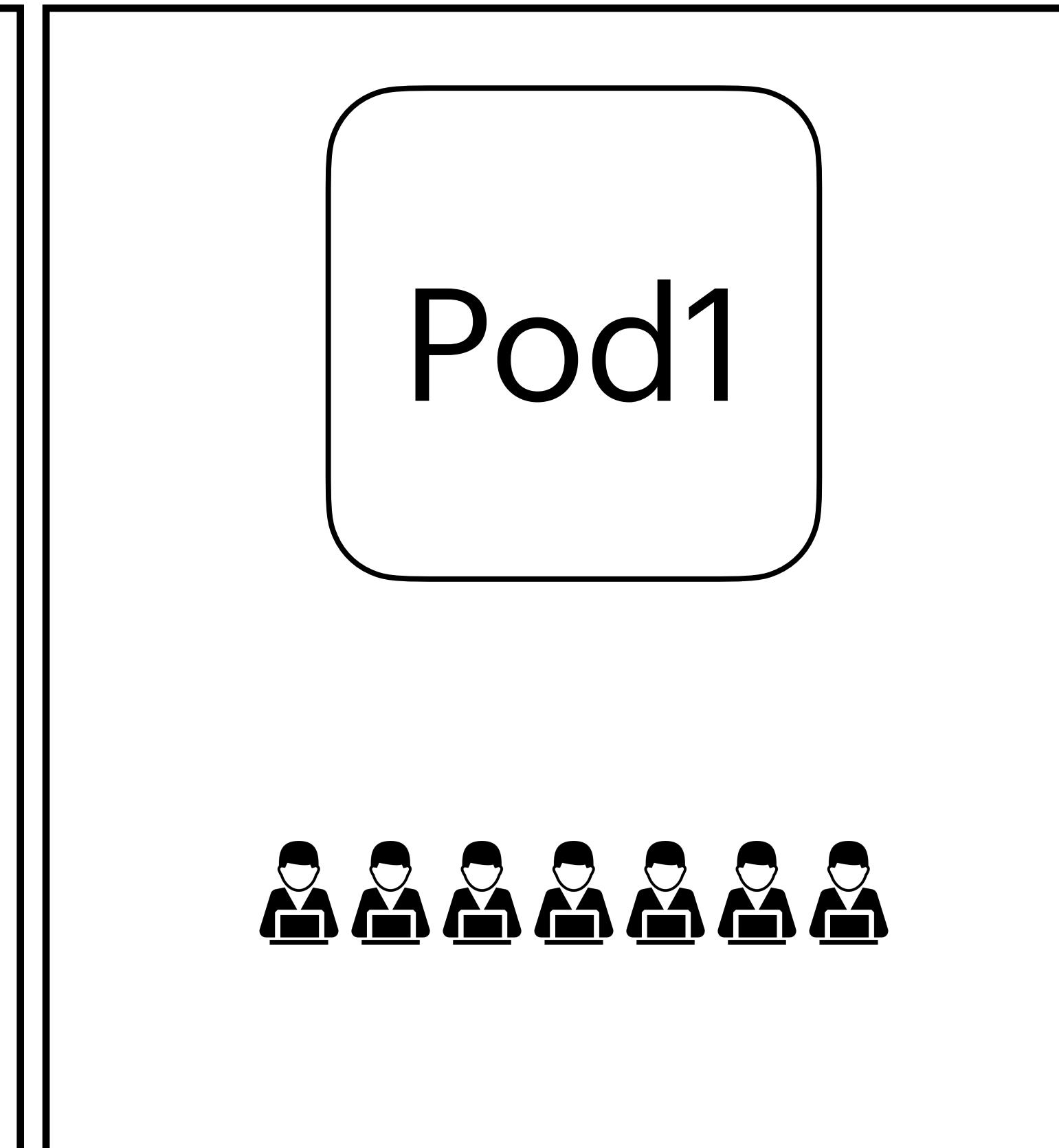


# Different types of Autoscalers in Kubernetes

Horizontal Pod Autoscaler

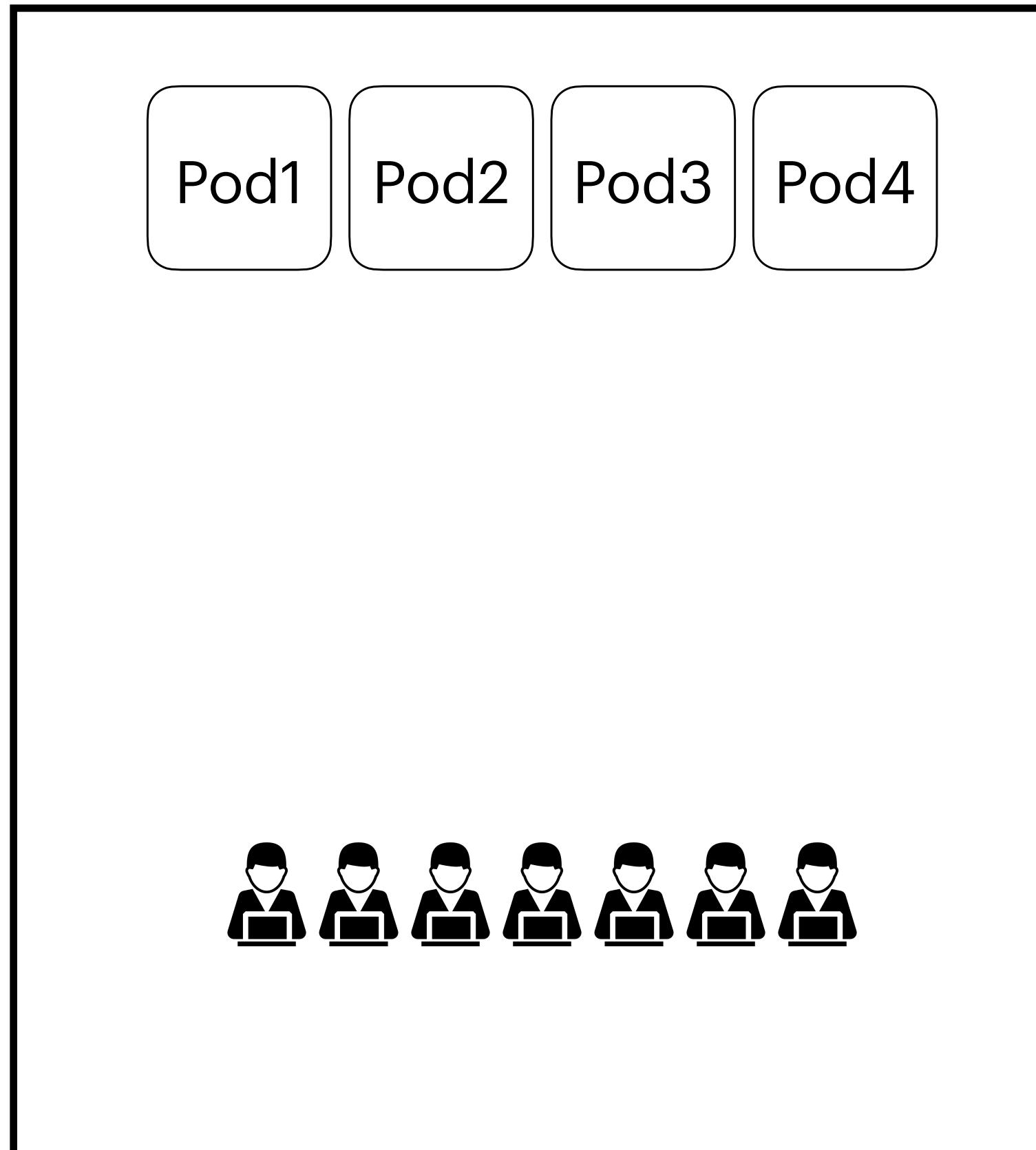


Vertical Pod Autoscaler

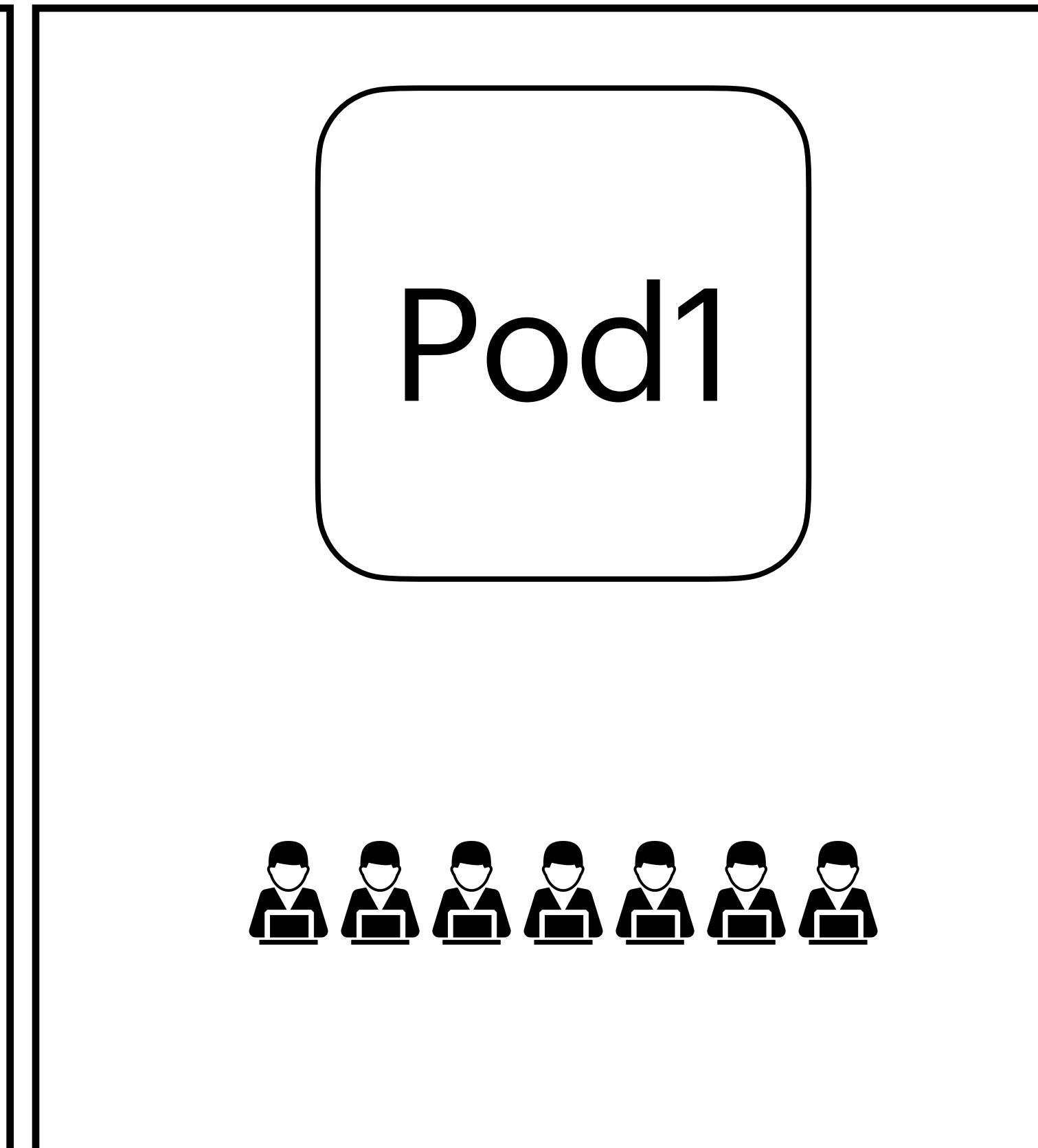


# Different types of Autoscalers in Kubernetes

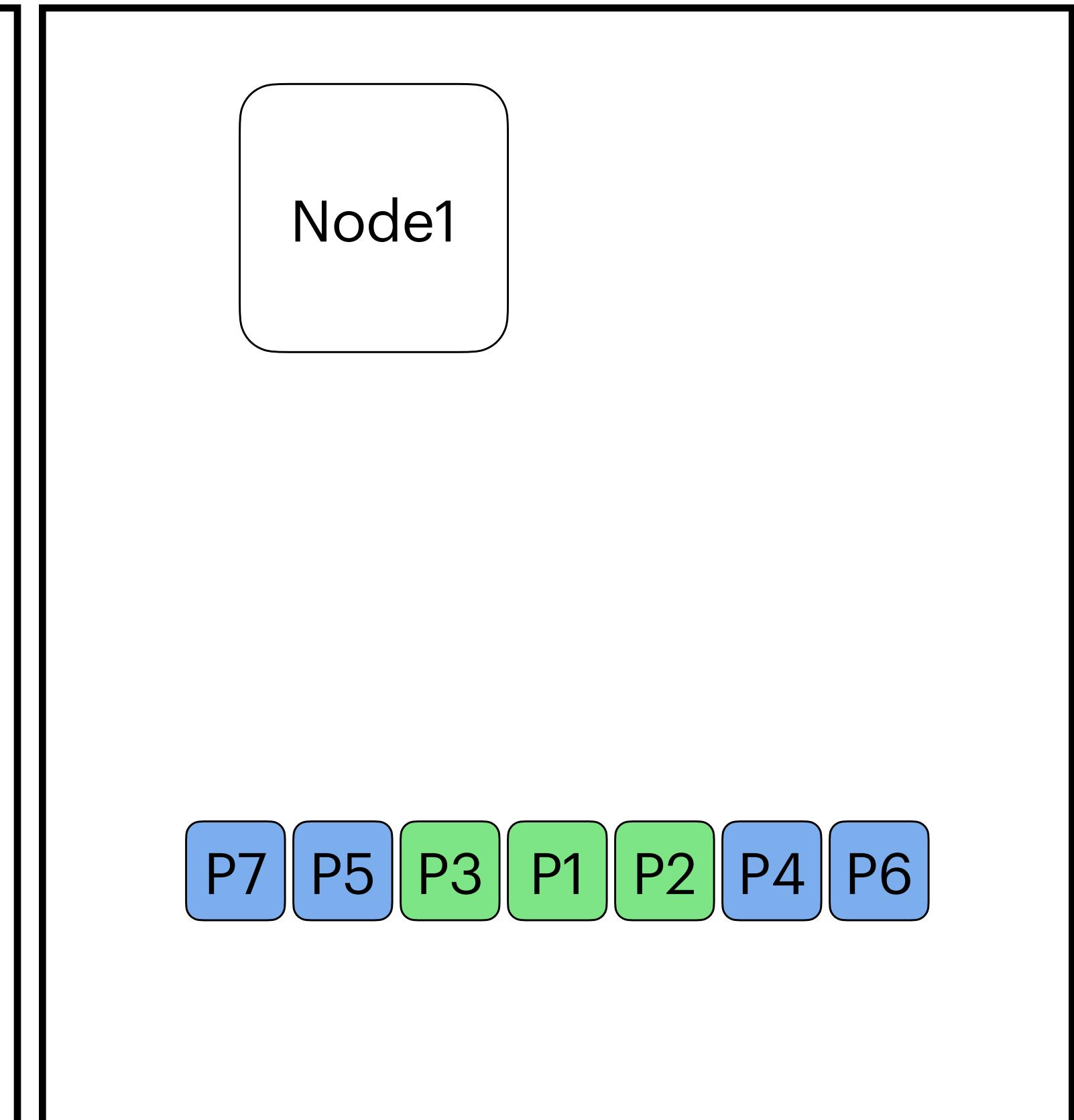
Horizontal Pod Autoscaler



Vertical Pod Autoscaler

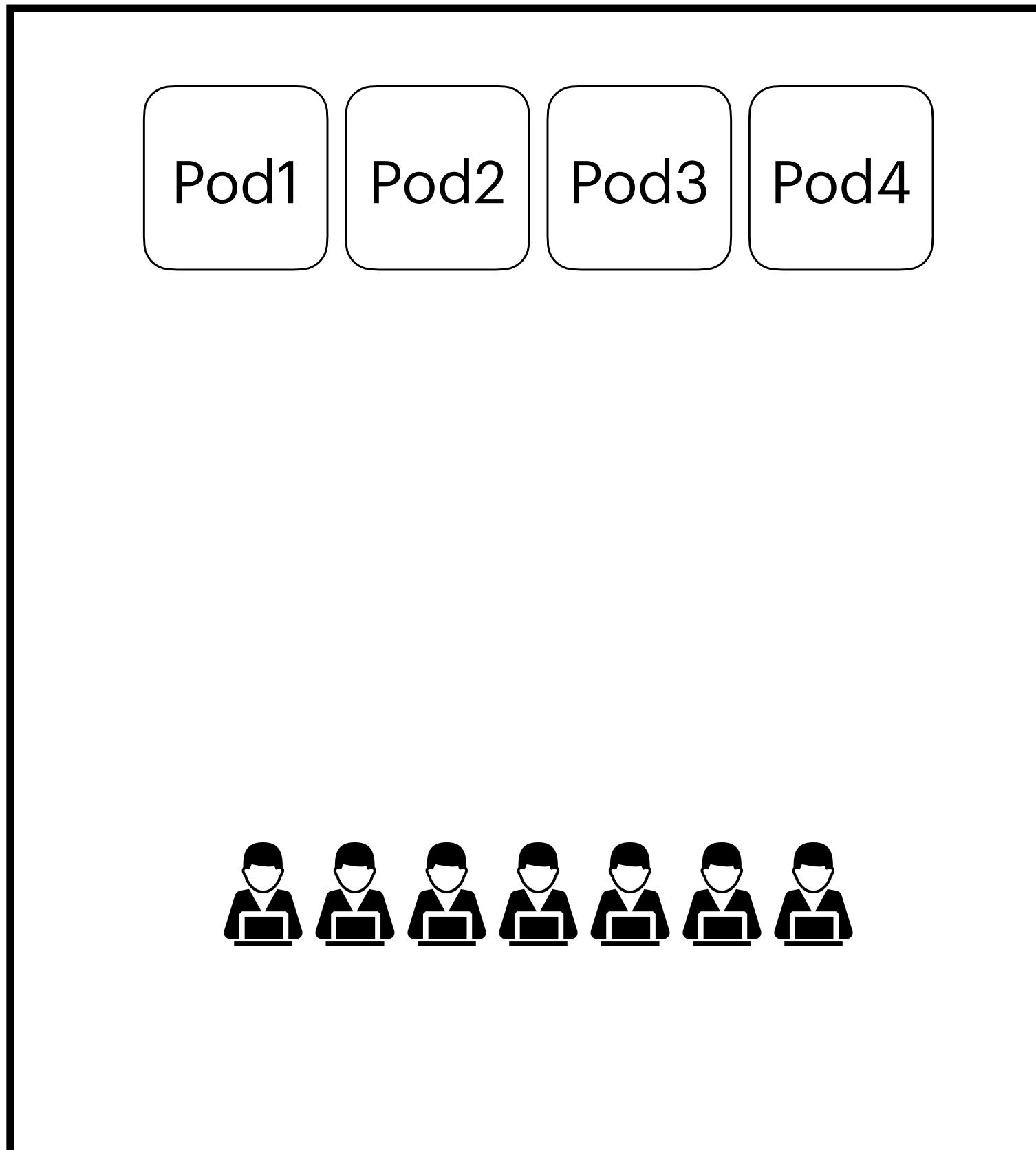


Cluster Autoscaler

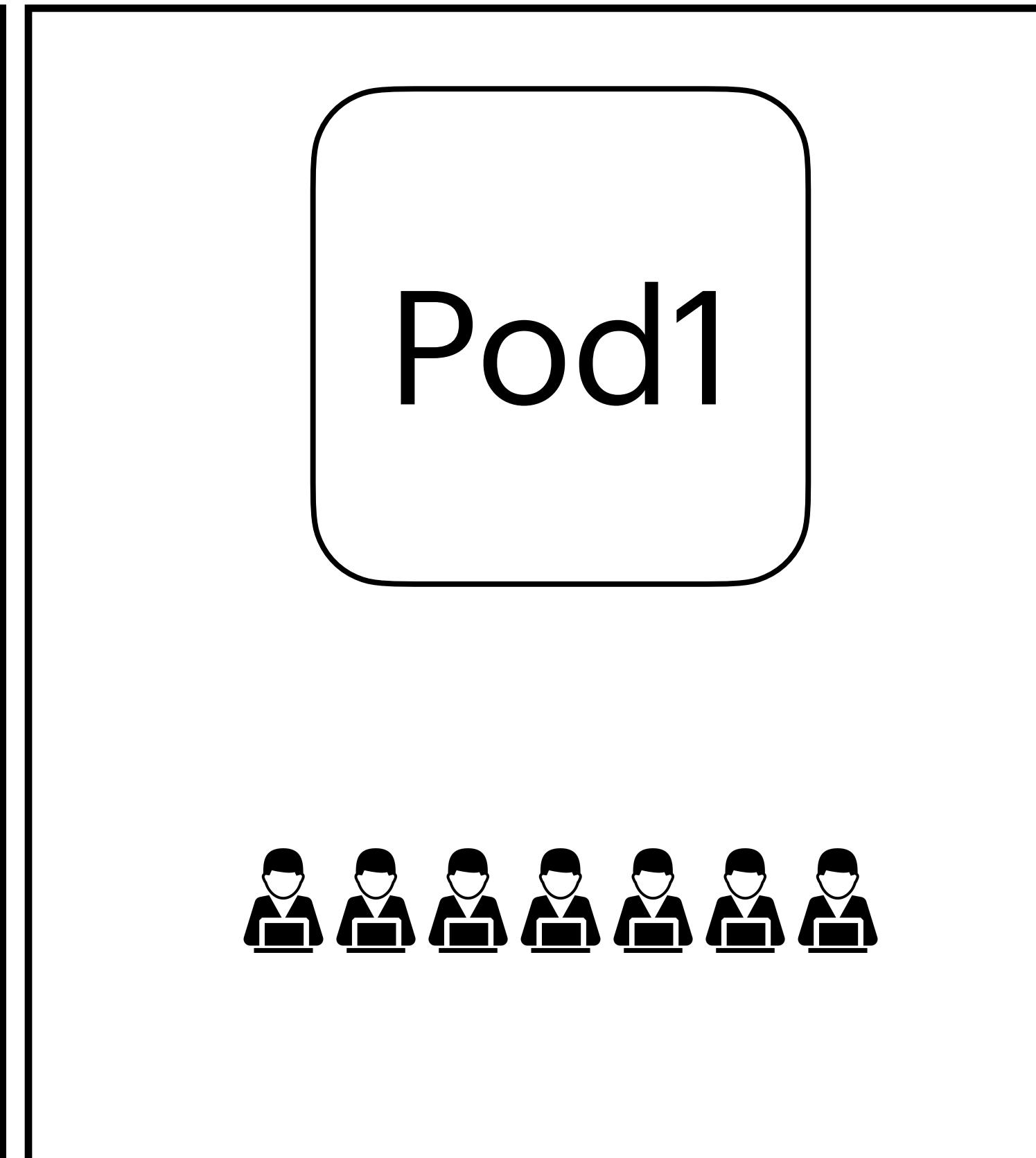


# Different types of Autoscalers in Kubernetes

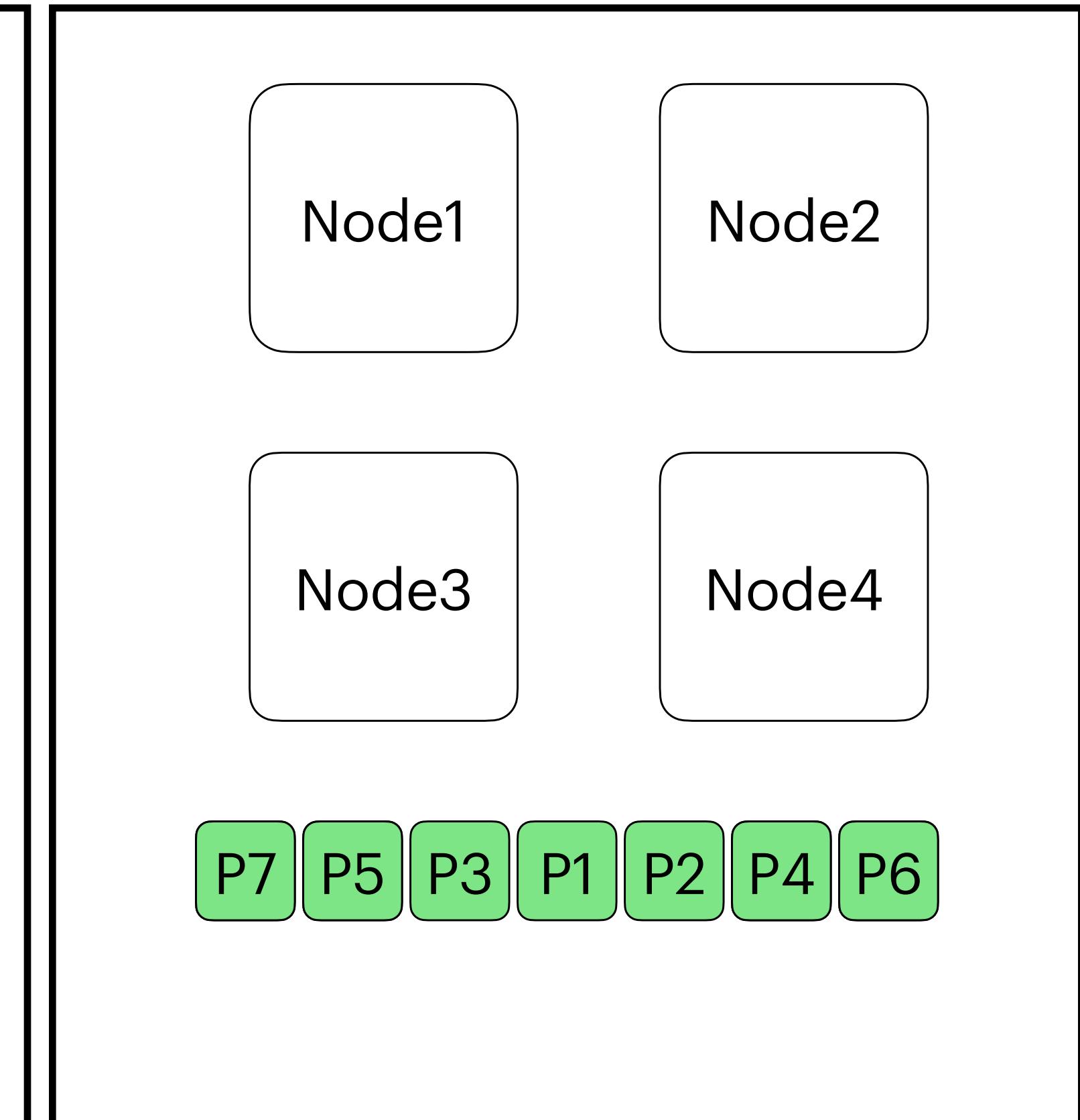
Horizontal Pod Autoscaler



Vertical Pod Autoscaler



Cluster Autoscaler





# What is Horizontal Pod Autoscaler?

# What is Horizontal Pod Autoscaler?

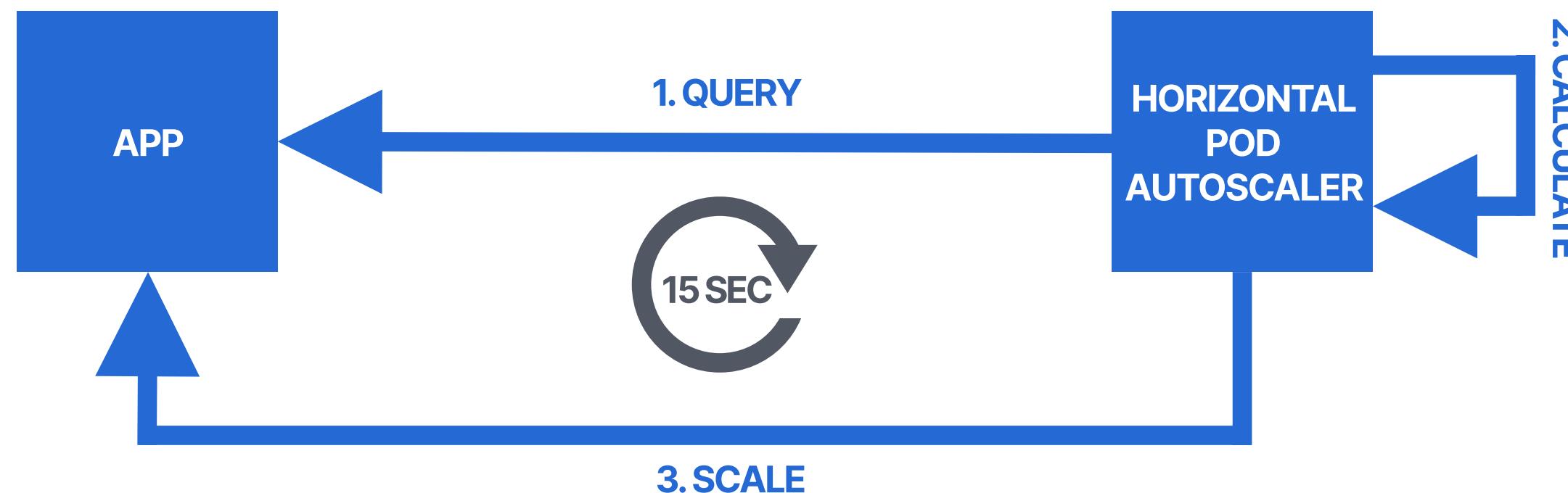
- Horizontal scaling means increasing and decreasing the number of replicas.

# What is Horizontal Pod Autoscaler?

- Horizontal scaling means increasing and decreasing the number of replicas.
- HPA is a controller and is configured by HPA resource objects.

# What is Horizontal Pod Autoscaler?

- Horizontal scaling means increasing and decreasing the number of replicas.
- HPA is a controller and is configured by HPA resource objects.
- For Autoscaling, the Horizontal Pod Autoscaler executes an eternal control loop.



- Query the scaling metric.
- Calculate the desired number of replicas.
- Scale the app to desired number of replicas.



# Horizontal Pod Autoscaler Algorithm

$$X = N^* (c/t)$$

Here,

**X** is the number of desired replicas,  
**N** is the current number of replicas,  
**c** is the current value of the metric and  
**t** is the target value.

# Horizontal Pod Autoscaler Algorithm

$$X = N^*(c/t)$$

# Horizontal Pod Autoscaler Algorithm

$$X = N^*(c/t)$$

# Horizontal Pod Autoscaler Algorithm

$$X = N^*(c/t)$$

**X = ?**

**N = 1**

**c = 100**

**t = 50**

# Horizontal Pod Autoscaler Algorithm

**X = ?**

**N = 1**

**c = 100**

**t = 50**

# Horizontal Pod Autoscaler Algorithm

**X = ?**

**N = 1**

**c = 100**

**t = 50**

# Horizontal Pod Autoscaler Algorithm

$$X = 1 * (100 / 50)$$

X = ?

N = 1

c = 100

t = 50

# Horizontal Pod Autoscaler Algorithm

**X = ?**

**N = 1**

**c = 100**

**t = 50**

# Horizontal Pod Autoscaler Algorithm

**X = ?**

**N = 1**

**c = 100**

**t = 50**

# Horizontal Pod Autoscaler Algorithm

X = 2

X = ?

N = 1

c = 100

t = 50





# Configure HPA through HPA Resource

```
apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata:
  name: MyHPA
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: MyApp
  minReplicas: 1
  maxReplicas: 10
  metrics:
    - type: Resource
      resource:
        name: cpu
      target:
        type: Utilization
        averageUtilization: 50
```

# Configure HPA through HPA Resource

```
apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata:
  name: MyHPA
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: MyApp
  minReplicas: 1
  maxReplicas: 10
  metrics:
    - type: Resource
      resource:
        name: cpu
      target:
        type: Utilization
        averageUtilization: 50
```

The deployment we are going to autoscale.

# Configure HPA through HPA Resource

```
apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata:
  name: MyHPA
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: MyApp
  minReplicas: 1
  maxReplicas: 10
  metrics:
    - type: Resource
      resource:
        name: cpu
      target:
        type: Utilization
        averageUtilization: 50
```

The deployment we are going to autoscale.

The scaling metric.

# Configure HPA through HPA Resource

```
apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata:
  name: MyHPA
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: MyApp
  minReplicas: 1
  maxReplicas: 10
  metrics:
    - type: Resource
      resource:
        name: cpu
      target:
        type: Utilization
        averageUtilization: 50
```

The deployment we are going to autoscale.

The scaling metric.

The scaling metrics target value.

# Configure HPA through HPA Resource

```
apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata:
  name: MyHPA
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: MyApp
  minReplicas: 1
  maxReplicas: 10
  metrics:
    - type: Resource
      resource:
        name: cpu
      target:
        type: Utilization
        averageUtilization: 50
```

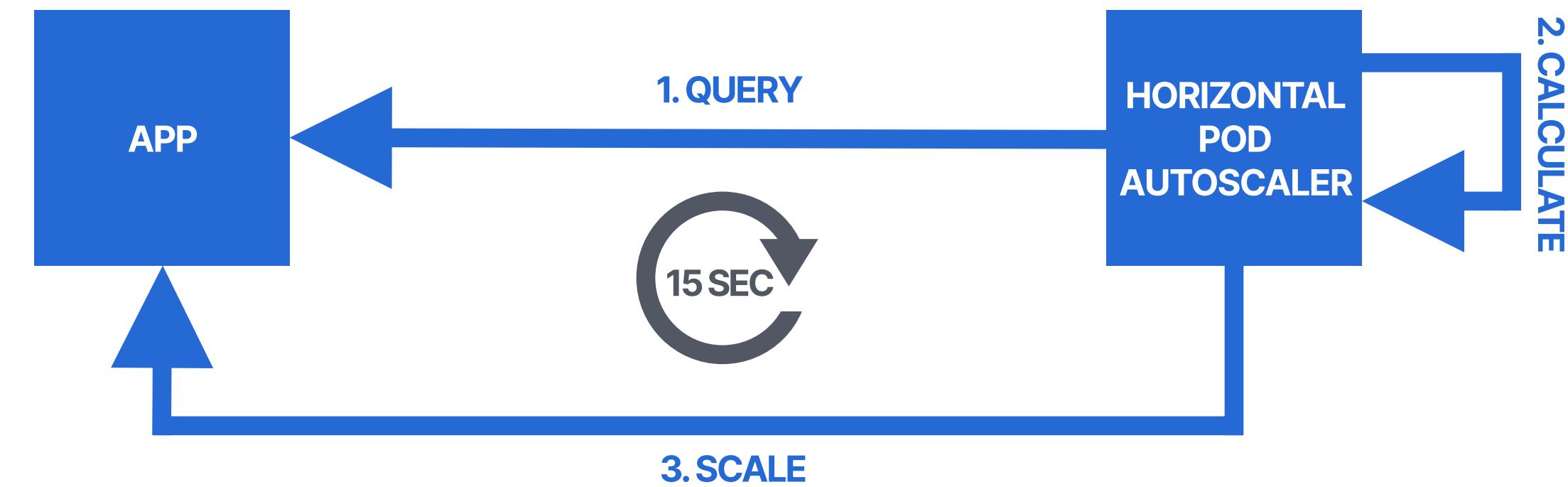
The deployment we are going to autoscale.

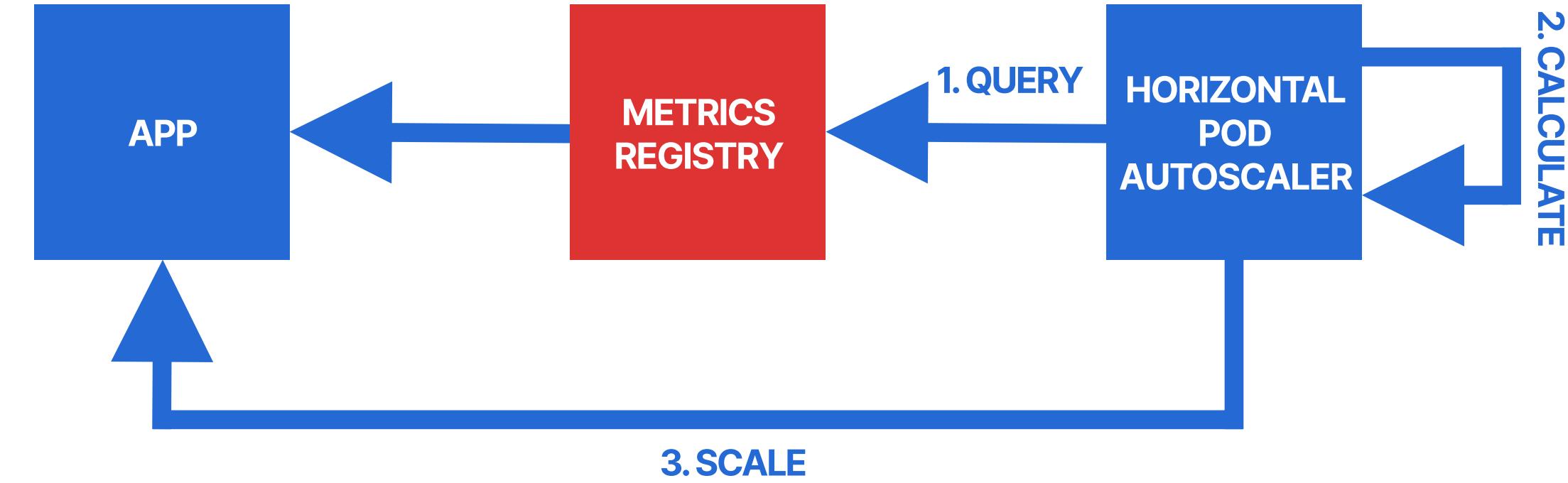
The scaling metric.

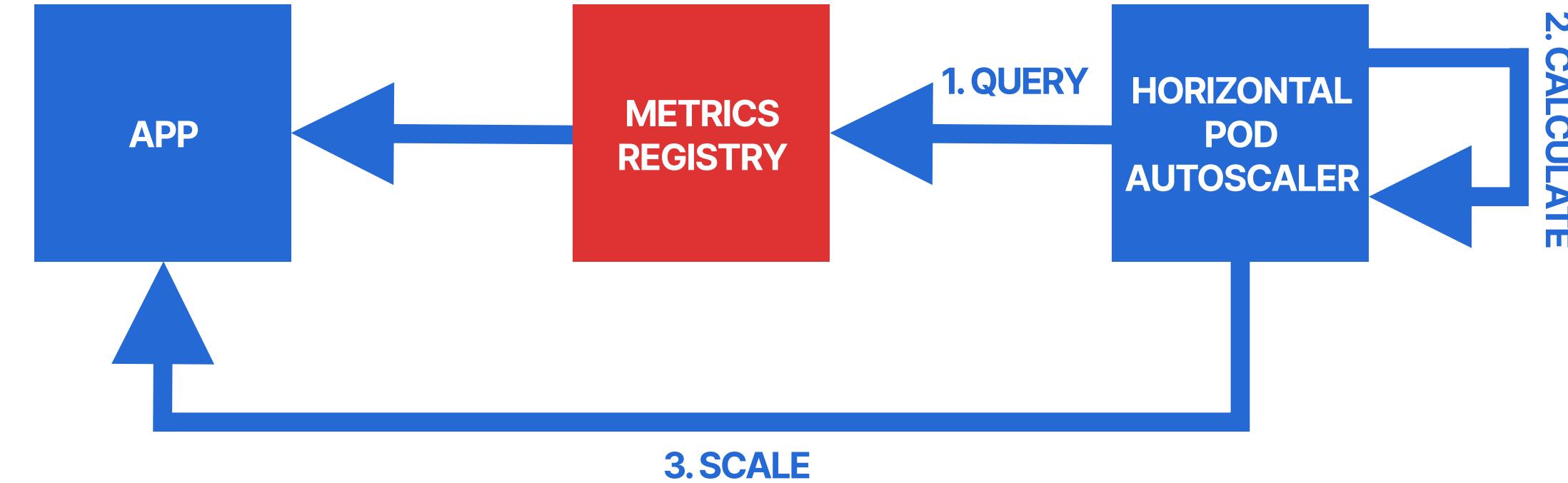
The scaling metrics target value.

Minimum and maximum deployment size

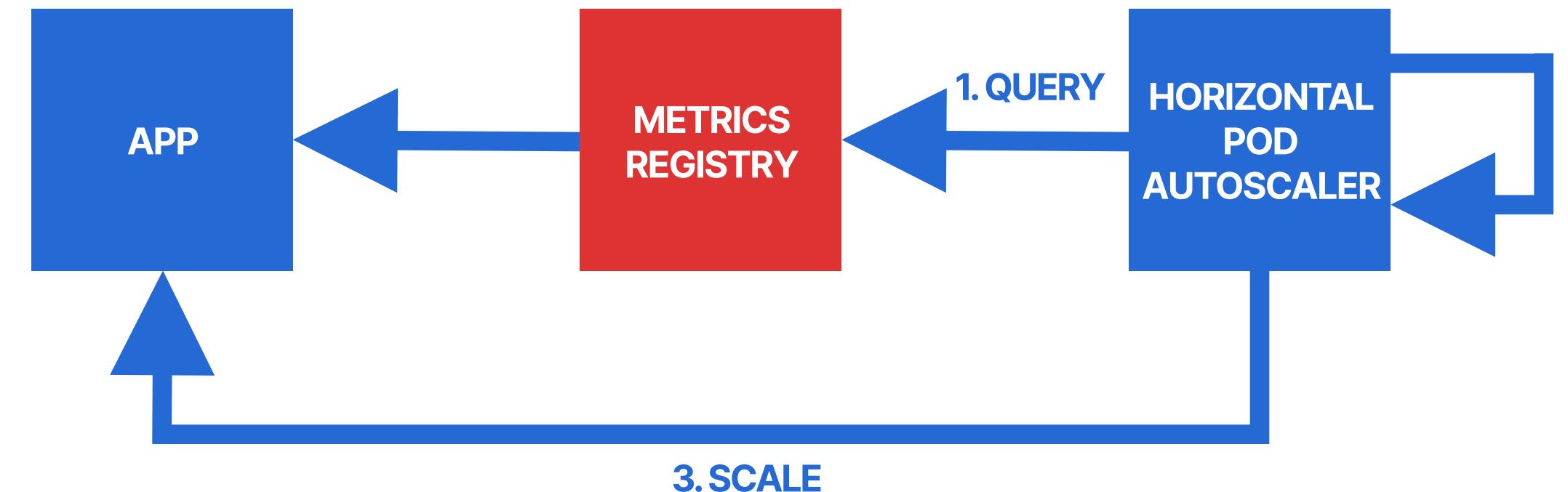






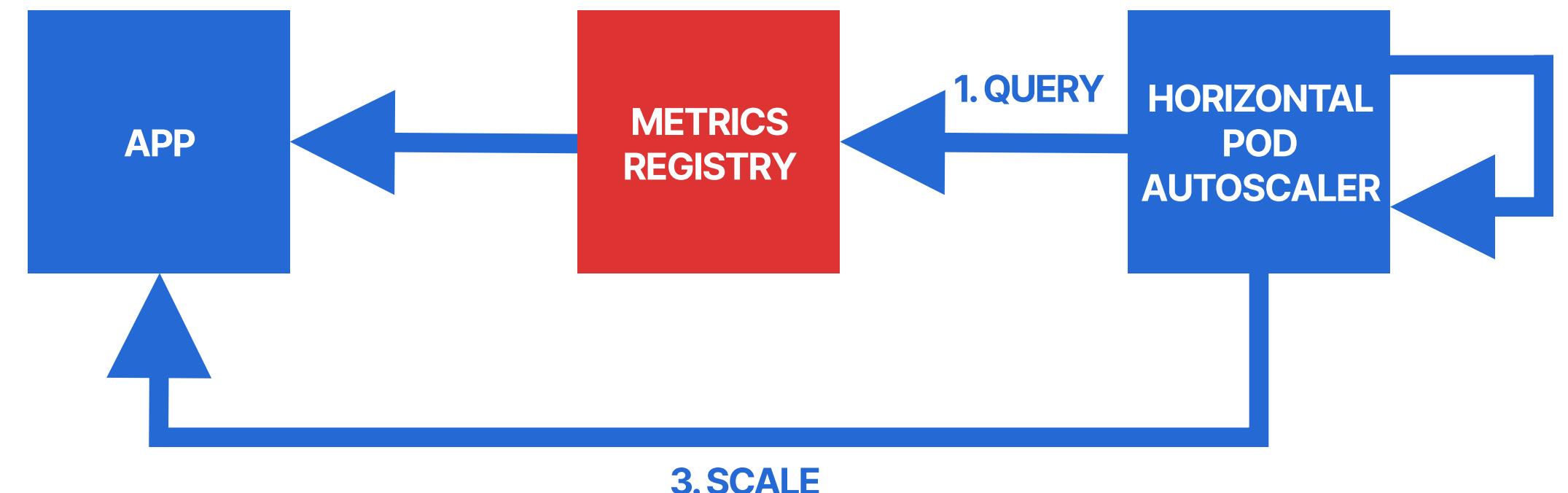


# What is metrics registry?



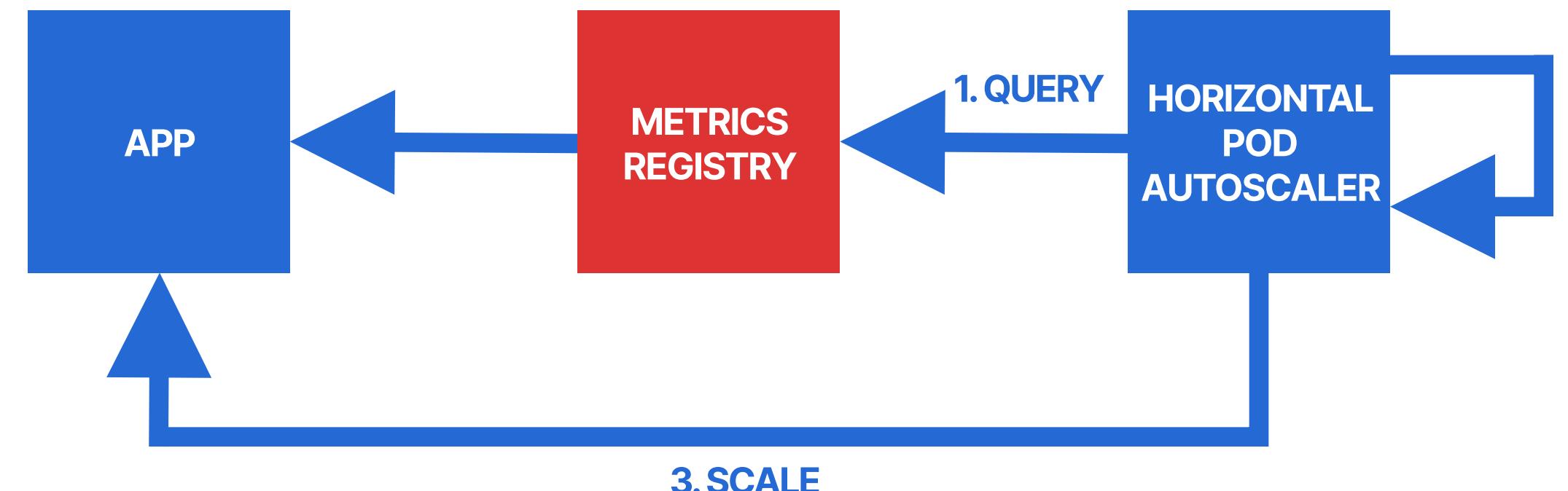
# What is metrics registry?

- The metric registry is a central place in the cluster where metrics (of any kind) are exposed to clients (of any kind).



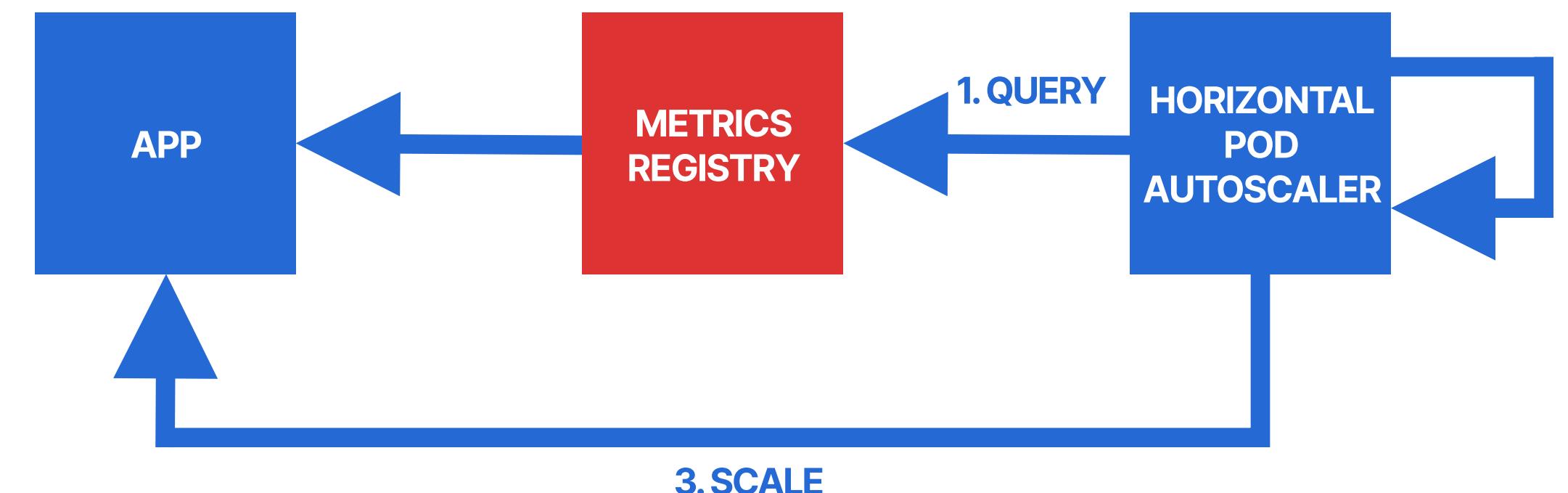
# What is metrics registry?

- The metric registry is a central place in the cluster where metrics (of any kind) are exposed to clients (of any kind).
- The HPA queries metrics from the metrics registry.



# What is metrics registry?

- The metric registry is a central place in the cluster where metrics (of any kind) are exposed to clients (of any kind).
- The HPA queries metrics from the metrics registry.
- The purpose of metrics registry is to provide a standard interface for clients to query metrics from.

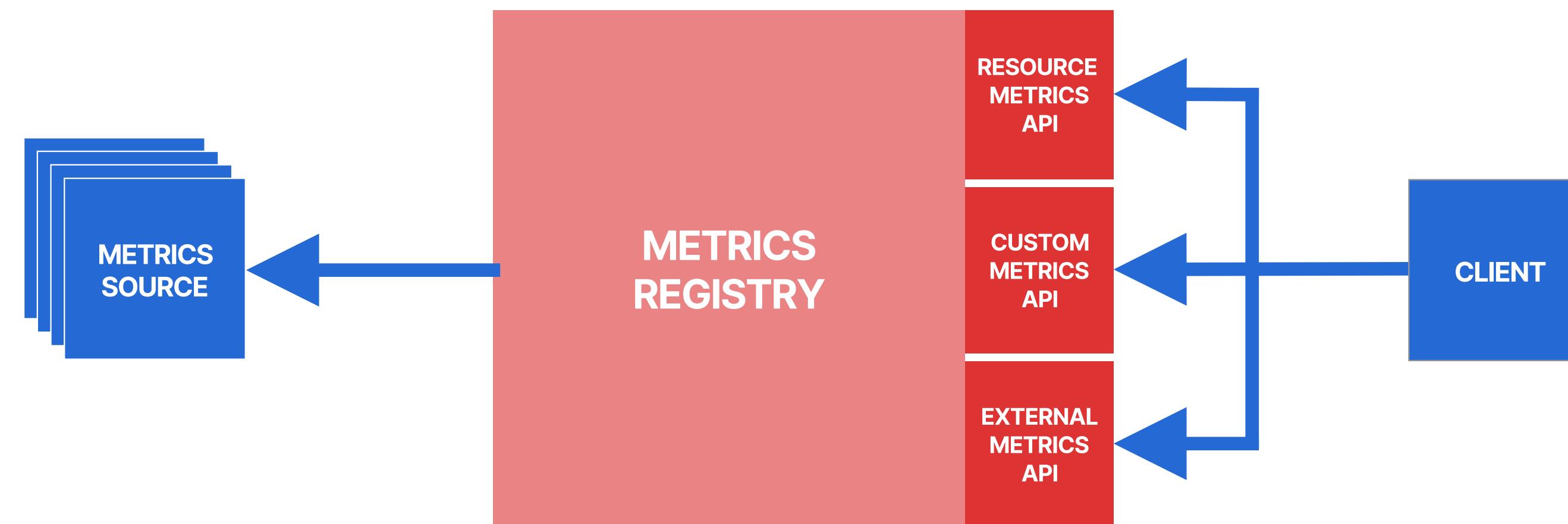


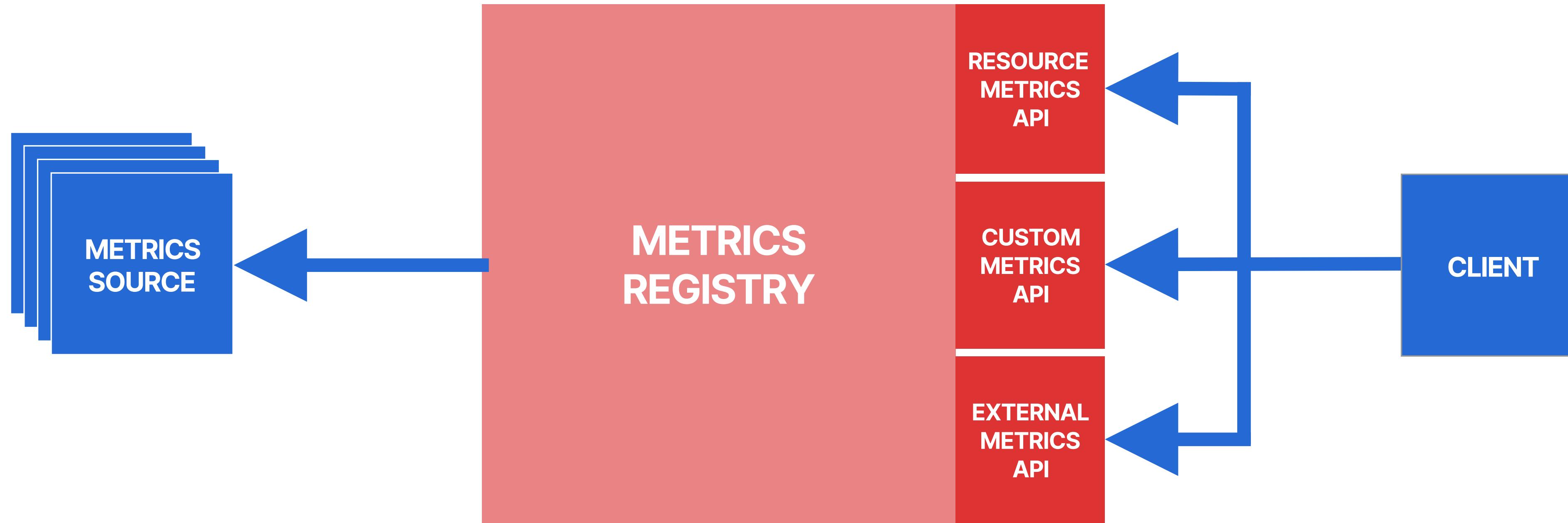


# Types of Metrics APIs

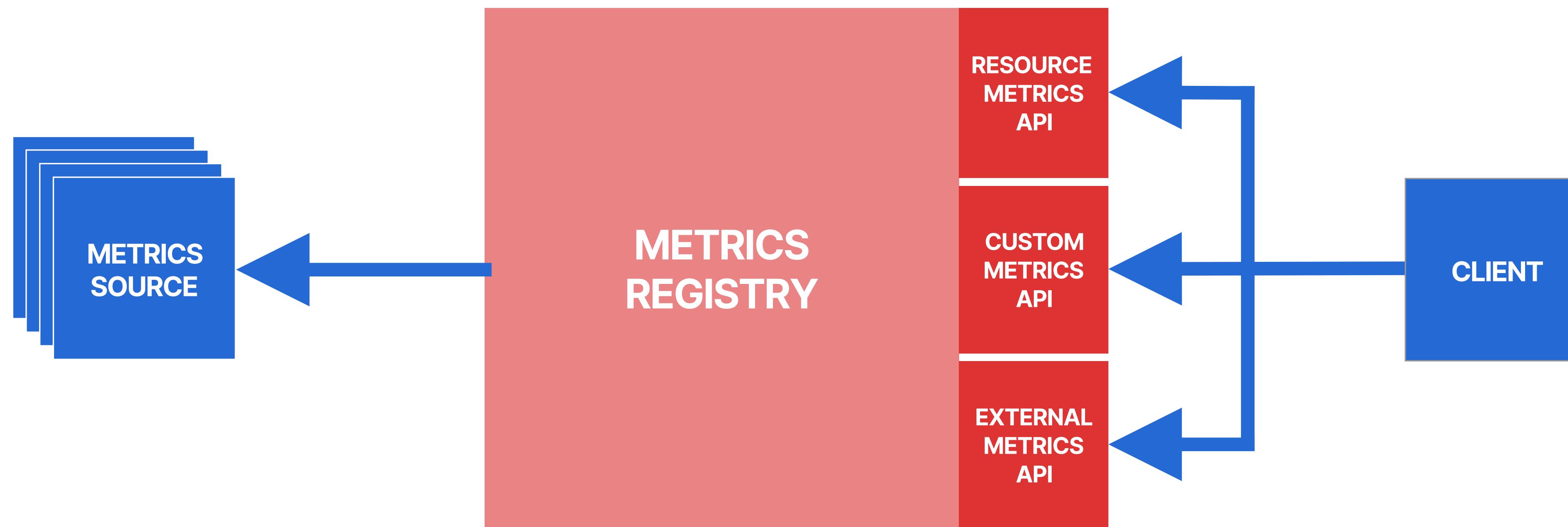
There are three different types of extension APIs designed to serve different type of metrics:

- Resource Metrics API
- Custom Metrics API
- External Metrics API

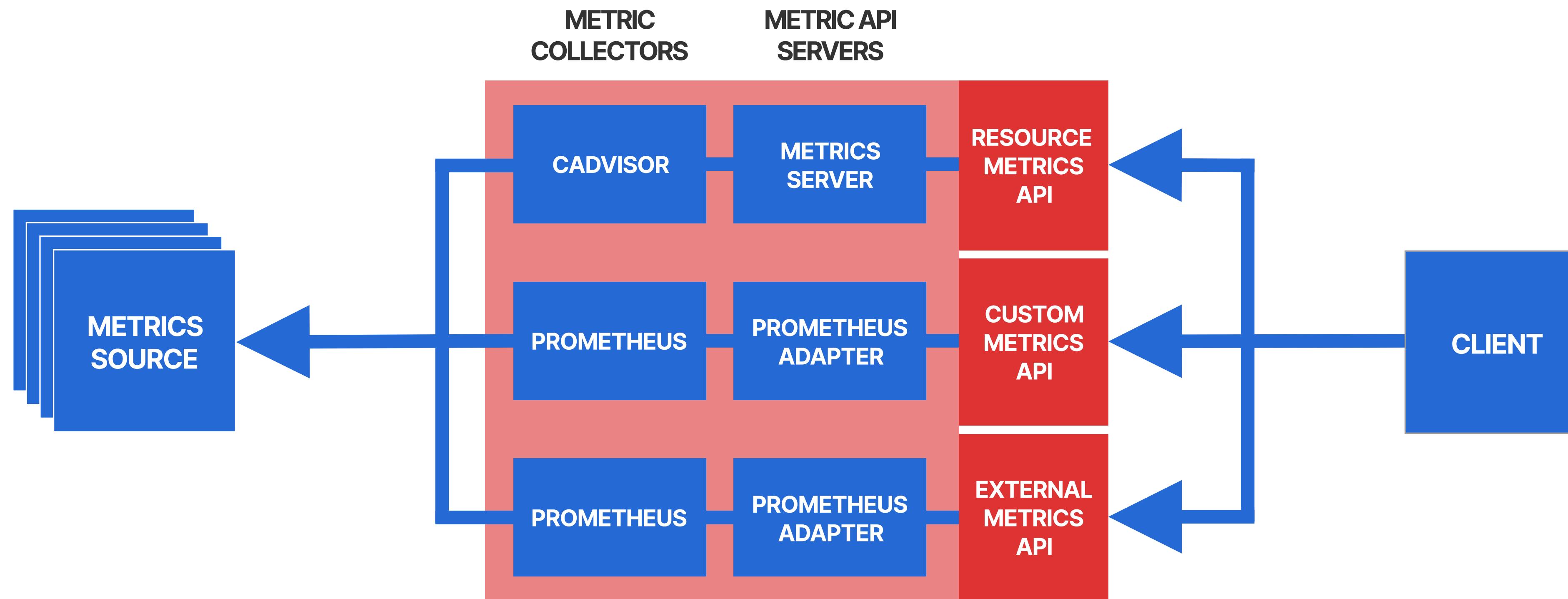




# How to expose a Metric through Metric Registry?



# How to expose a Metric through Metric Registry?





# Conclusion

## Requirements for Autoscaling

- Configure Horizontal Pod Autoscaler
- Setup a metrics collector and configure it to collect the desired metric from source.
- Install a metric API server and configure it to expose the metrics through the corresponding metrics API.

# Code Demo



# Code Demo





# Custom Metrics in Action

We have a pod running a HTTP server.

The pod exposes 2 end points, a GET and a PUT api.

The GET api at / returns an integer.

This integer is the custom metric value for that pod.

The PUT api at /:metric is an api meant to set the same integer.

app-pod-1  
metric : 0

# Custom Metrics in Action

app-pod-1  
metric : 0

# Custom Metrics in Action

app-pod-1  
metric : 0

# Custom Metrics in Action

The pod is part of a deployment which is currently running one replica.

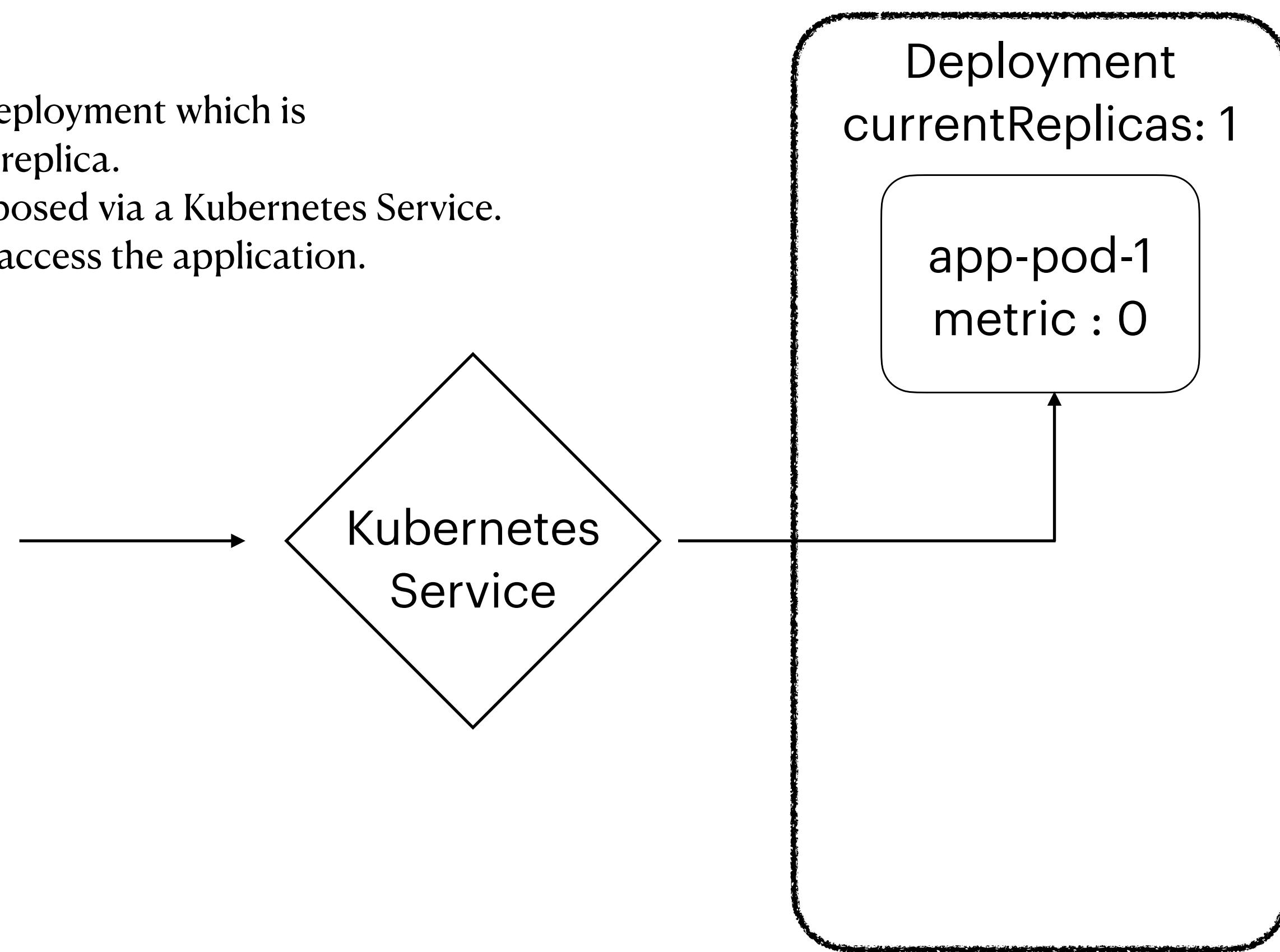
The deployment is exposed via a Kubernetes Service.  
We use this service to access the application.

app-pod-1  
metric : 0

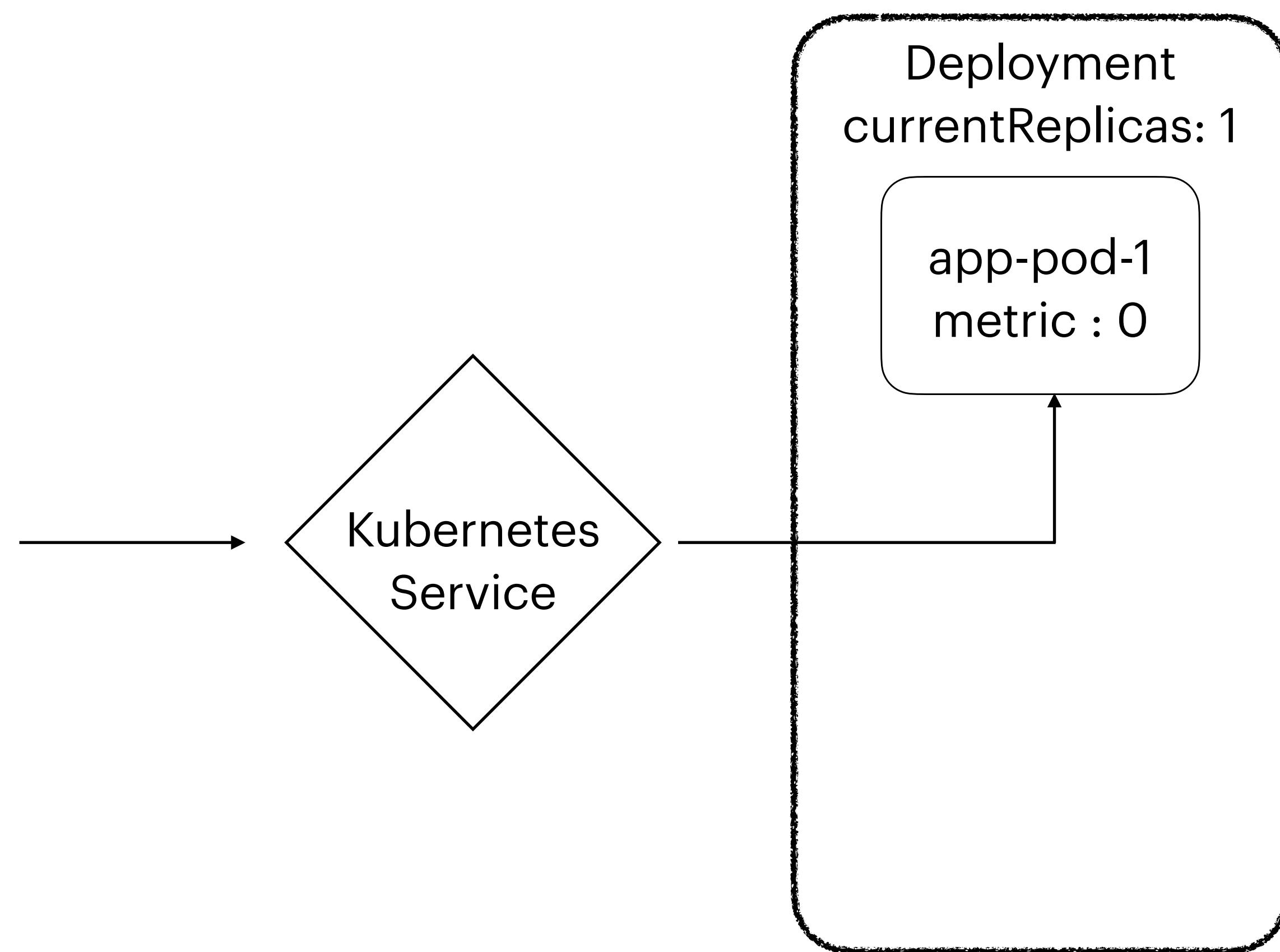
# Custom Metrics in Action

The pod is part of a deployment which is currently running one replica.

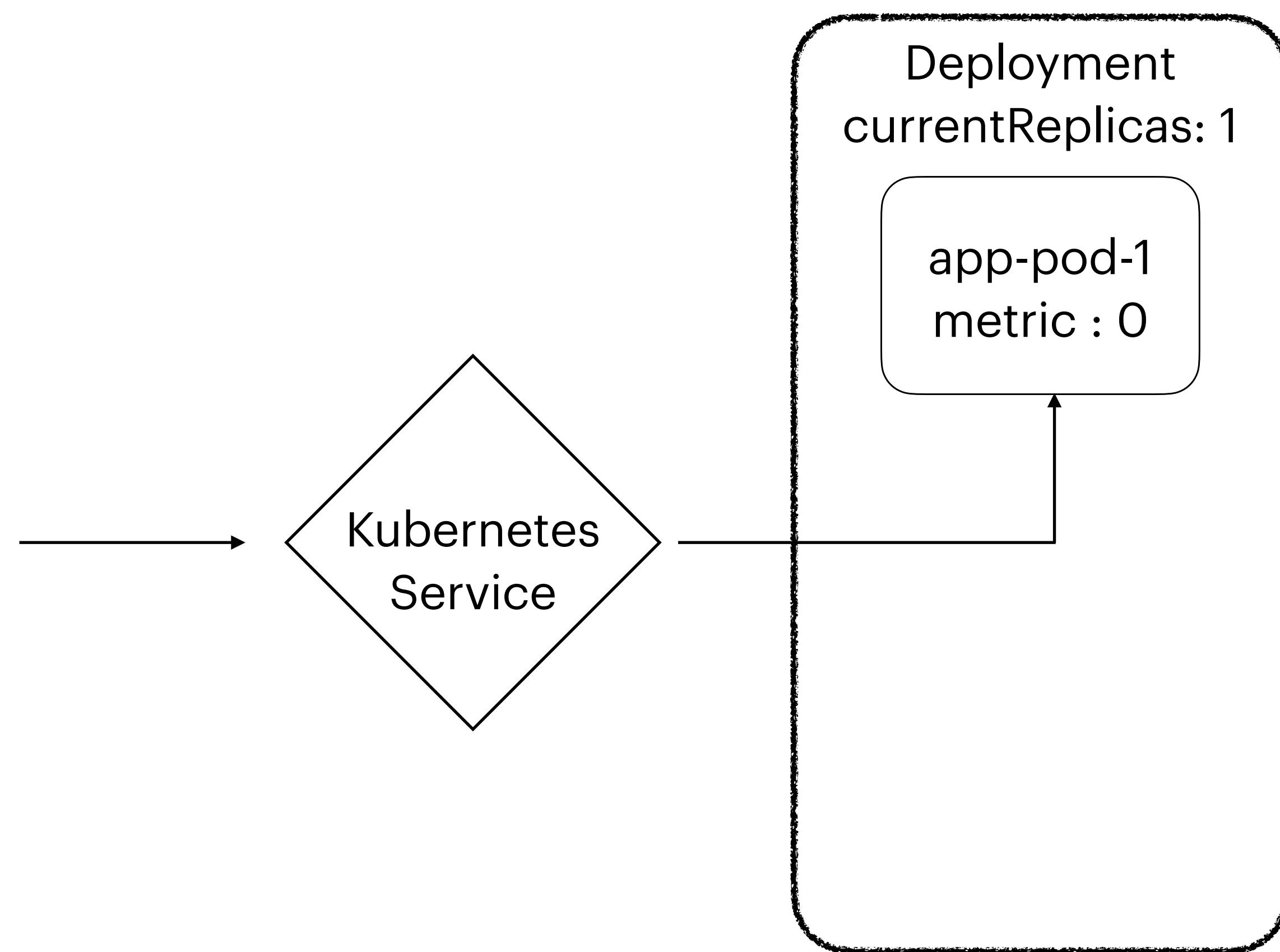
The deployment is exposed via a Kubernetes Service.  
We use this service to access the application.



# Custom Metrics in Action



# Custom Metrics in Action

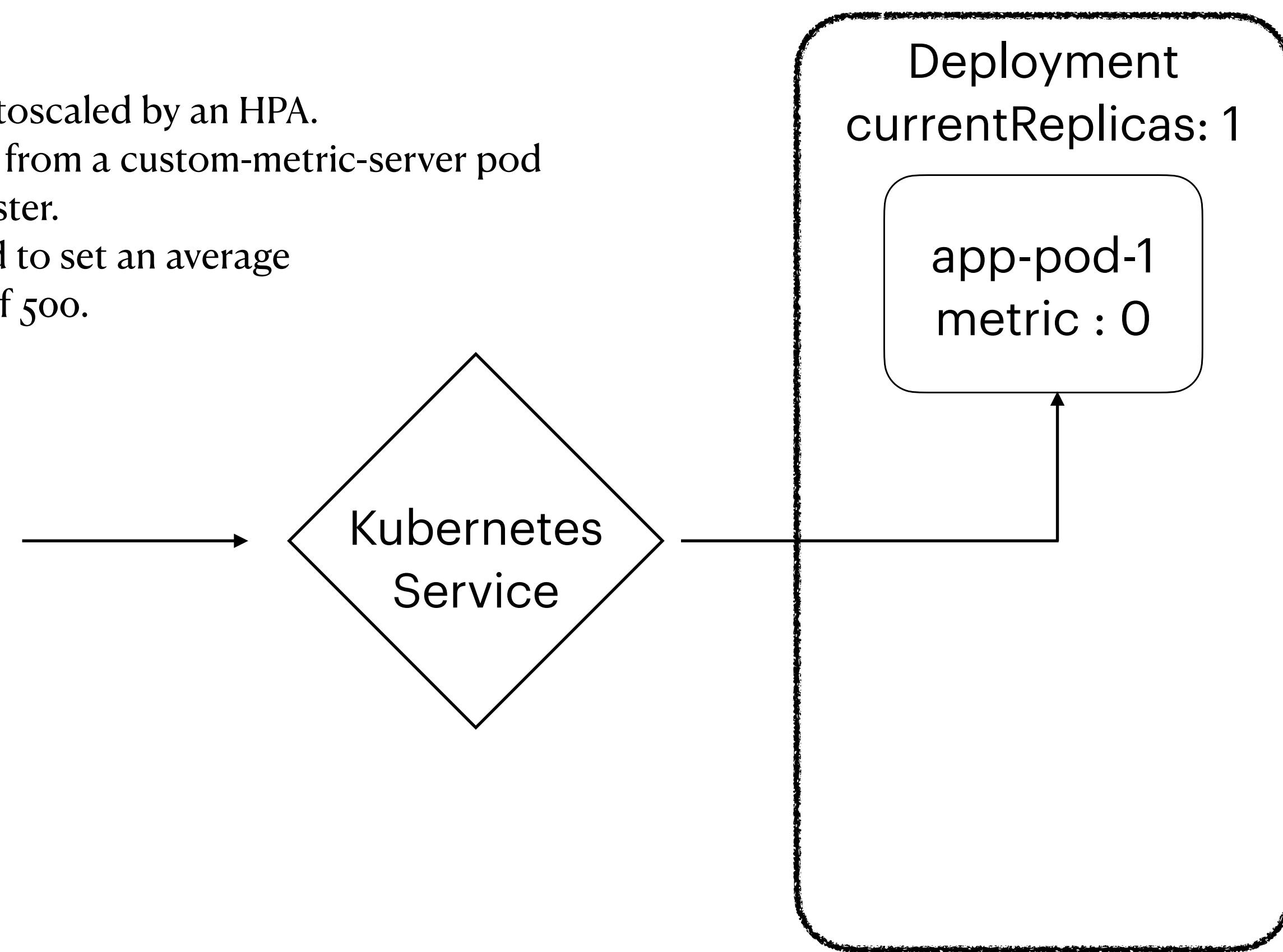


# Custom Metrics in Action

The deployment is autoscaled by an HPA.

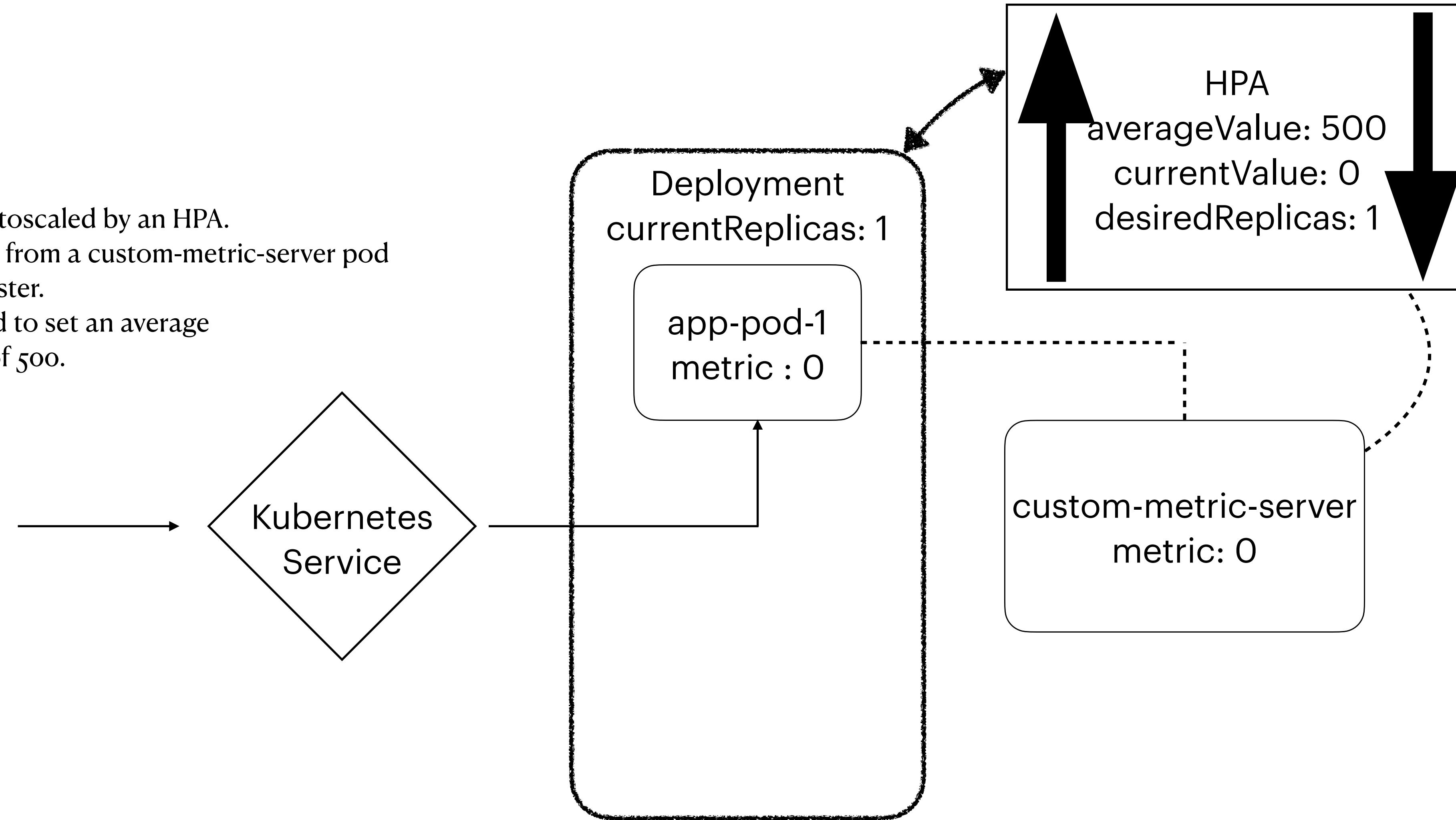
The HPA uses metrics from a custom-metric-server pod running inside the cluster.

The HPA is configured to set an average custom metric value of 500.

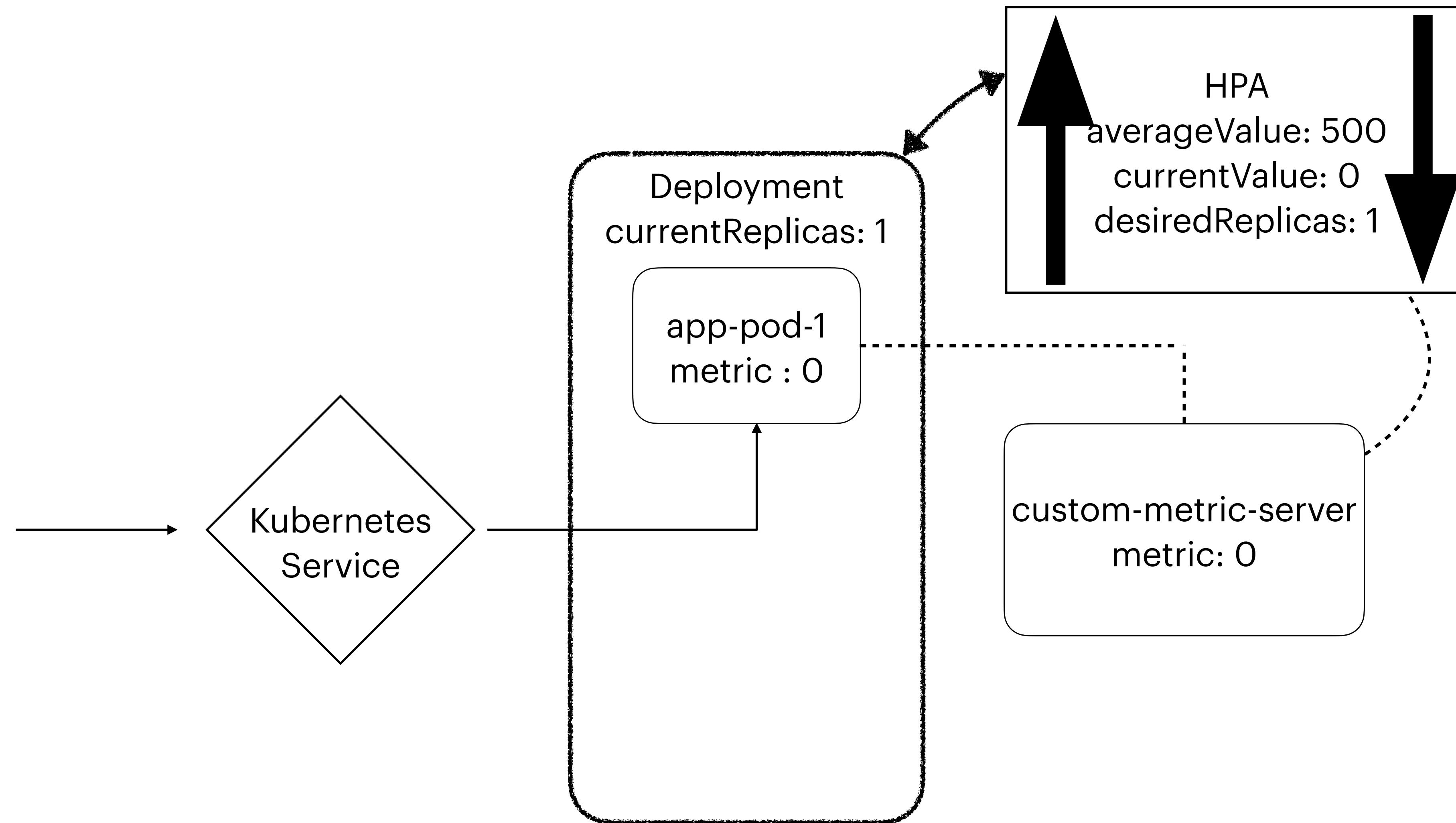


# Custom Metrics in Action

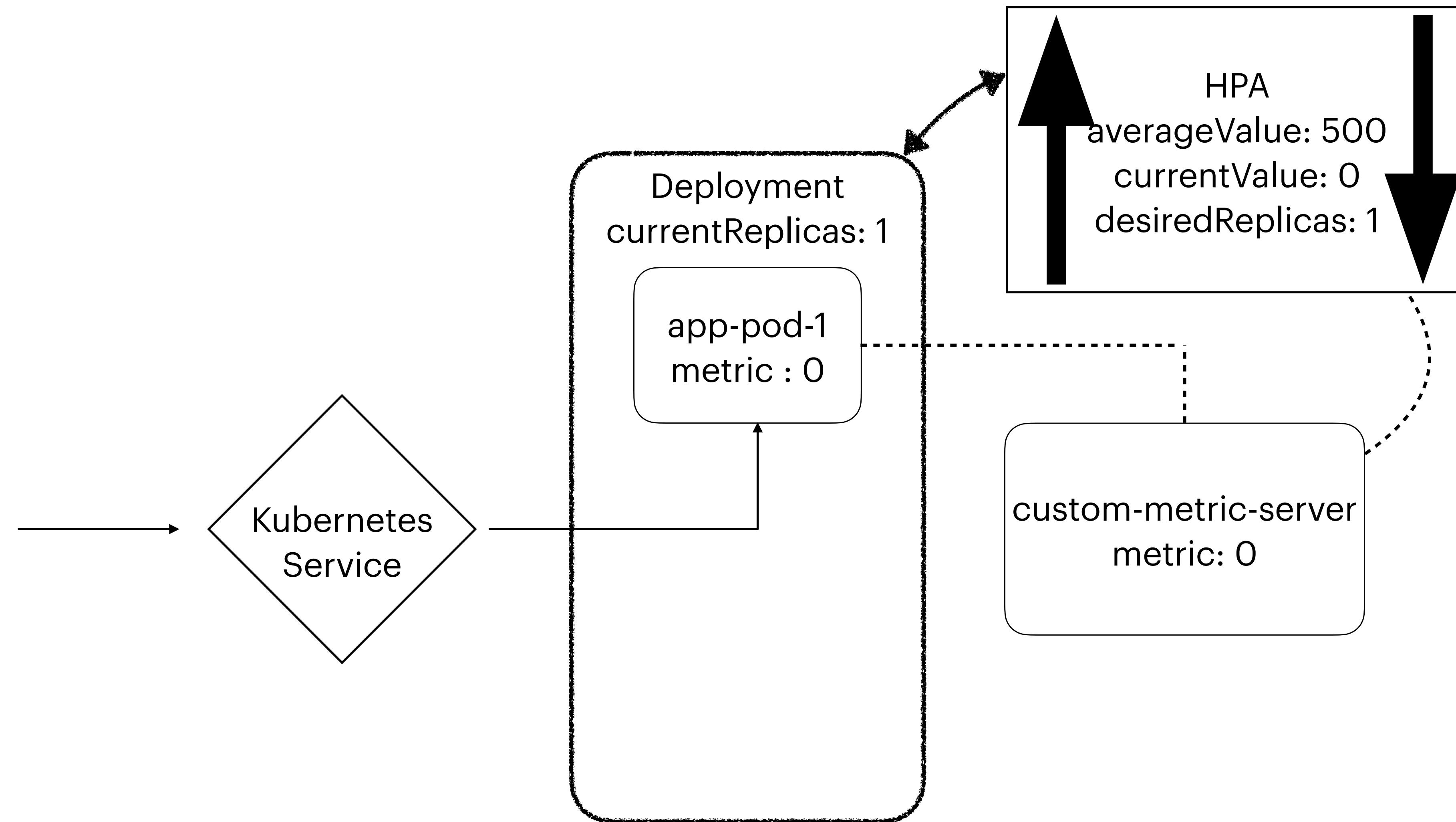
The deployment is autoscaled by an HPA.  
The HPA uses metrics from a custom-metric-server pod running inside the cluster.  
The HPA is configured to set an average custom metric value of 500.



# Custom Metrics in Action

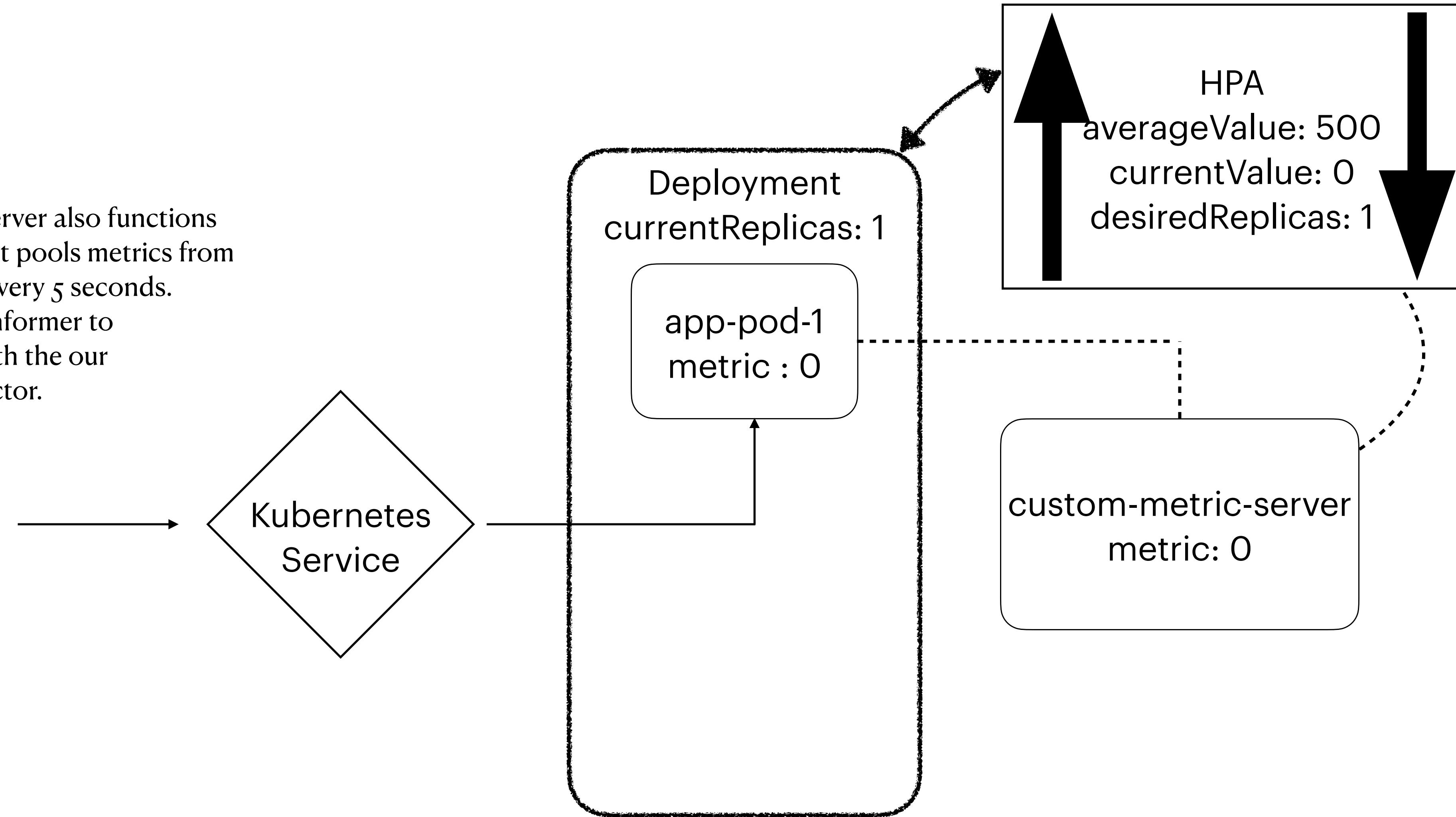


# Custom Metrics in Action

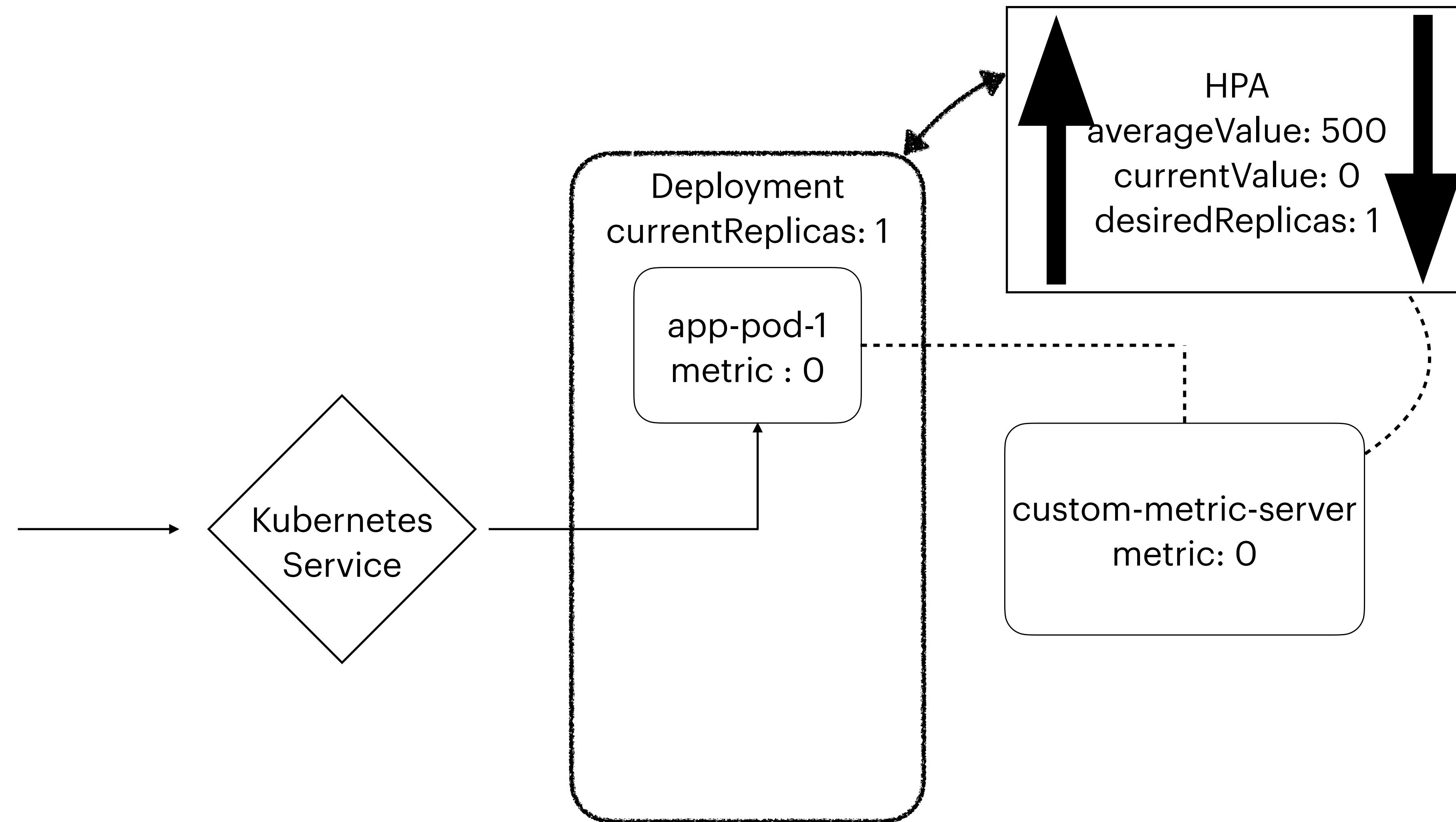


# Custom Metrics in Action

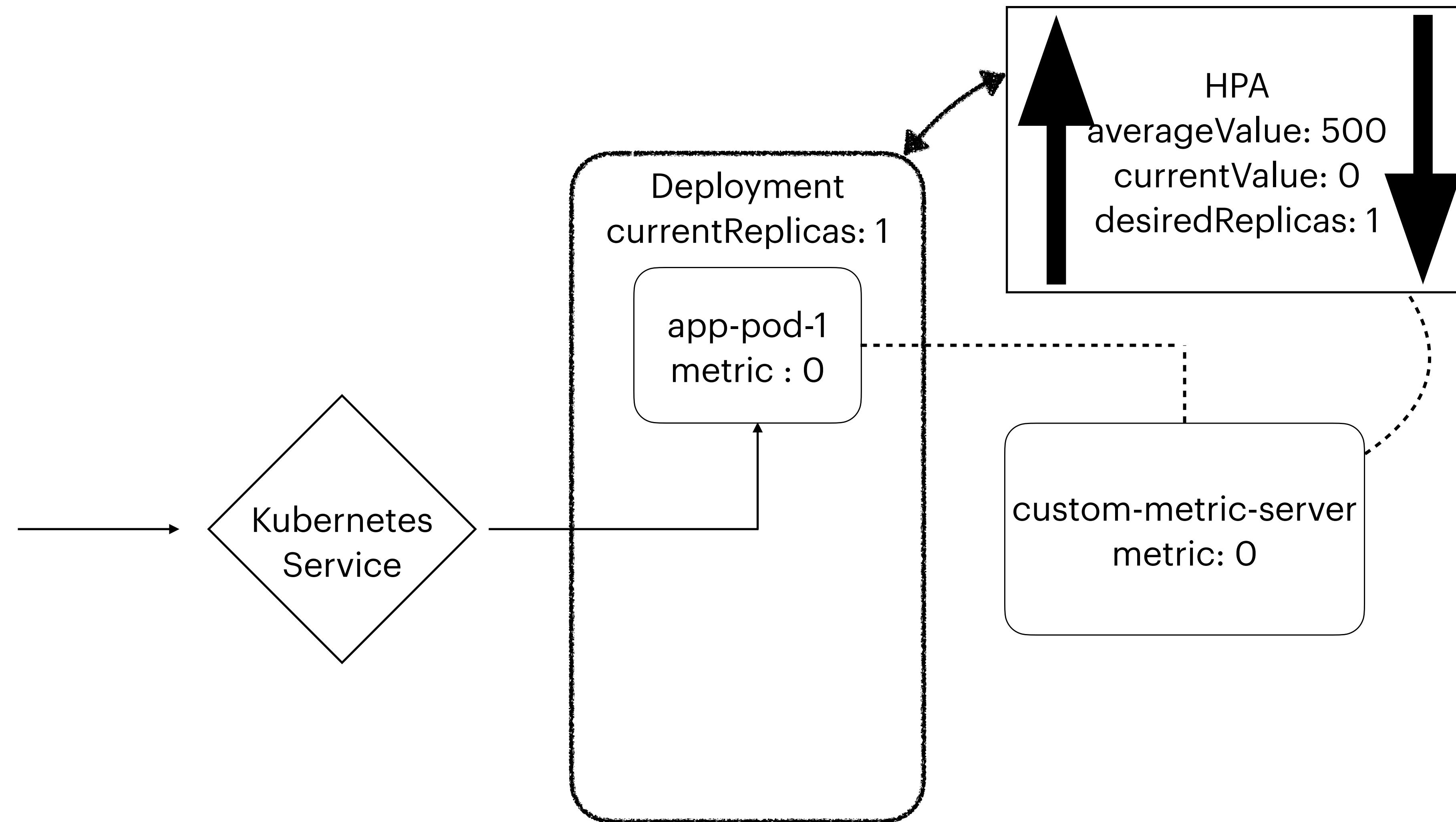
The custom-metric-server also functions as a metric collector. It pools metrics from all application pods every 5 seconds. It uses a kubernetes informer to find ips of all pods with the our application label selector.



# Custom Metrics in Action

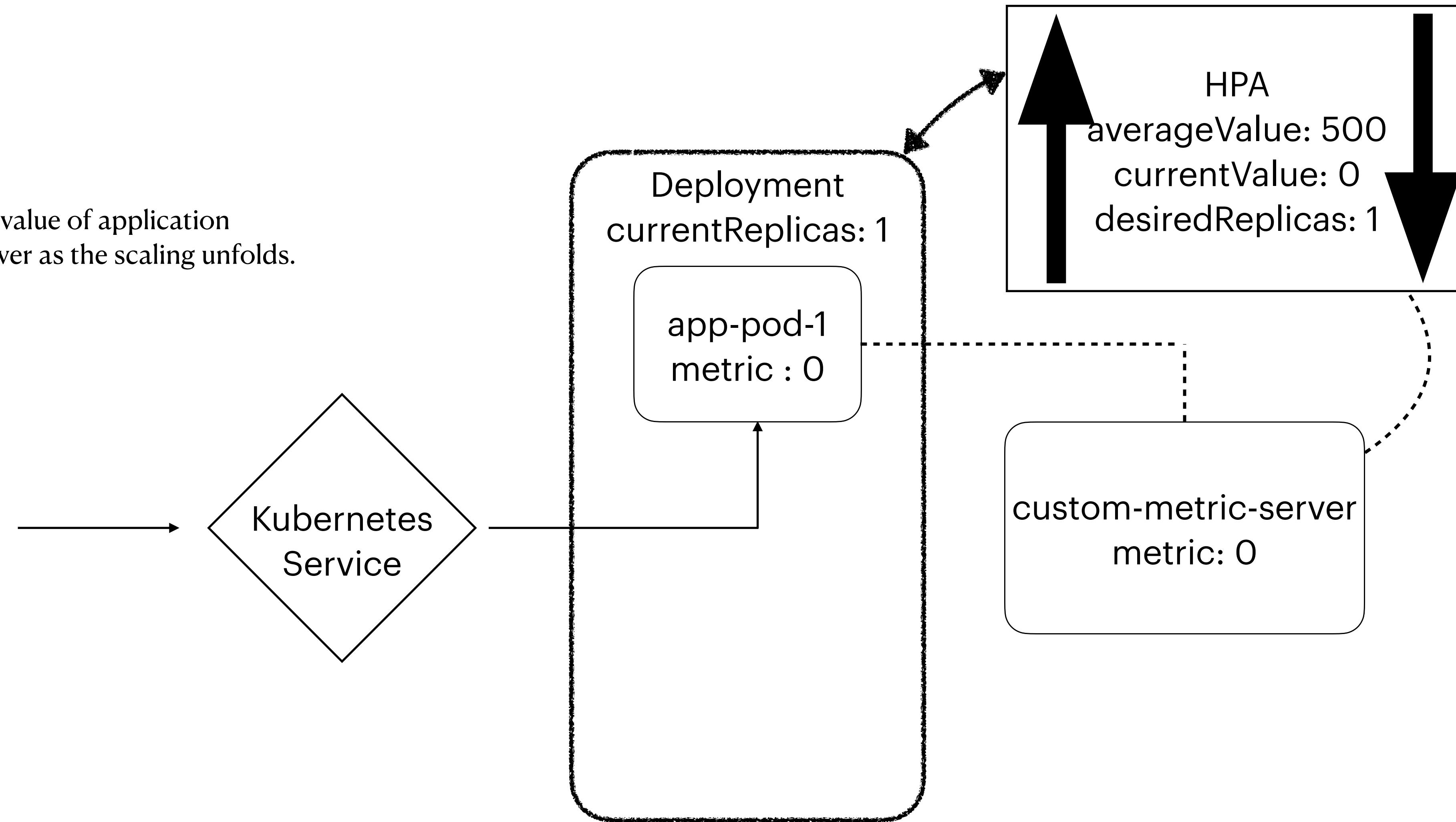


# Custom Metrics in Action



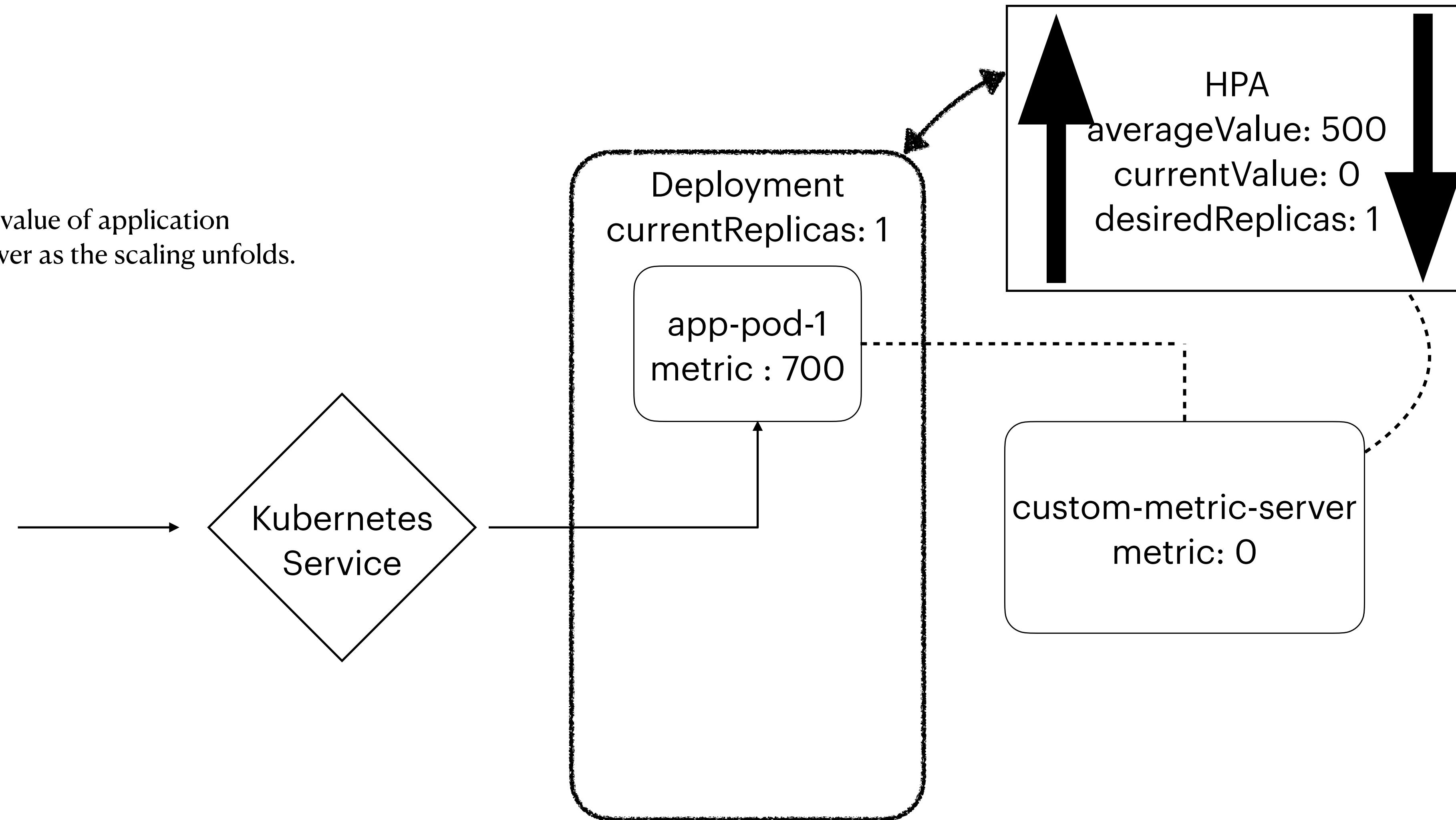
# Custom Metrics in Action

We will set the metric value of application pod to 700 and observe as the scaling unfolds.

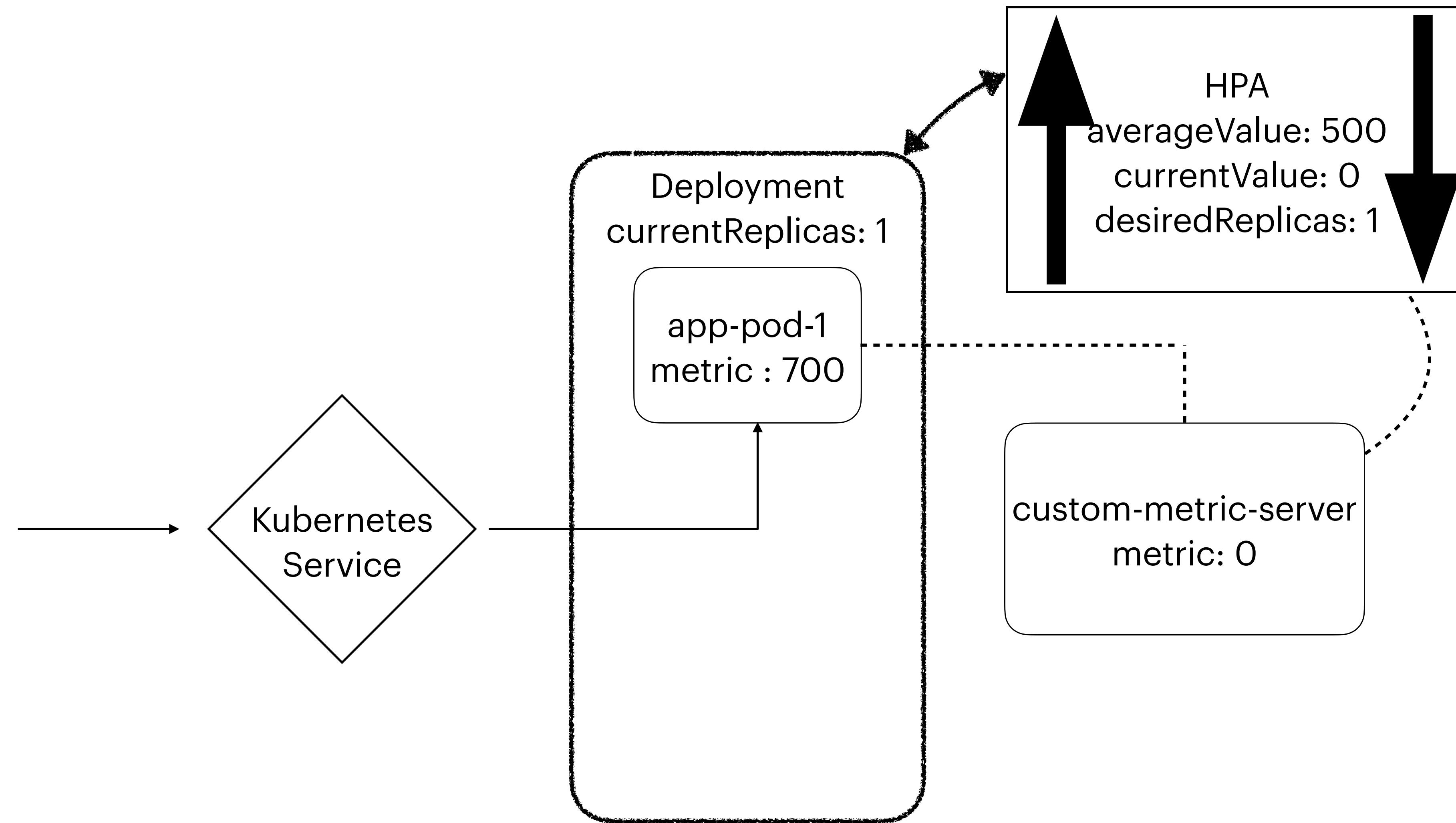


# Custom Metrics in Action

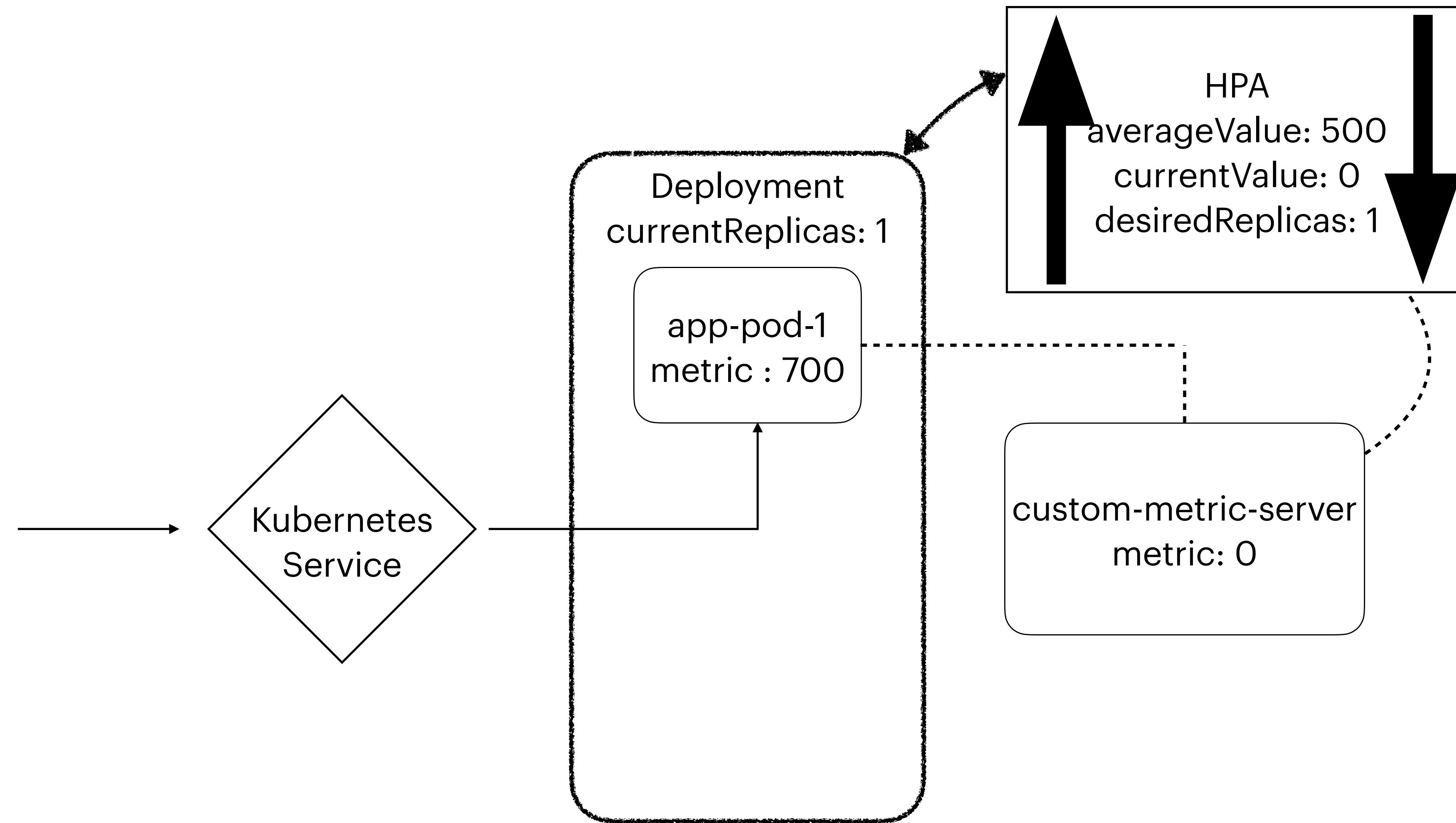
We will set the metric value of application pod to 700 and observe as the scaling unfolds.



# Custom Metrics in Action

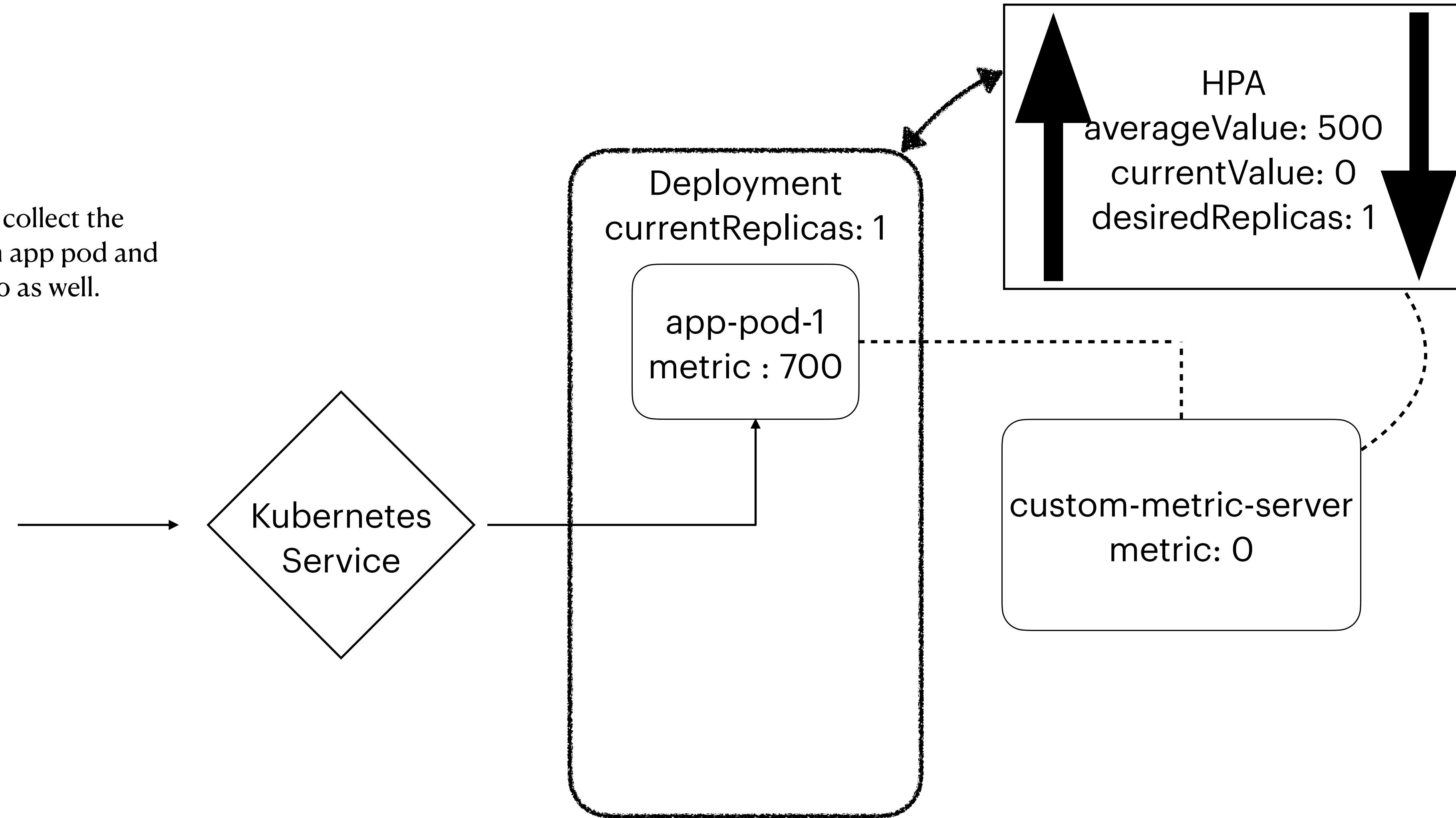


# Custom Metrics in Action



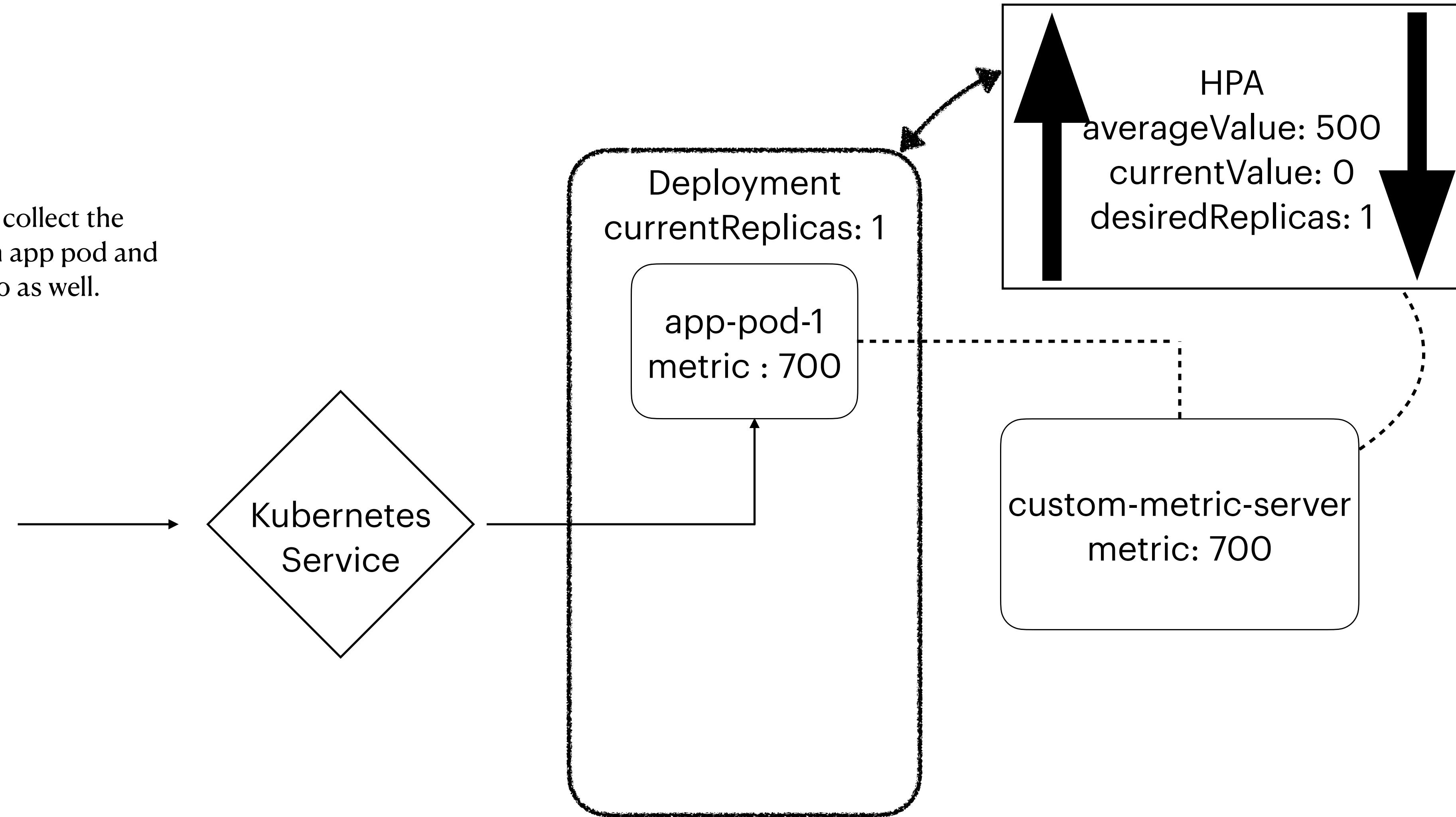
# Custom Metrics in Action

The metric server will collect the new metric value from app pod and change its value as 700 as well.

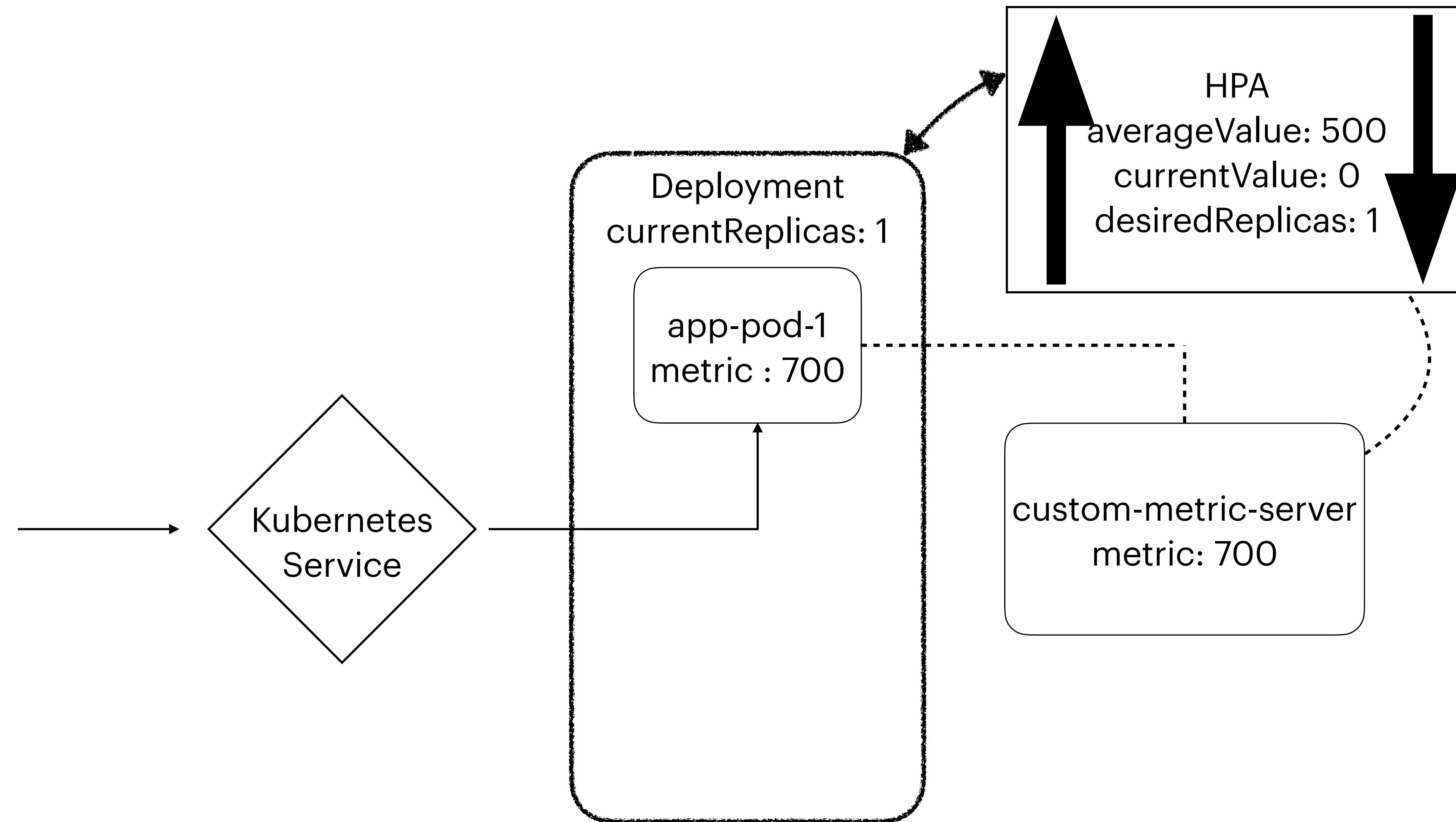


# Custom Metrics in Action

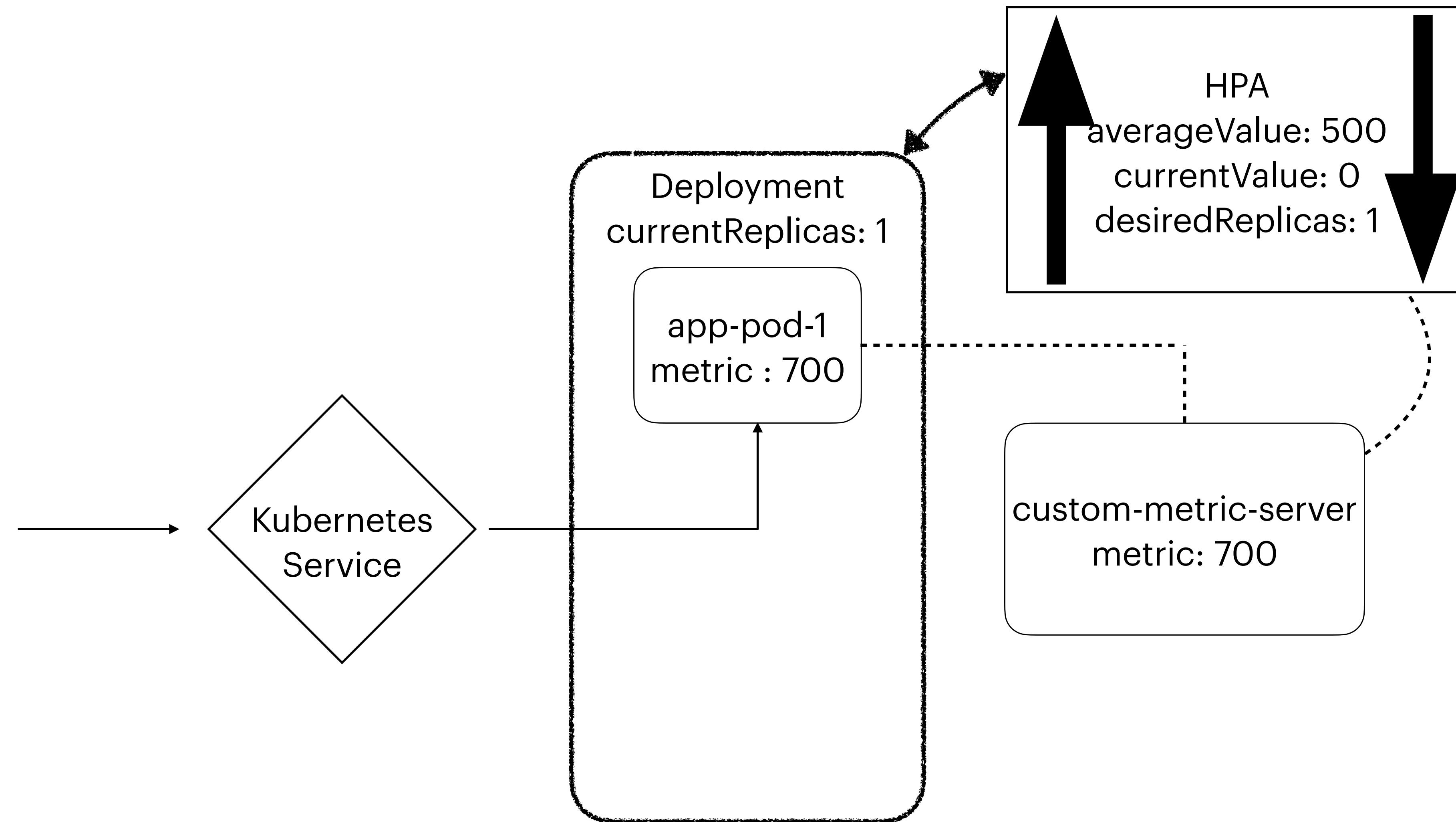
The metric server will collect the new metric value from app pod and change its value as 700 as well.



# Custom Metrics in Action

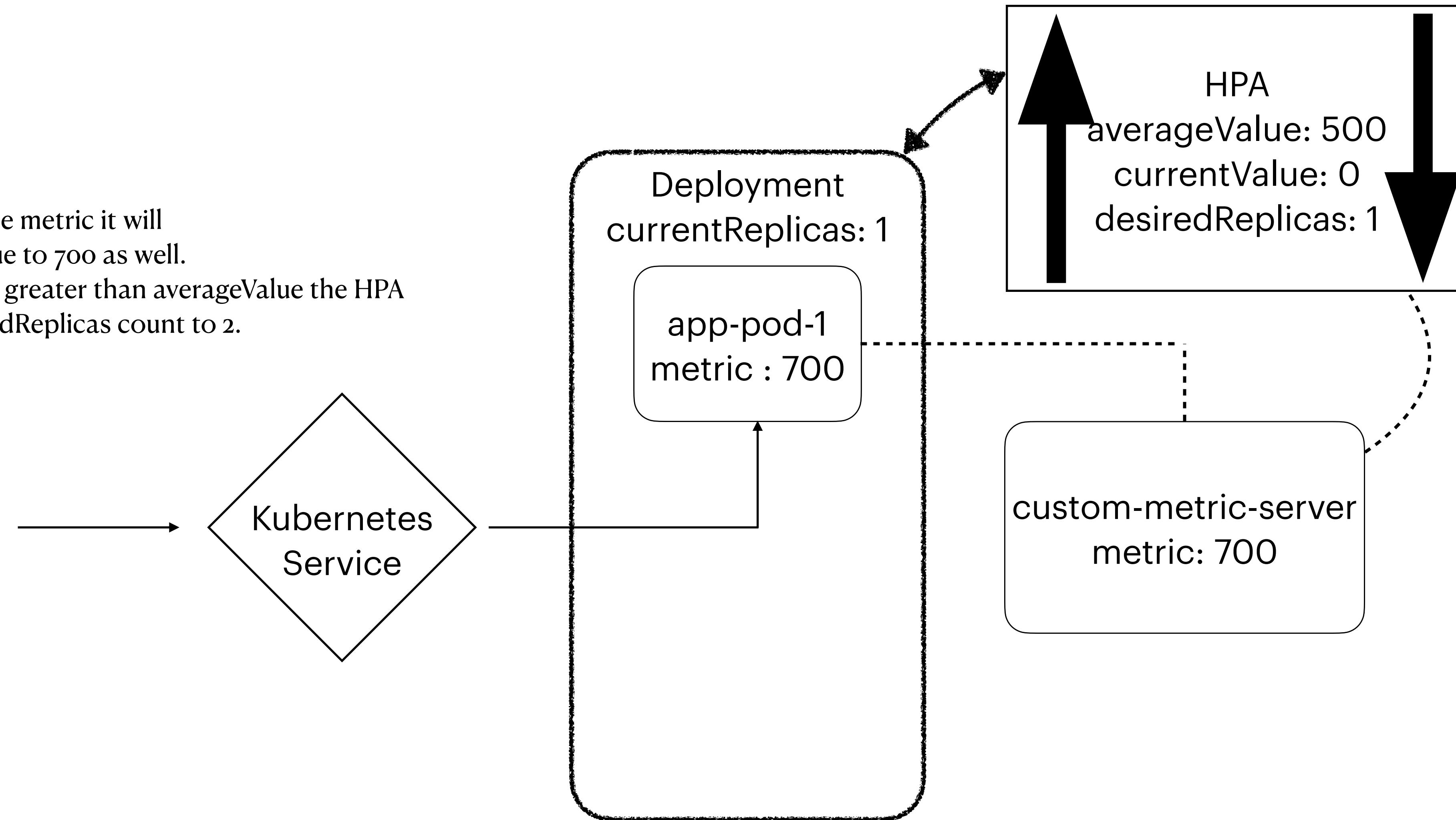


# Custom Metrics in Action



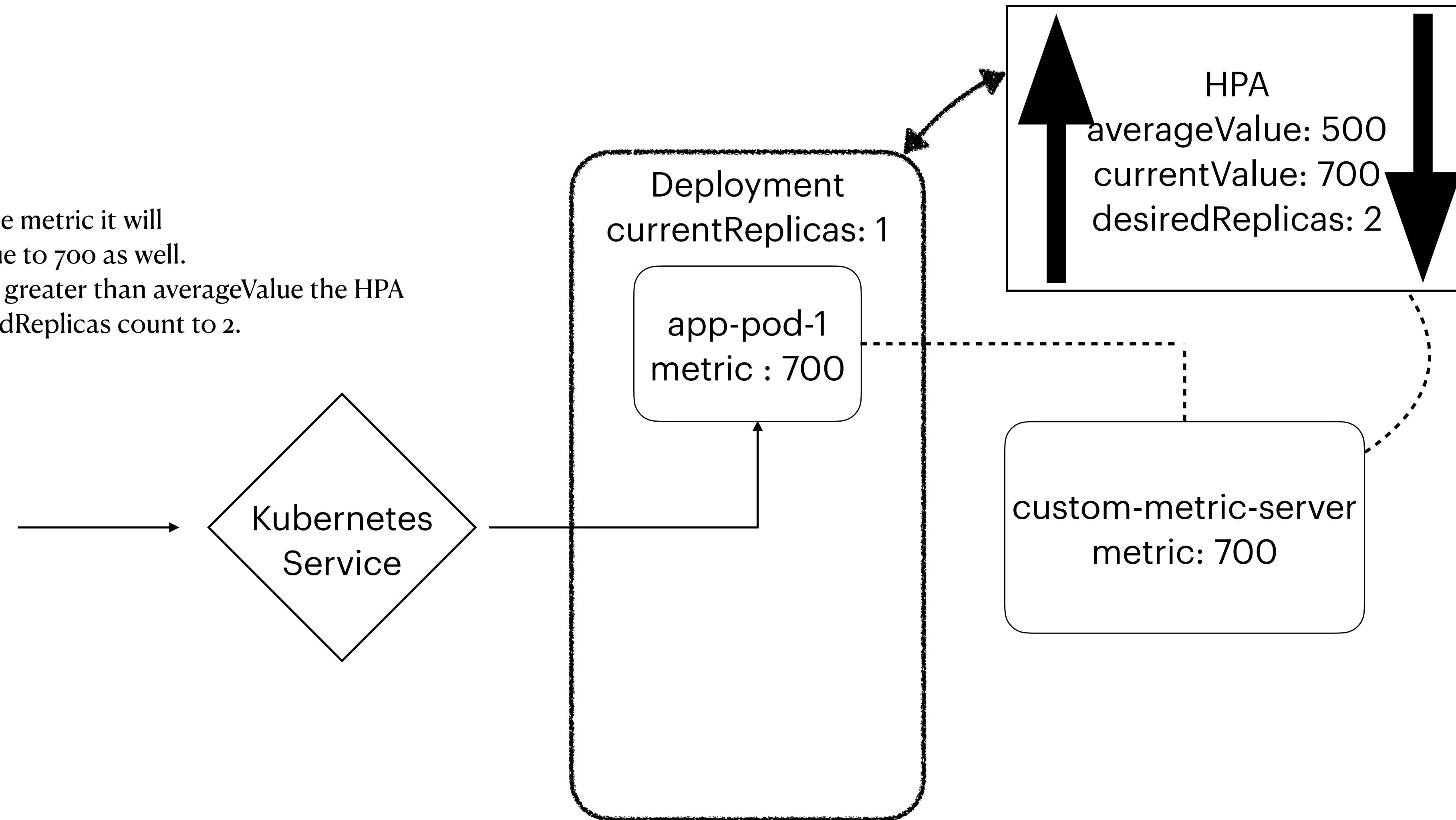
# Custom Metrics in Action

When HPA will poll the metric it will update its currentValue to 700 as well. Since the new value is greater than averageValue the HPA will increase its desiredReplicas count to 2.

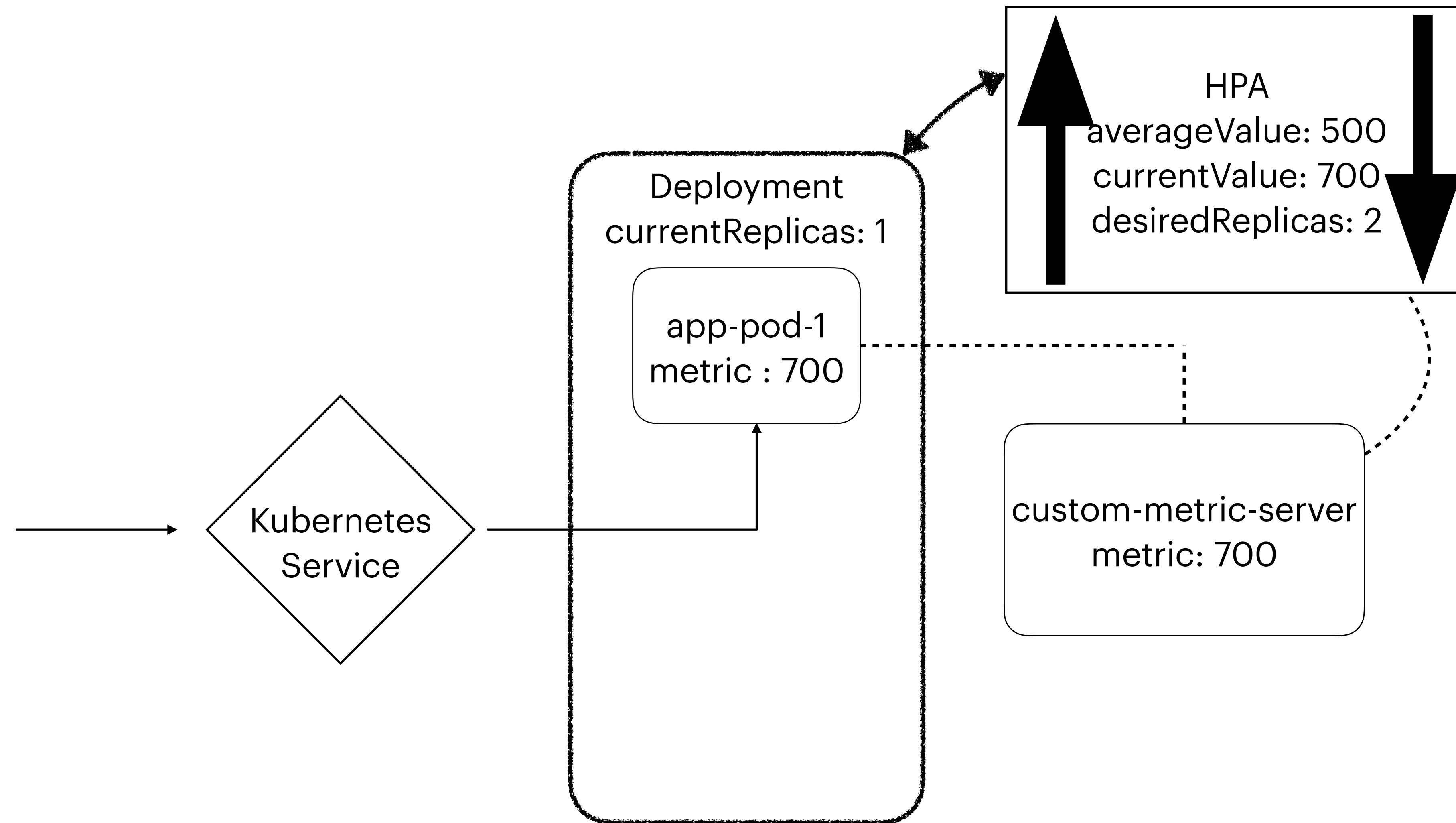


# Custom Metrics in Action

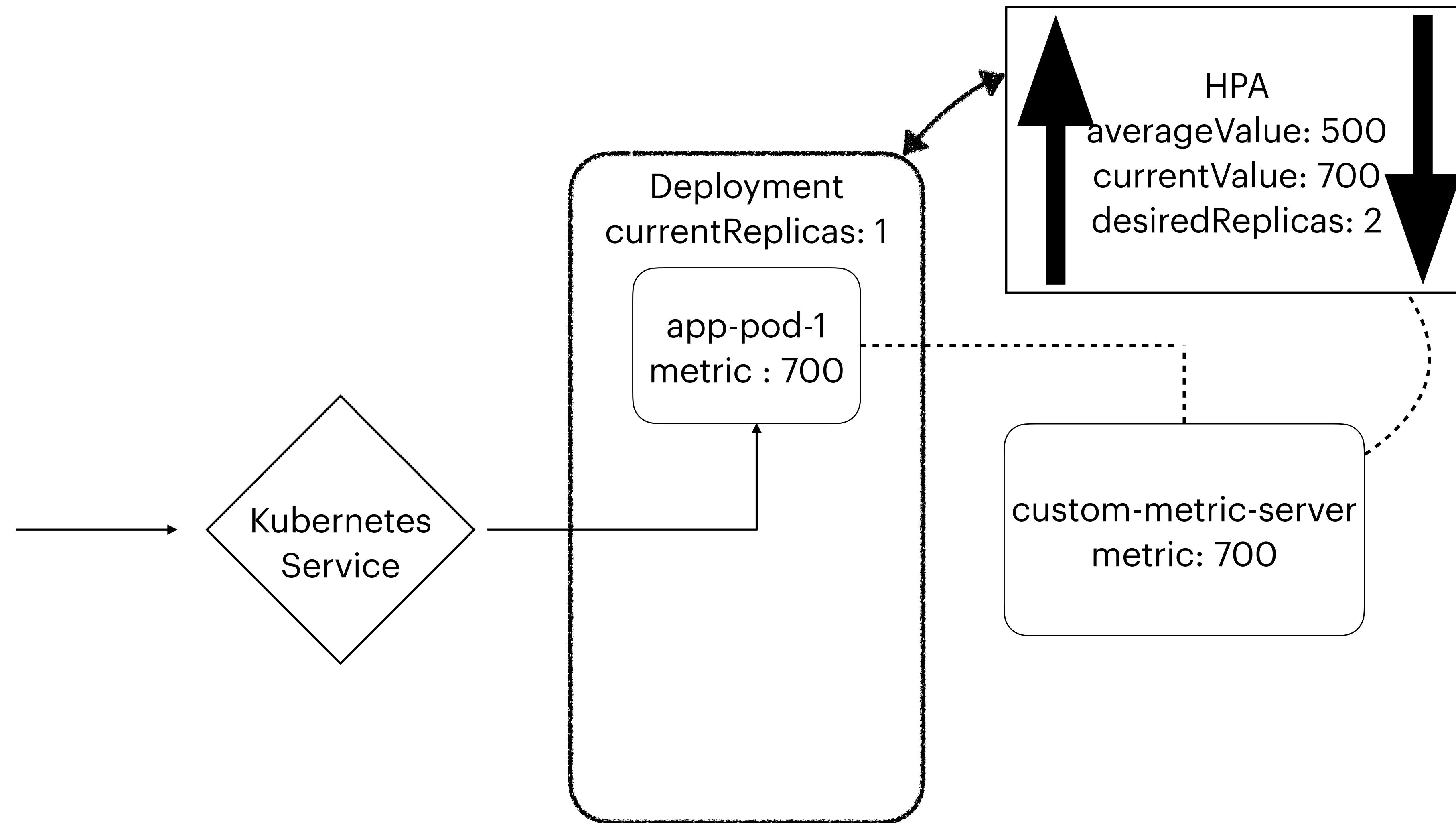
When HPA will poll the metric it will update its currentValue to 700 as well. Since the new value is greater than averageValue the HPA will increase its desiredReplicas count to 2.



# Custom Metrics in Action

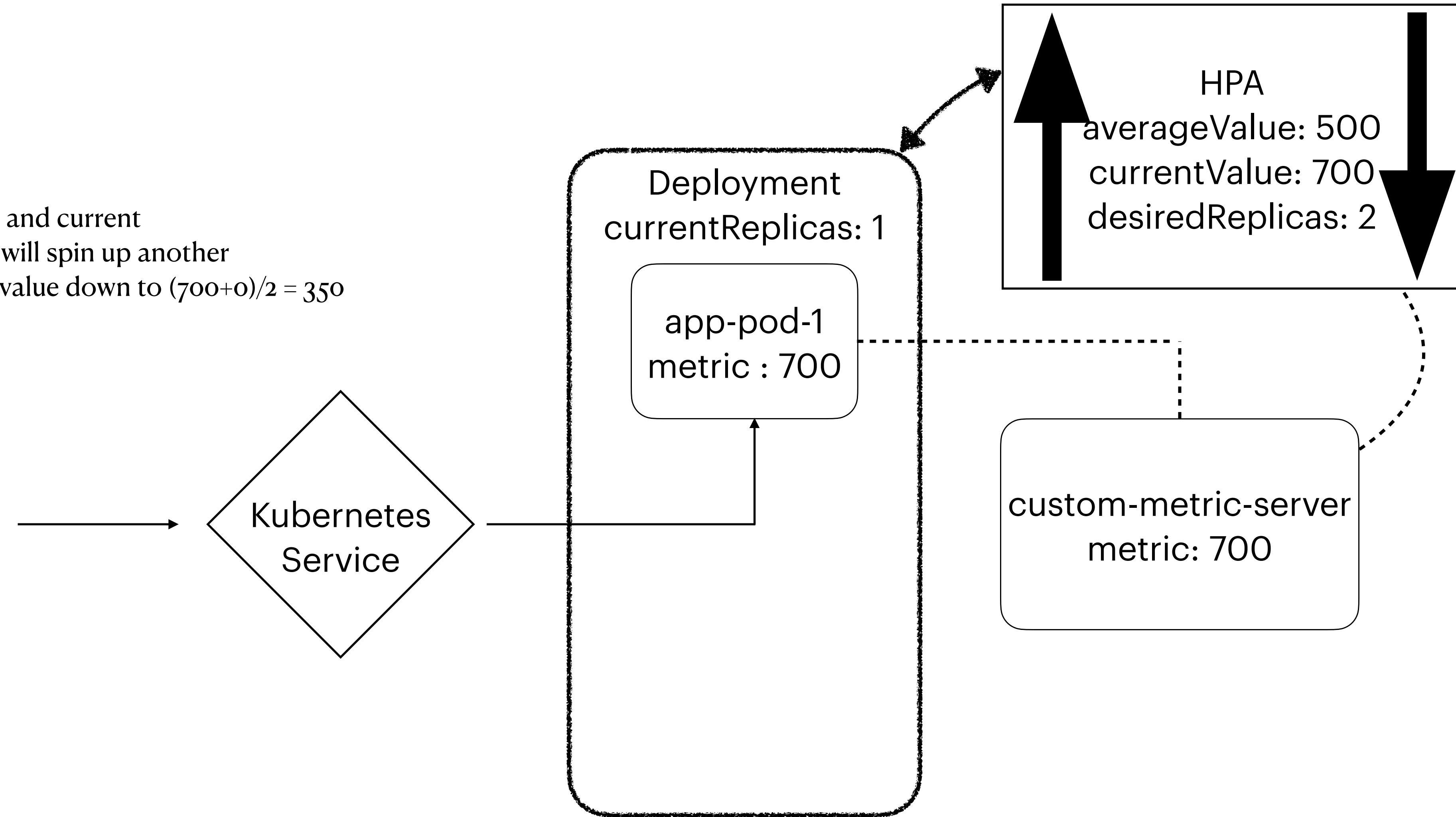


# Custom Metrics in Action



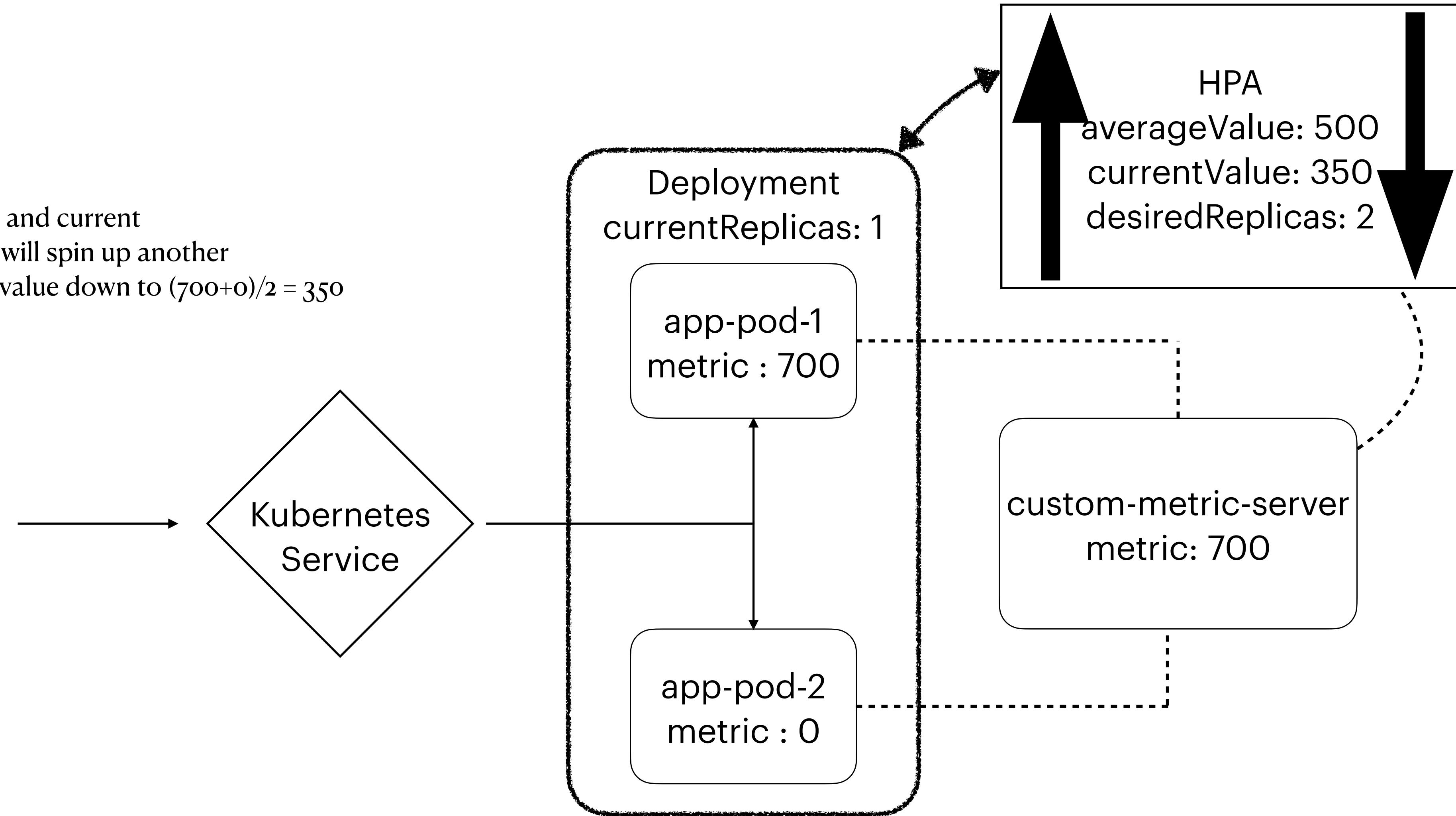
# Custom Metrics in Action

As HPA's desired state and current state do not match, it will spin up another pod to get its current value down to  $(700+0)/2 = 350$

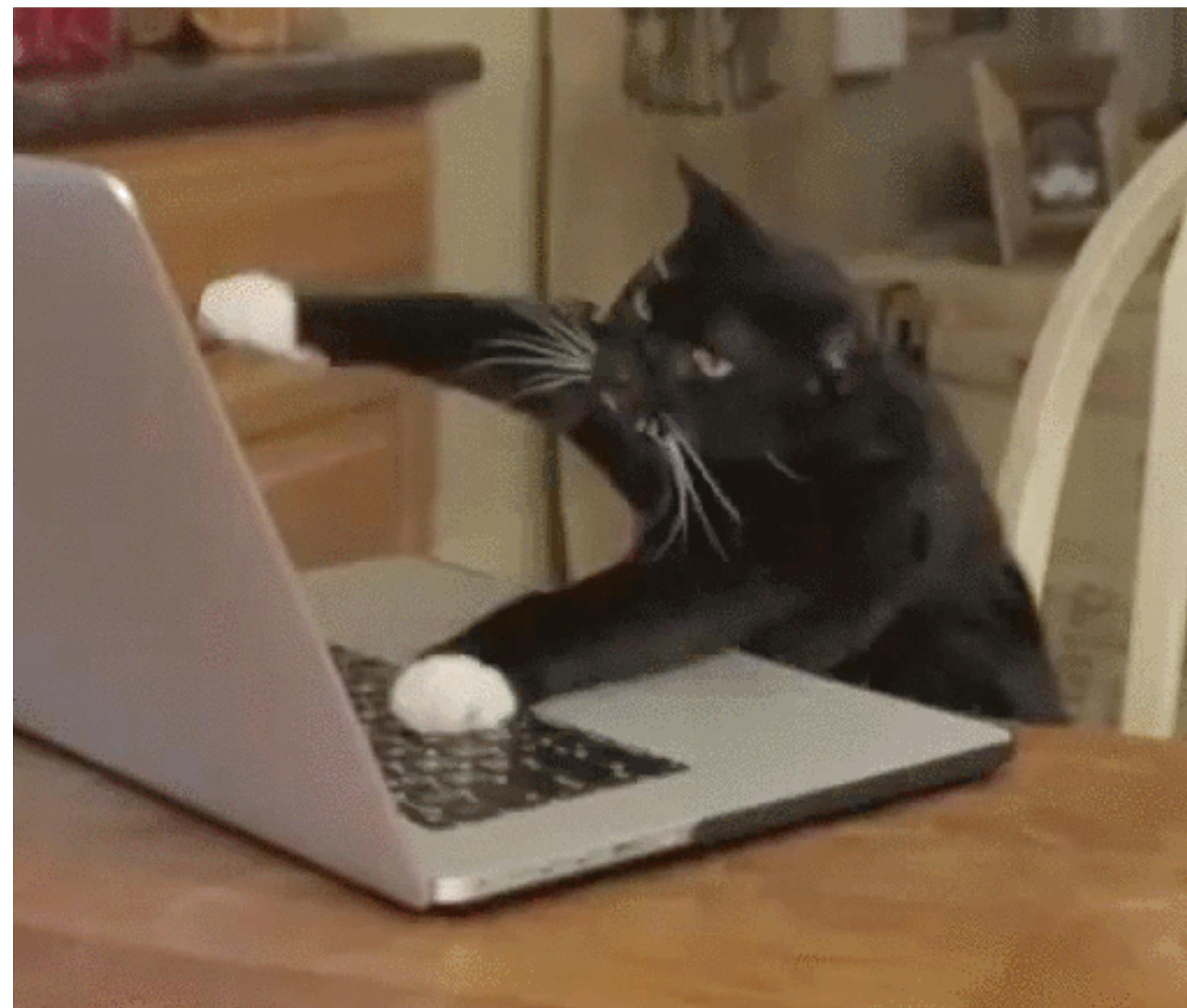


# Custom Metrics in Action

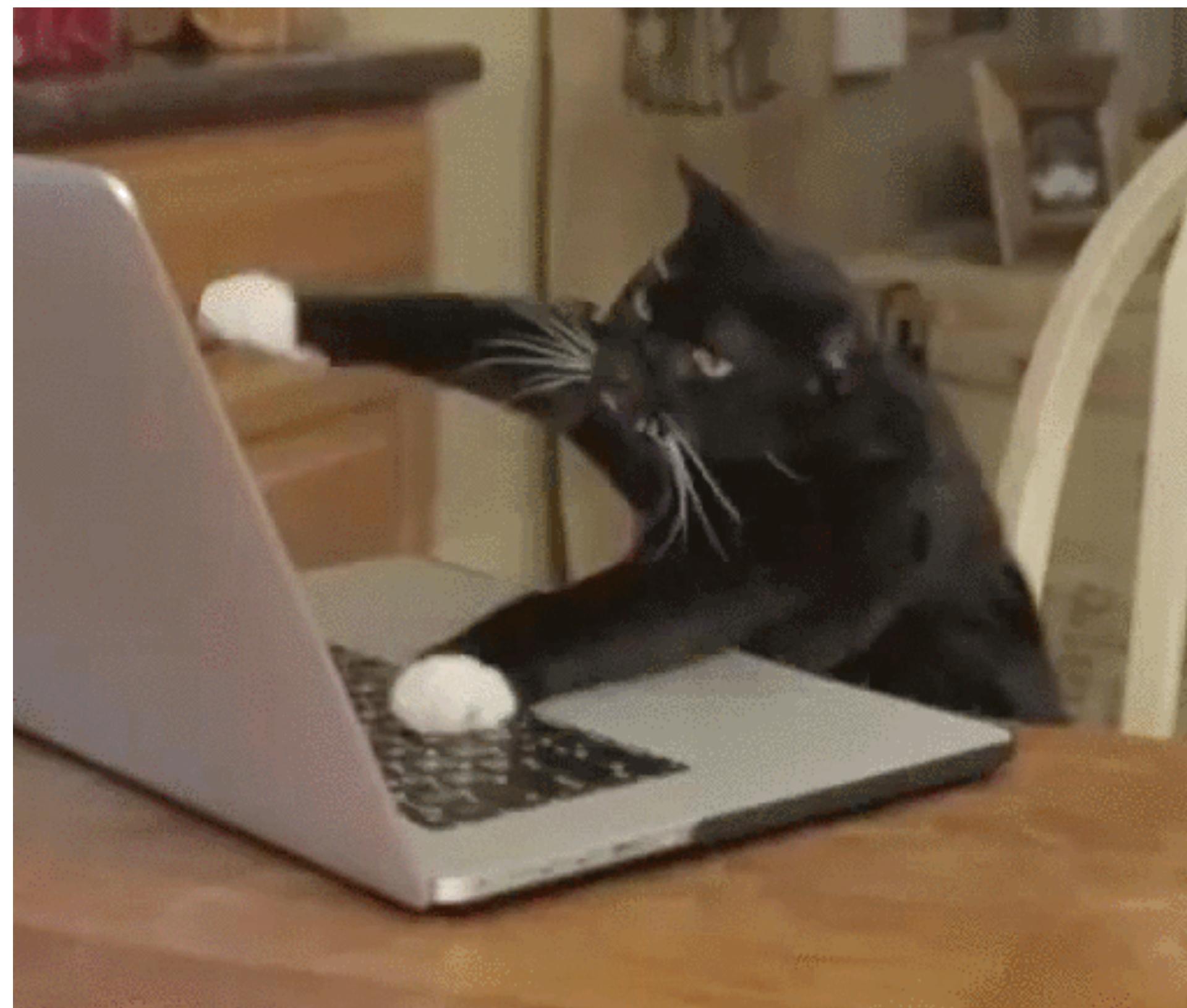
As HPA's desired state and current state do not match, it will spin up another pod to get its current value down to  $(700+0)/2 = 350$



# SHOW ME THE CODE



# SHOW ME THE CODE





# Source Code

<https://bit.ly/k8s-custom-metric-collector-adapter>





Thank You!