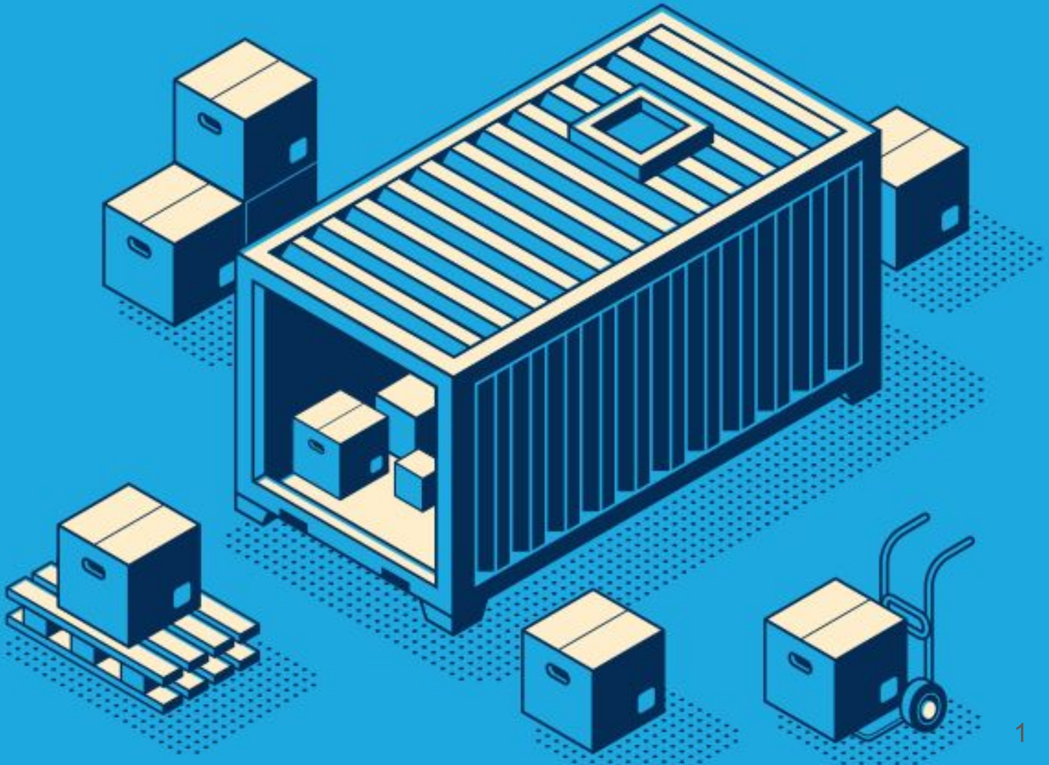
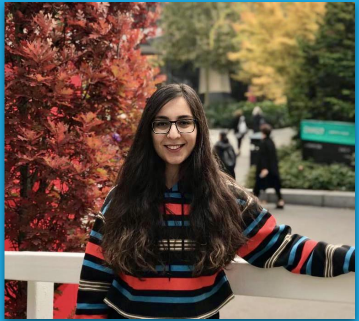
Kubernetes

Vertical Pod Autoscaler Operator (VPA)



The Team


Selen

Laxmi

Apoorva

Akshay

Sindhu

Mentors (Redhat Operate-First)

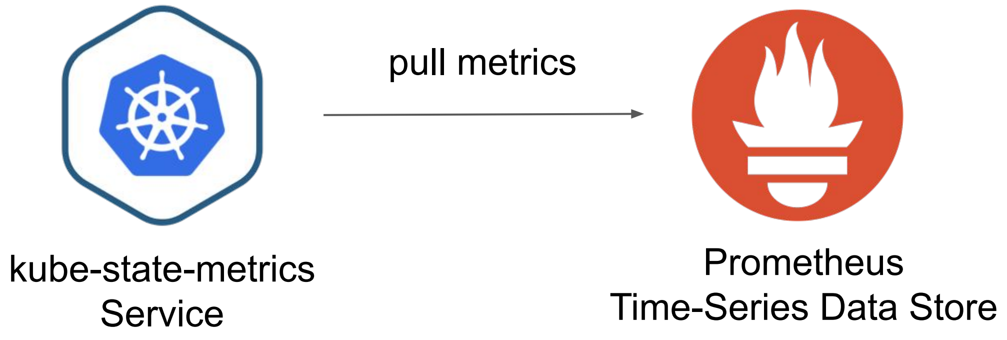
Humair Khan & Anand Sanmukhani

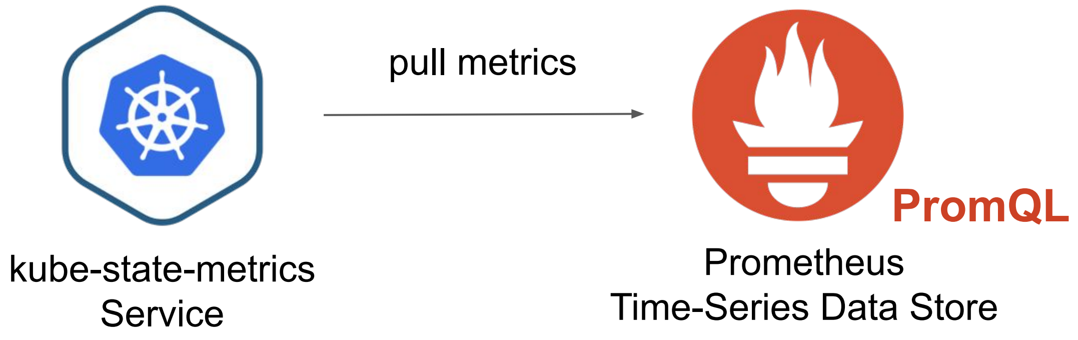
What we achieved in THIS sprint

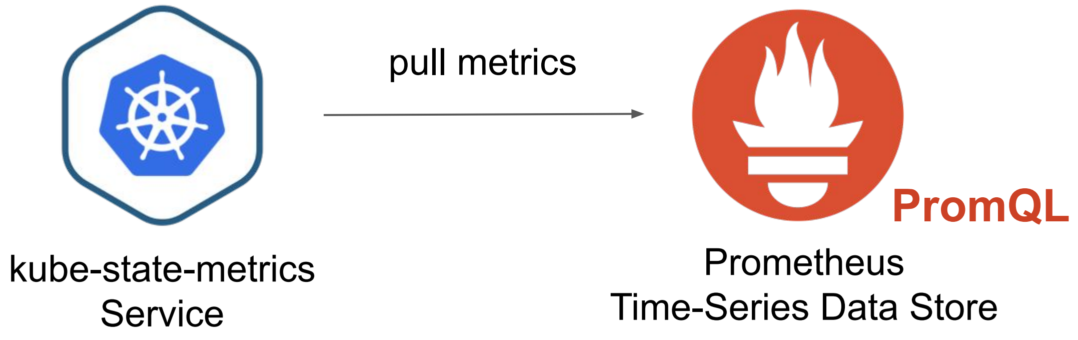
(Sprint Dallas)

- Applied VPA testing scenarios
- Observed VPA behaviour on Smaug & MOC clusters
- Created our own Grafana dashboard
- Visualized VPA resource consumption on this Grafana dashboard

kube-state-metrics
Service

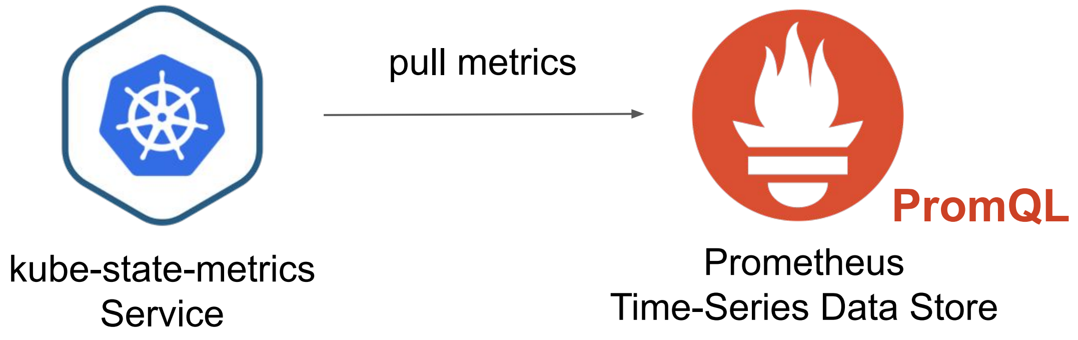

pull metrics
kube-state-metrics
Service
Prometheus
Time-Series Data Store


pull metrics
PromQL
kube-state-metrics
Service
Prometheus
Time-Series Data Store



binary arithmetic operators

- `+` (addition)
- `-` (subtraction)
- `*` (multiplication)

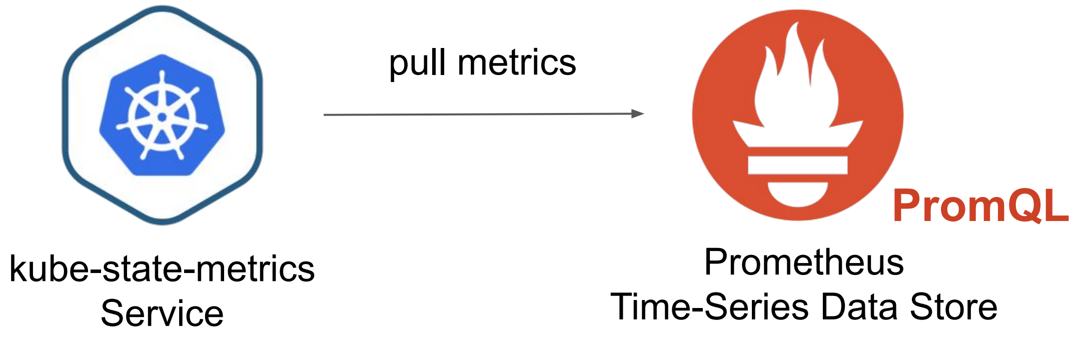


binary arithmetic operators

- `+` (addition)
- `-` (subtraction)
- `*` (multiplication)

binary comparison operators

- `==` (equal)
- `!=` (not-equal)
- `>` (greater-than)



binary arithmetic operators

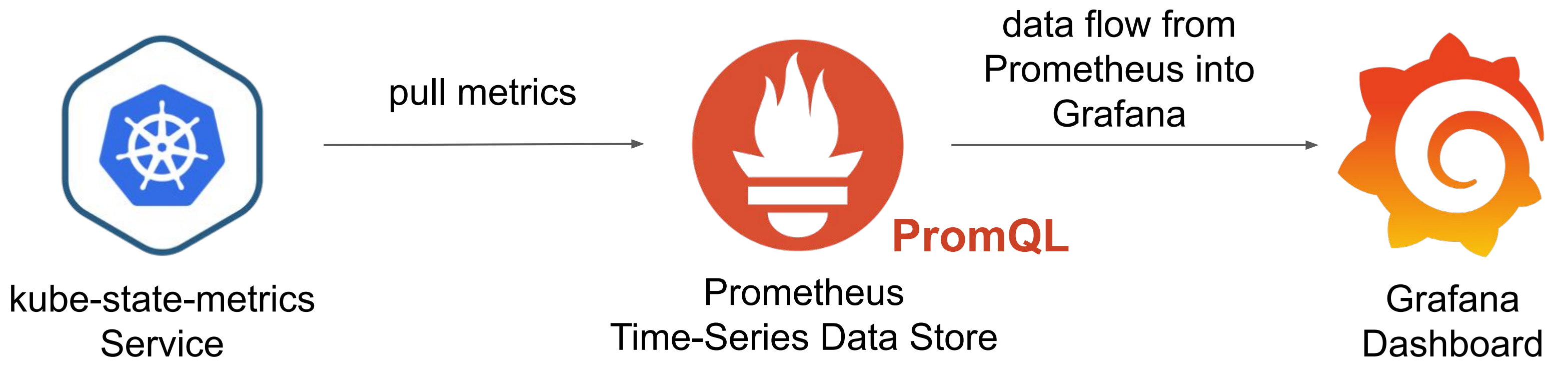
- `+` (addition)
- `-` (subtraction)
- `*` (multiplication)

binary comparison operators

- `==` (equal)
- `!=` (not-equal)
- `>` (greater-than)

aggregations

- `sum` (calculate sum over dimensions)
- `min` (select minimum over dimensions)
- `max` (select maximum over dimensions)
- `avg` (calculate the average over dimensions)


kube-state-metrics Service
pull metrics
Prometheus Time-Series Data Store
PromQL
data flow from Prometheus into Grafana
Grafana Dashboard

binary arithmetic operators

- `+` (addition)
- `-` (subtraction)
- `*` (multiplication)

binary comparison operators

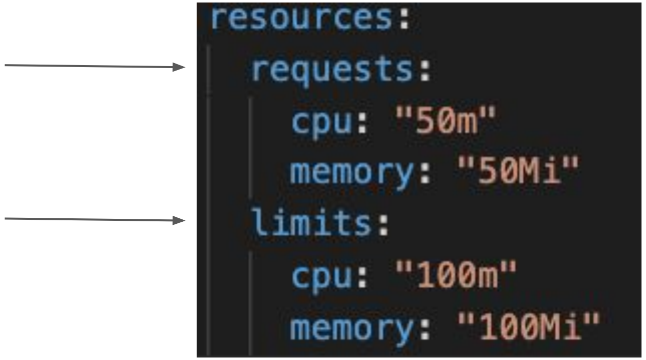
- `==` (equal)
- `!=` (not-equal)
- `>` (greater-than)

aggregations

- `sum` (calculate sum over dimensions)
- `min` (select minimum over dimensions)
- `max` (select maximum over dimensions)
- `avg` (calculate the average over dimensions)

VPA

GOAL: Find an automatic way to configure Pod's resource requests



```
resources:
  requests:
    cpu: "50m"
    memory: "50Mi"
  limits:
    cpu: "100m"
    memory: "100Mi"
```

VPA Scenarios

How does VPA react during...

- **New Pod** initialization
- **Under-utilization** of CPU/Memory resources
- **Over-utilization** of CPU/Memory resources
- Auto-Update policy

VPA scenario 1

How does resource recommendations change when a new pod is configured with VPA?

VPA Scenario 2 - Over utilization

As discussed earlier, **Limits** – Safety valve

- What if a container uses beyond provided CPU limit?
- What if a container is using more memory than the limit?

VPA Scenario 2 - Over utilization

When a container uses more CPU time than provided CPU limit? - **It throttles**

What if a container is using more memory than given memory limit? - **The whole pod gets killed (OOM Event)**

VPA Scenario 2 - Over utilization

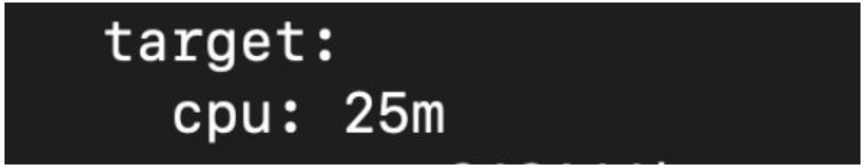
VPA's Goal - To reduce resource wastage while minimizing the risk of performance degradation due to

- CPU throttling
- Out Of Memory kills.

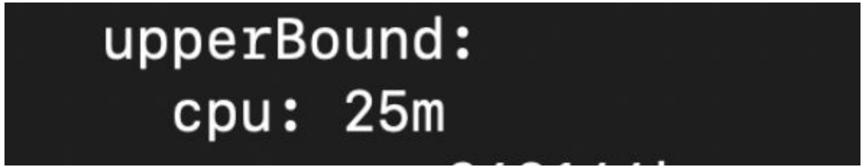
VPA Scenario 2 - Over utilization

Upper Bound: The maximum recommended resource estimation for the Container.

What if we cross this bound?



```
  target:
    cpu: 25m
```



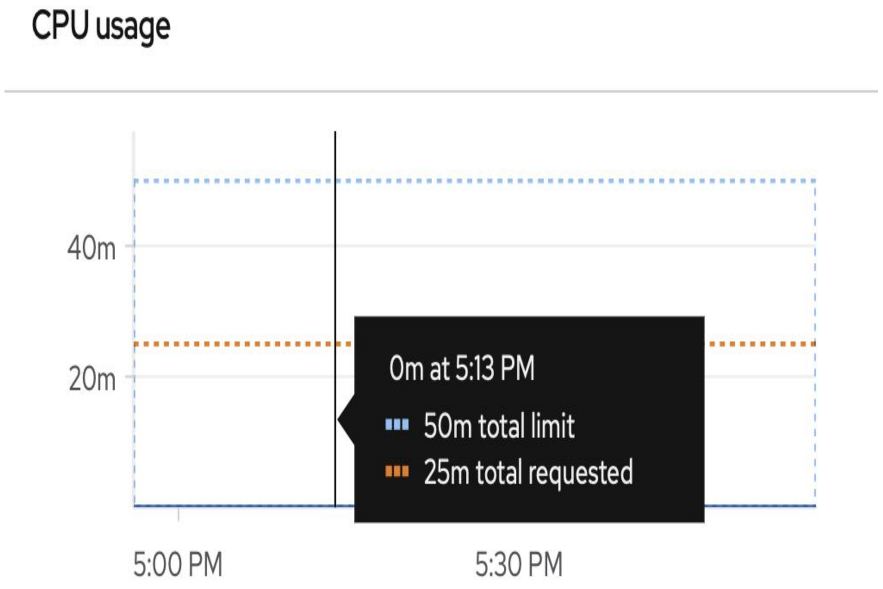
```
  upperBound:
    cpu: 25m
```

VPA Scenario 2 - Over utilization

Example:

- Before: CPU
  - Requests: 25m
  - Limits: 50m

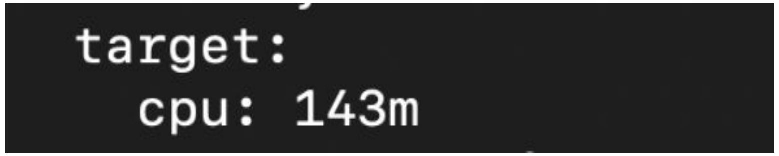
Requested a CPU usage of 120mCores for a duration of 1800 sec.



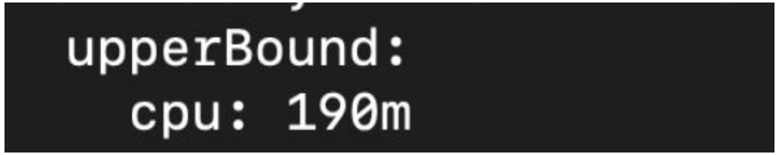
VPA Scenario 2 - Over utilization

After: CPU

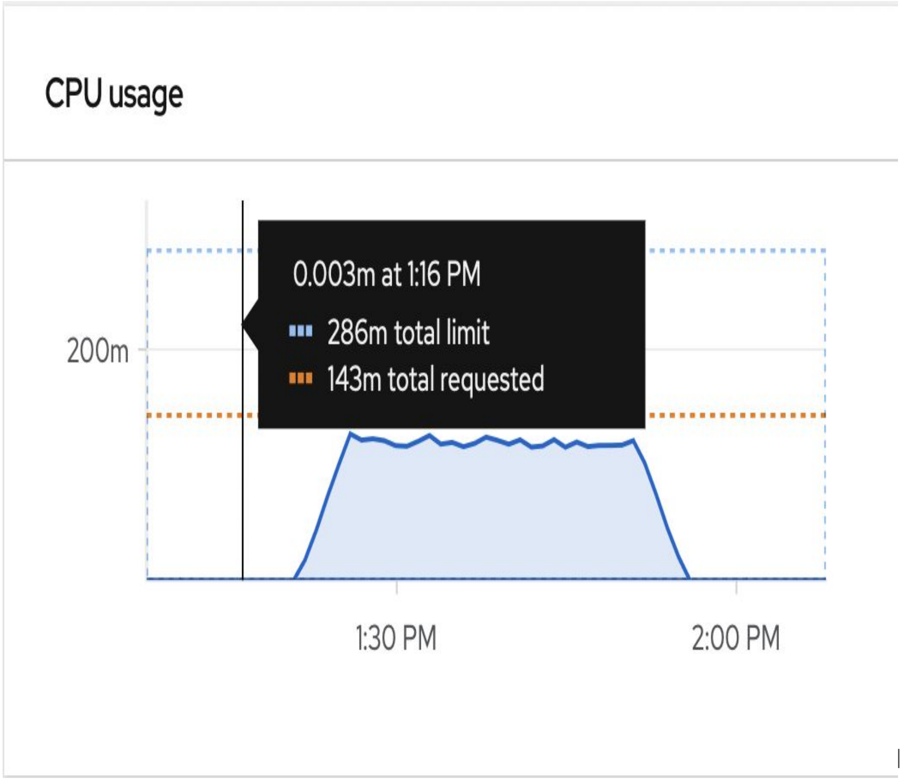
- Requests: 143m
- Limits: 286m



```
target:
  cpu: 143m
```



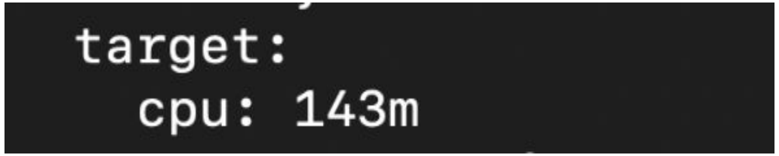
```
upperBound:
  cpu: 190m
```



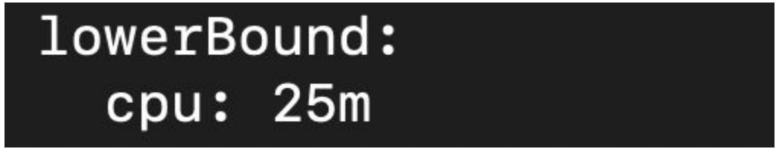
VPA Scenario 3 - Under utilization

Requested a CPU usage of 20mCores for a duration of 1800 sec.

Before: CPU

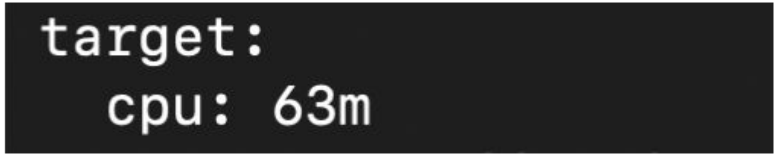


```
  target:
    cpu: 143m
```

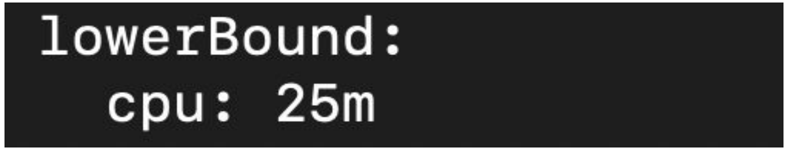


```
lowerBound:
  cpu: 25m
```

After: CPU

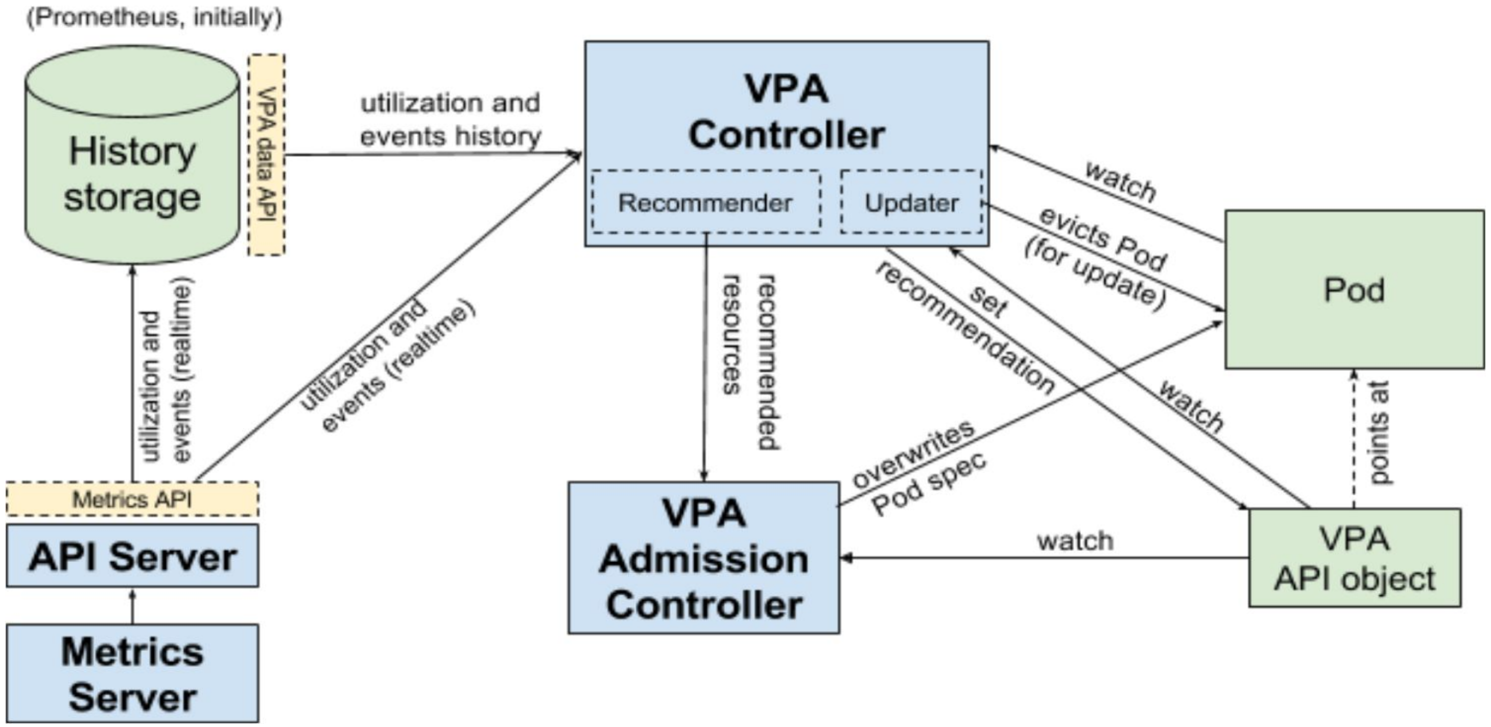


```
target:
  cpu: 63m
```

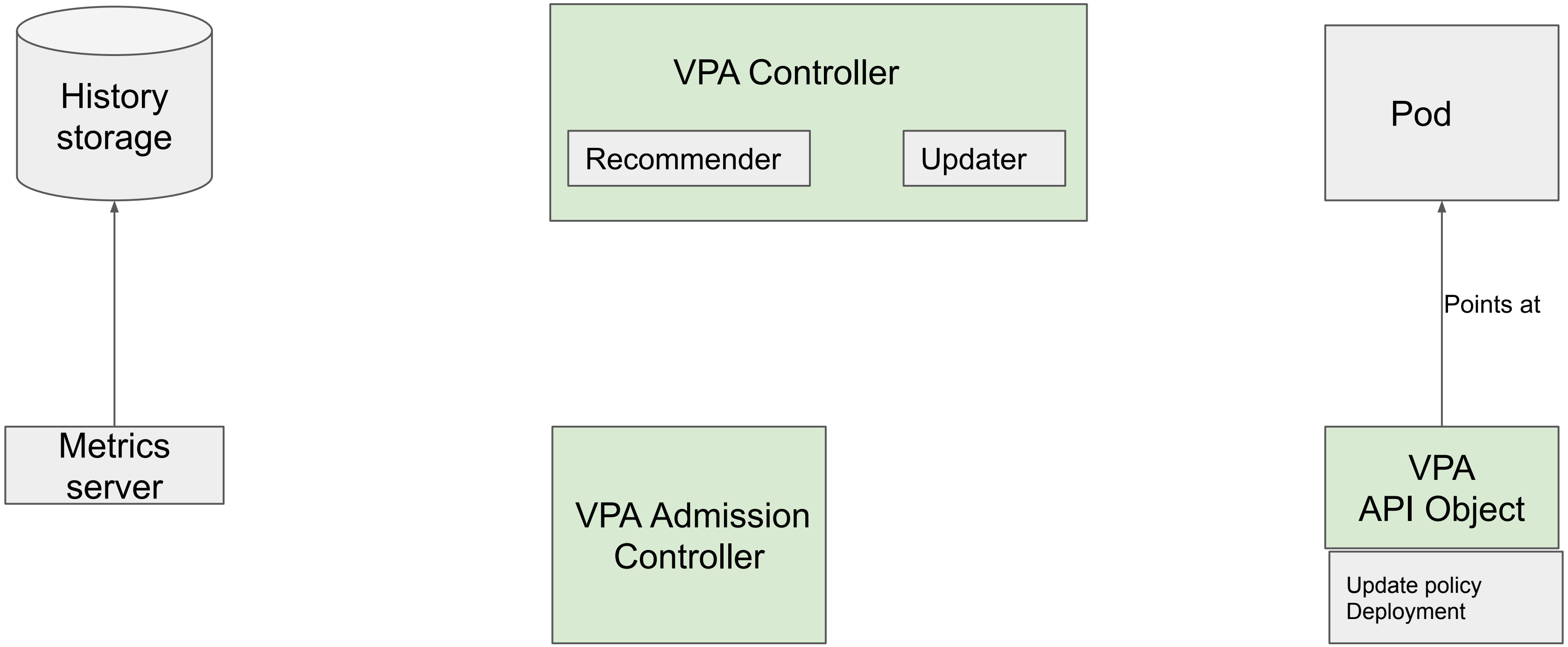


```
lowerBound:
  cpu: 25m
```

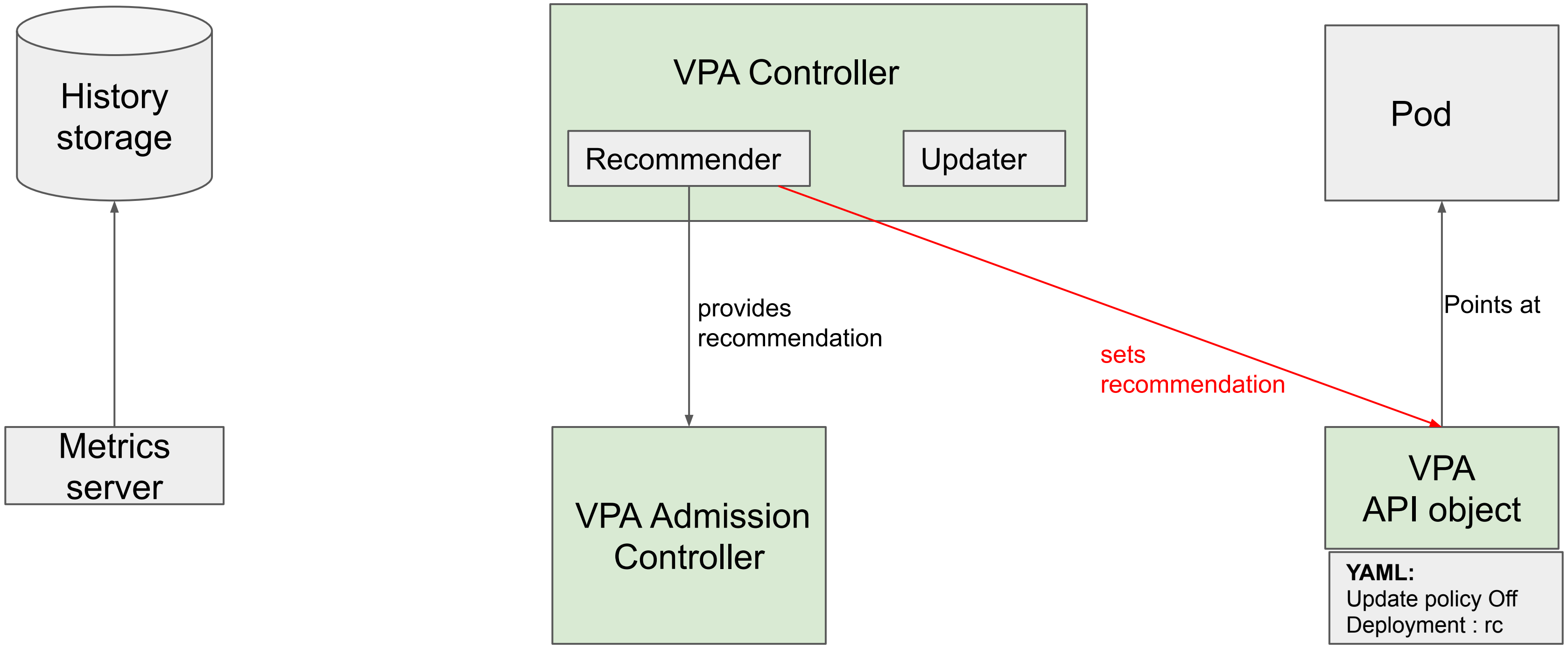
VPA Architecture



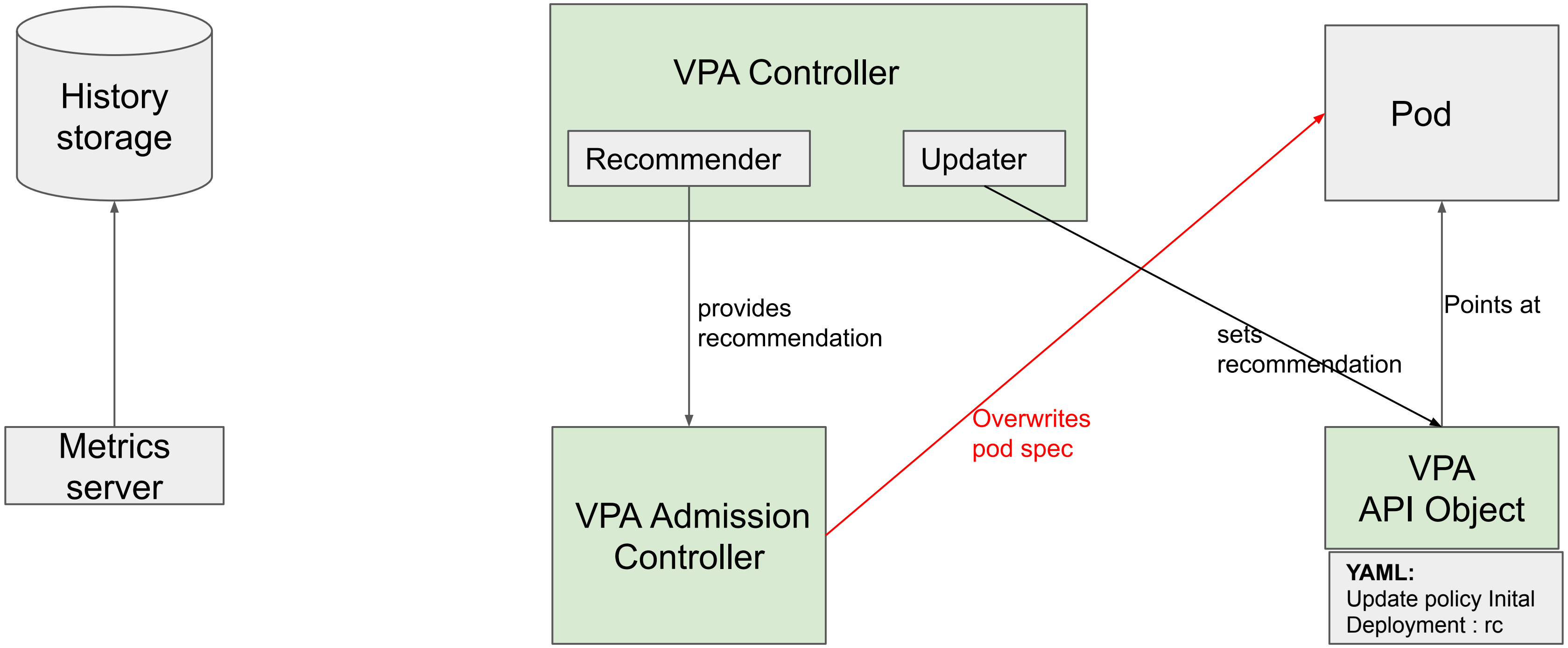
VPA - 3 components



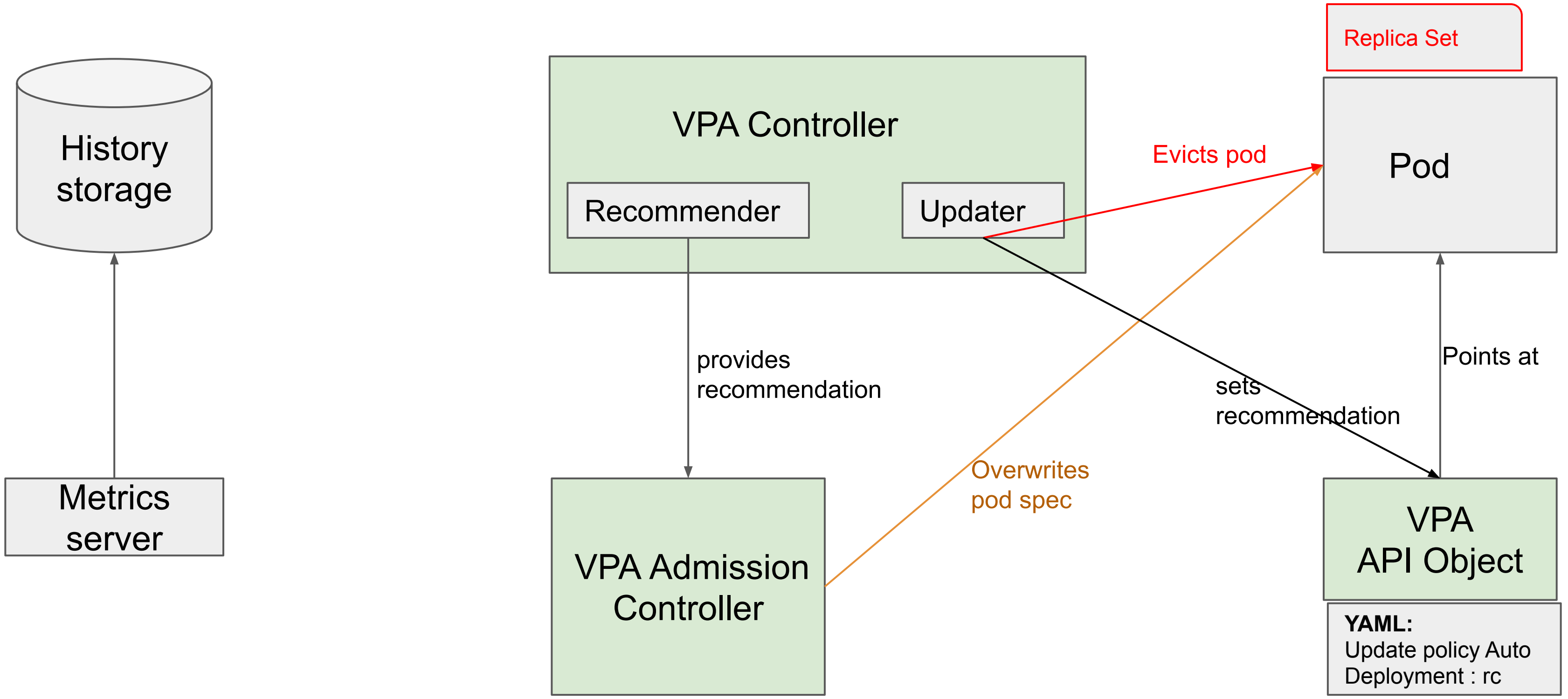
VPA recommendation mode



VPA Inital mode



VPA Auto mode



YAML:
Update policy Auto
Deployment : rc

VPA Demo

VPA - What we learnt...   :)

- VPA Recommendations do not drastically spike always
    - It depends on container resource usage history
- VPA Auto update does not act always
    - resource usage should go beyond **upper** and **lower** bounds

VPA Limitations

VPA Limitations

VPA may cause **down time**

VPA Limitations

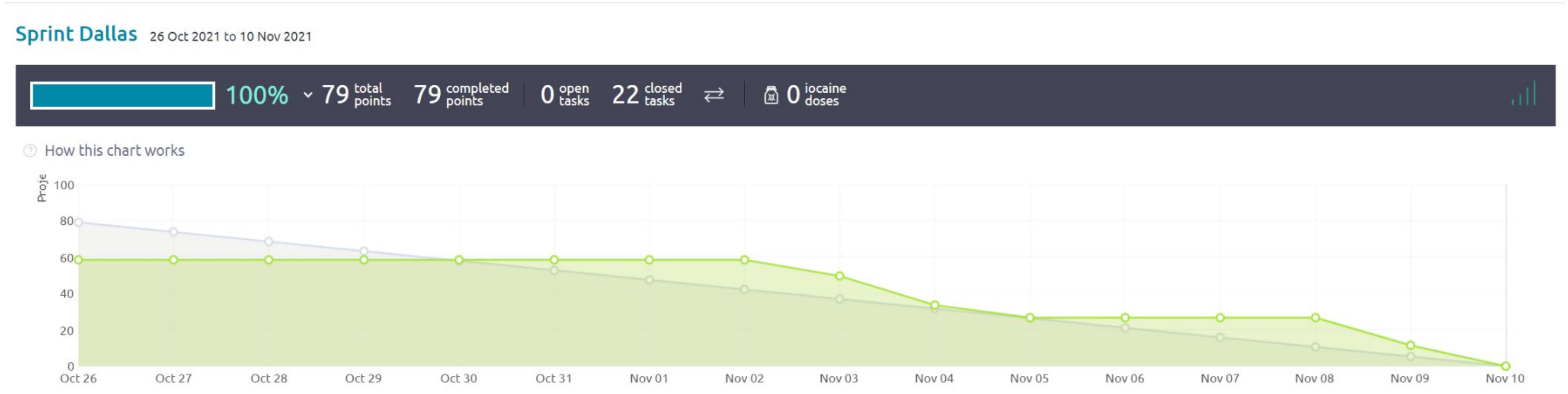
Vertical Pod Autoscaler should **NOT** be used with the **Horizontal Pod Autoscaler (HPA)**

for the same resources eg. CPU & Memory.

VPA Limitations

VPA performance has not been tested in **large clusters**

Burndown chart



Sprint Dallas 26 Oct 2021 to 10 Nov 2021

100% 79 total points 79 completed points 0 open tasks 22 closed tasks 0 iocaine doses

What we plan for Next sprint

(Sprint edinburgh)

- Test VPA with different usage patterns - Burst workloads vs Constant workload
- Based on best usage pattern, suggest a good candidate to apply VPA
- Deploy VPA on that candidate in RedHat's Production Cluster