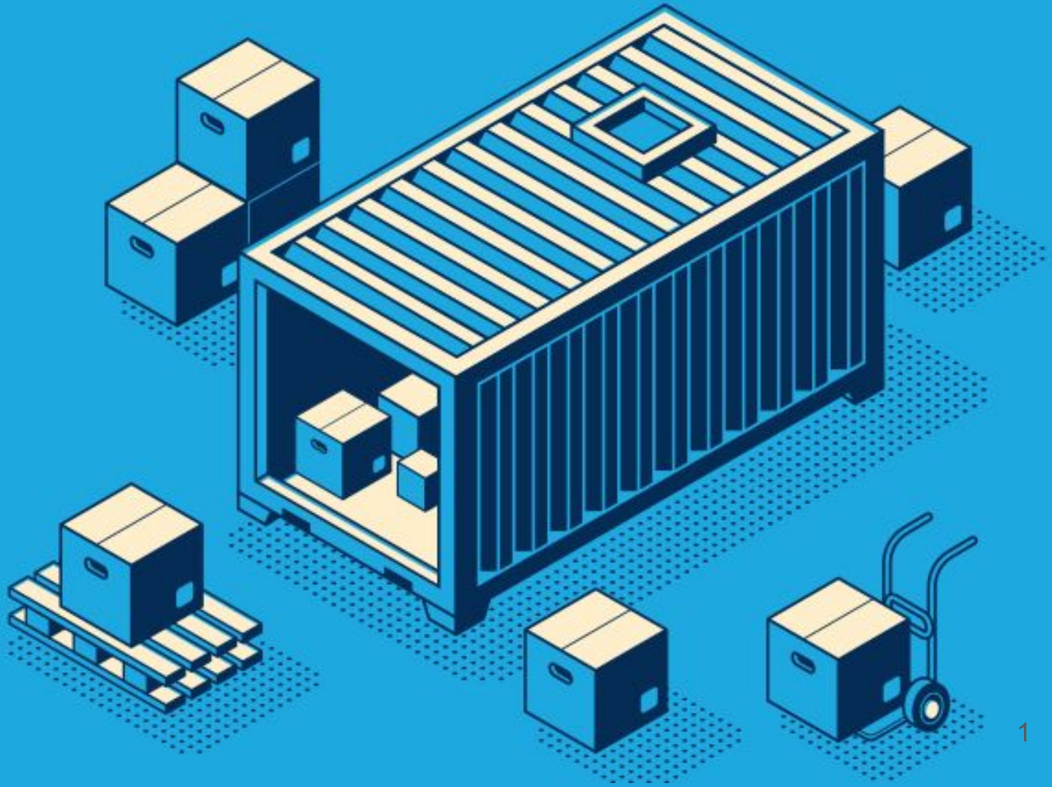
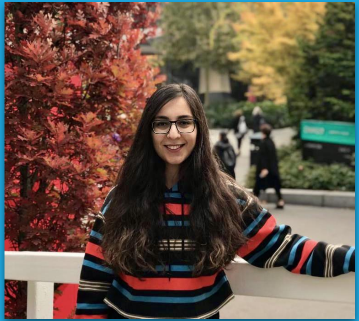
Kubernetes

Vertical Pod Autoscaler Operator



The Team



Selen

Laxmi

Apoorva

Akshay

Sindhu

Mentors (Redhat)

Humair Khan & Anand Sanmukhani

Kubernetes

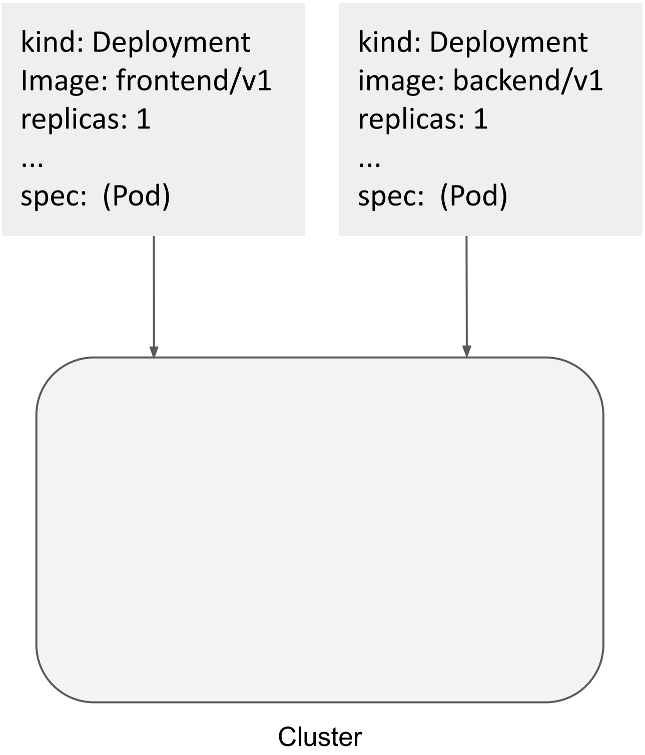
An Orchestration tool for automating deployment, scaling and management of containerized applications.

Operators

Operators are software **extensions to Kubernetes** that make use of custom resources to manage applications and their components.

Without Operator

With Operator



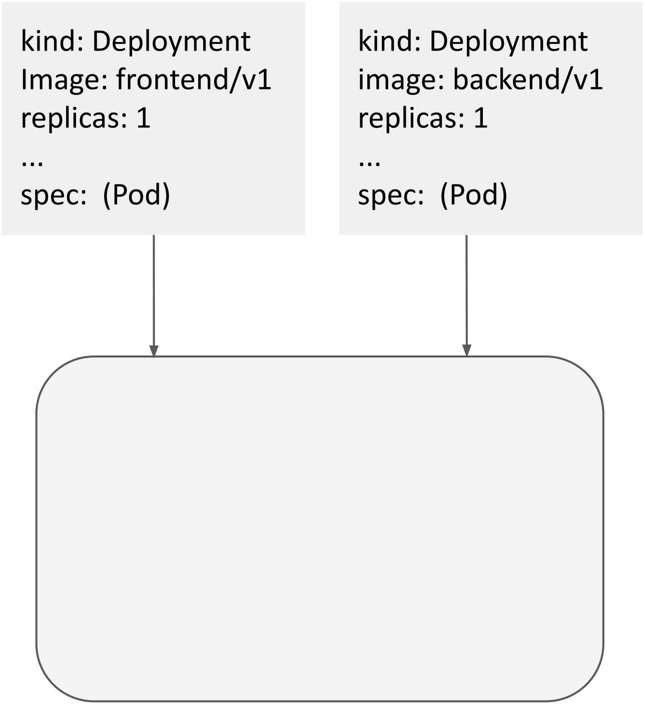
```
kind: Deployment
Image: frontend/v1
replicas: 1
...
spec:  (Pod)
```

```
kind: Deployment
image: backend/v1
replicas: 1
...
spec:  (Pod)
```

Cluster

Without Operator

With Operator

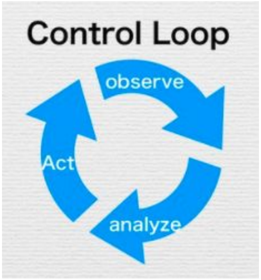


kind: Deployment
Image: frontend/v1
replicas: 1
...
spec: (Pod)

kind: Deployment
image: backend/v1
replicas: 1
...
spec: (Pod)

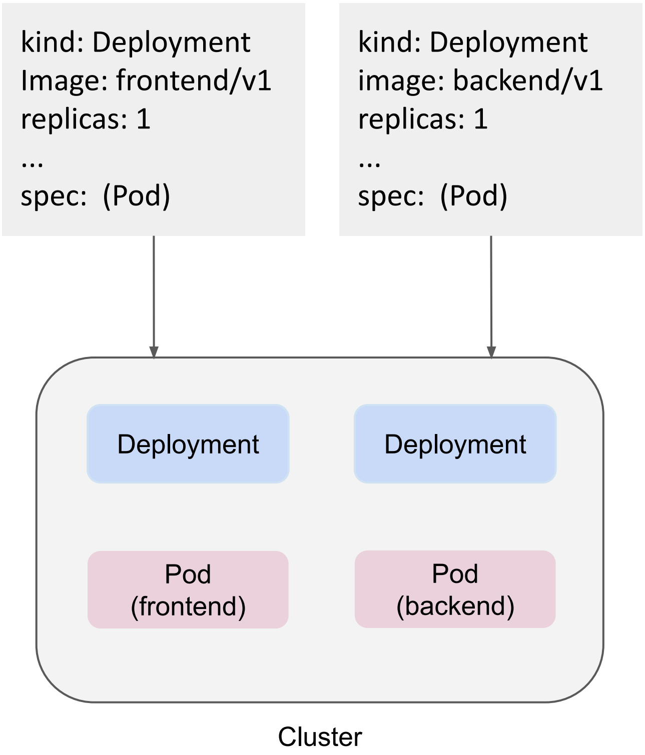
Cluster

Controller



Without Operator

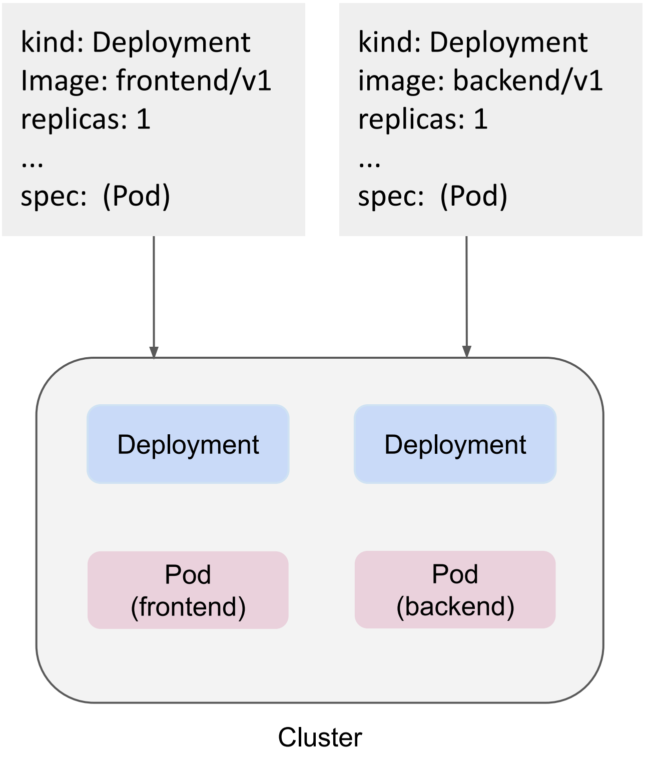
With Operator



```
kind: Deployment
Image: frontend/v1
replicas: 1
...
spec:  (Pod)
```

```
kind: Deployment
image: backend/v1
replicas: 1
...
spec:  (Pod)
```

Without Operator

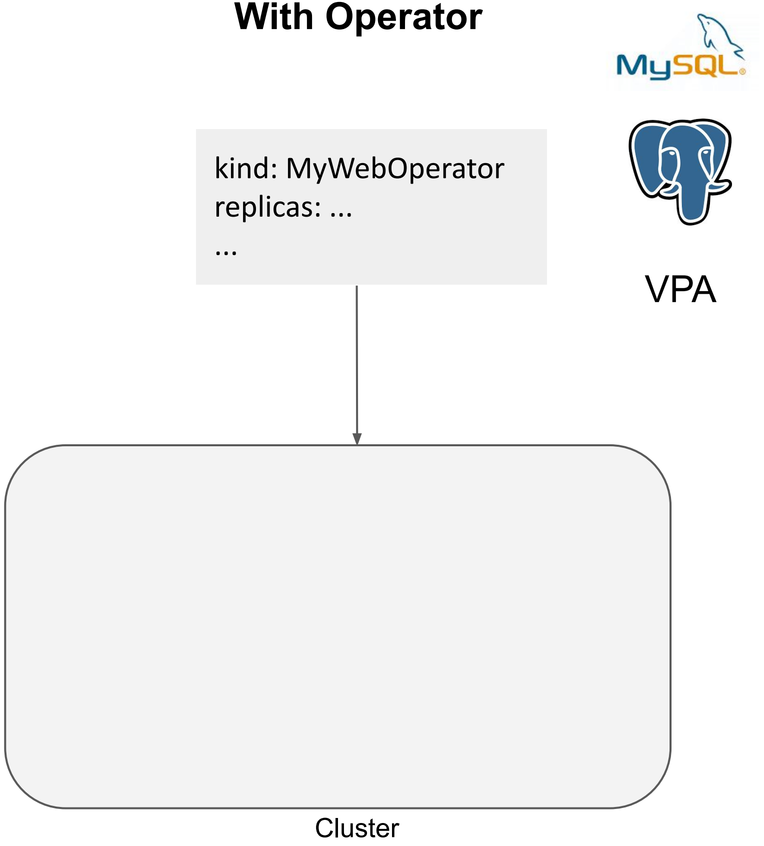


```
kind: Deployment
Image: frontend/v1
replicas: 1
...
spec:  (Pod)
```

```
kind: Deployment
image: backend/v1
replicas: 1
...
spec:  (Pod)
```

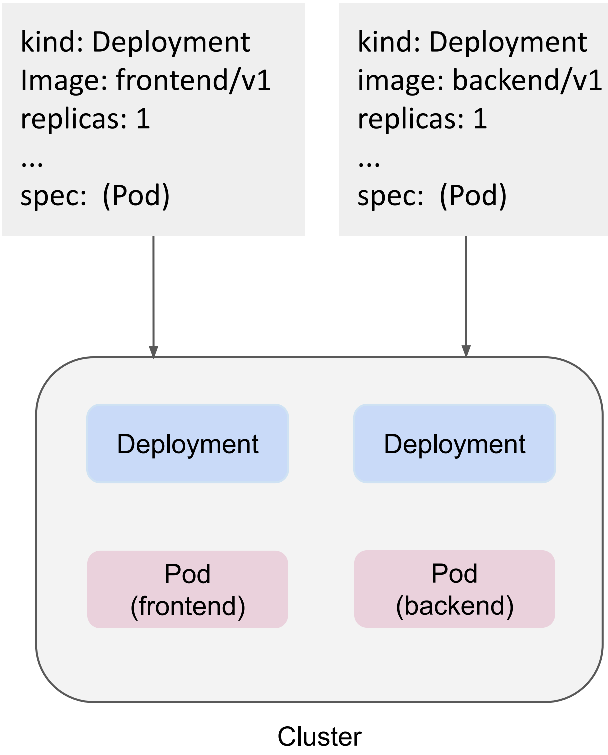
Native and
Custom
Resources

With Operator



```
kind: MyWebOperator
replicas: ...
...
```

Without Operator



```
kind: Deployment
Image: frontend/v1
replicas: 1
...
spec:  (Pod)
```

```
kind: Deployment
image: backend/v1
replicas: 1
...
spec:  (Pod)
```

Deployment

Deployment

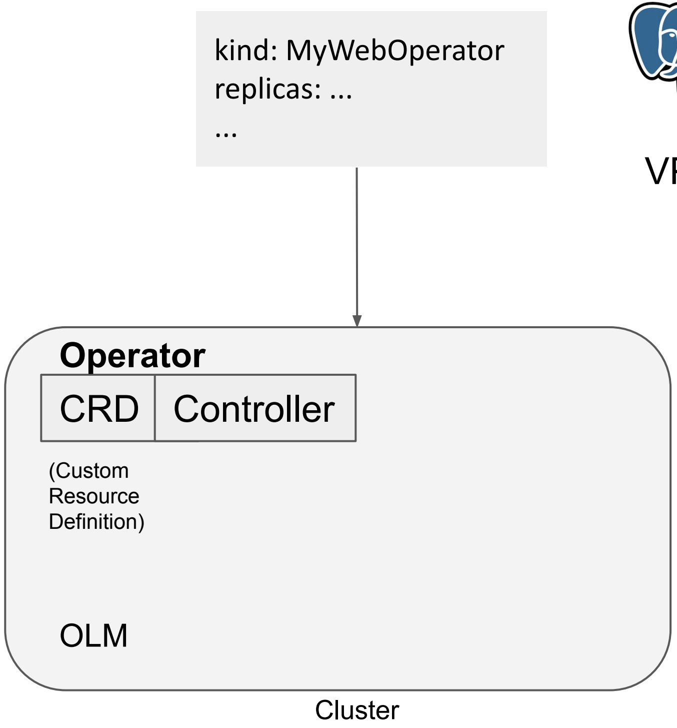
Pod
(frontend)

Pod
(backend)

Cluster

With Operator

Native and
Custom
Resources



```
kind: MyWebOperator
replicas: ...
...
```

Operator

CRD | Controller

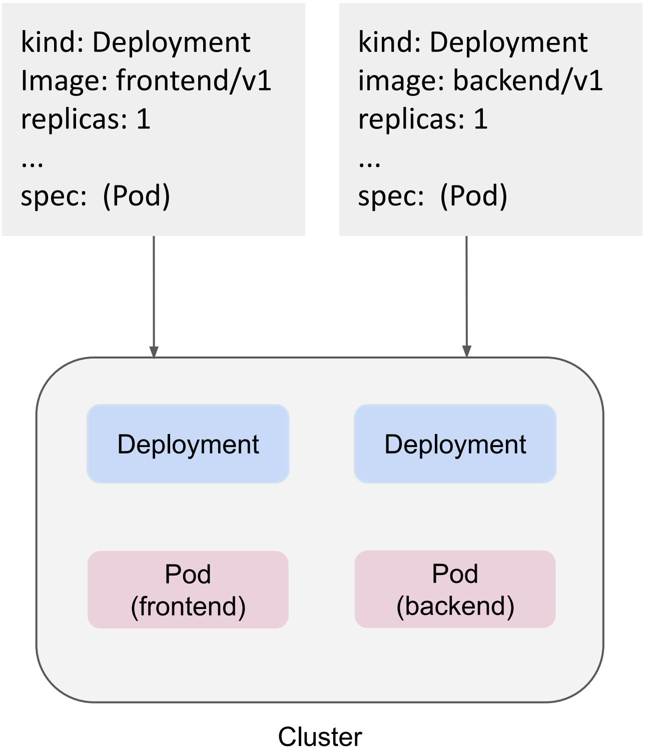
(Custom
Resource
Definition)

OLM

Cluster

VPA

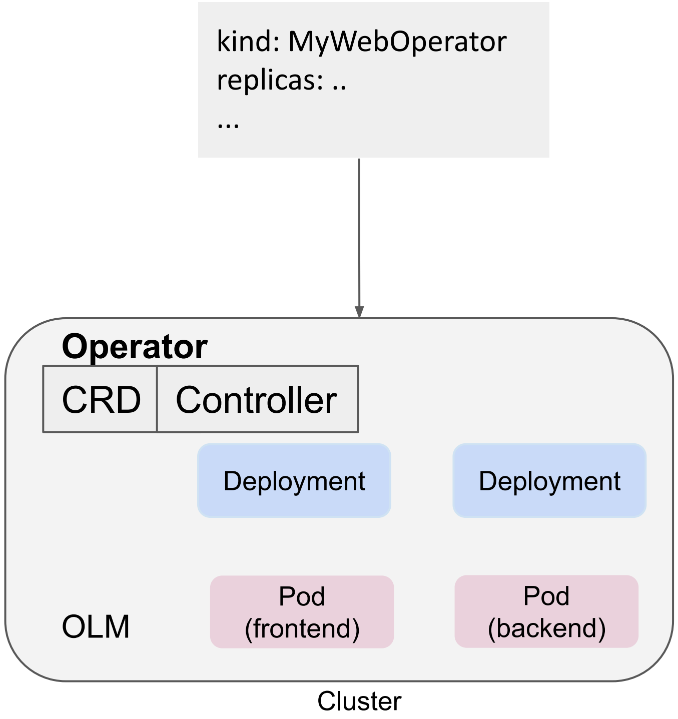
Without Operator



```
kind: Deployment
Image: frontend/v1
replicas: 1
...
spec:  (Pod)
```

```
kind: Deployment
image: backend/v1
replicas: 1
...
spec:  (Pod)
```

With Operator



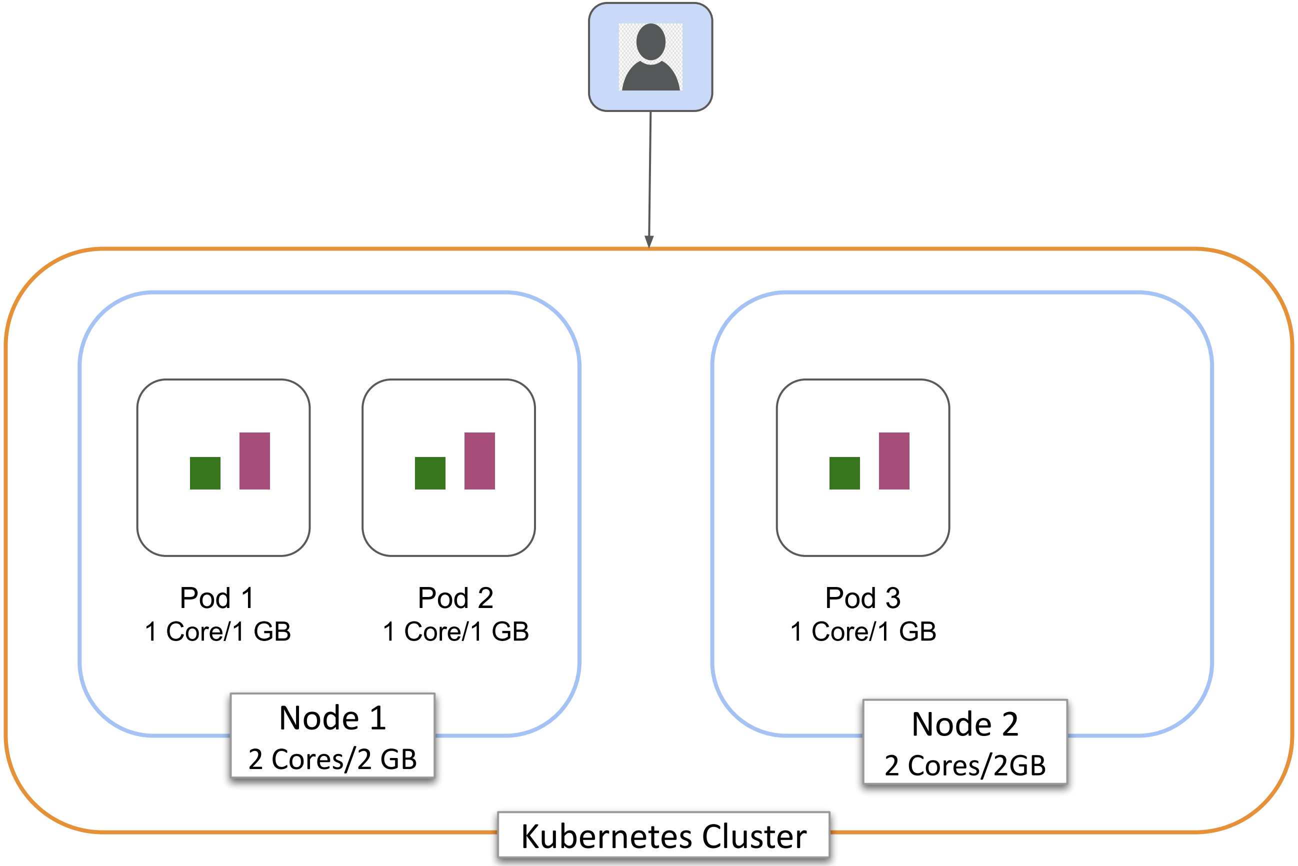
```
kind: MyWebOperator
replicas: ..
...
```

Scaling

- Horizontal Pod Autoscaling (HPA)
- Cluster Autoscaling
- Vertical Pod Autoscaling (VPA)

Container spec

```
resources:
  requests:
    memory: 1 GB
    cpu: 1 Core
  limits:
    memory: 1 GB
    cpu: 1 Core
```



Pod 1
1 Core/1 GB

Pod 2
1 Core/1 GB

Pod 3
1 Core/1 GB

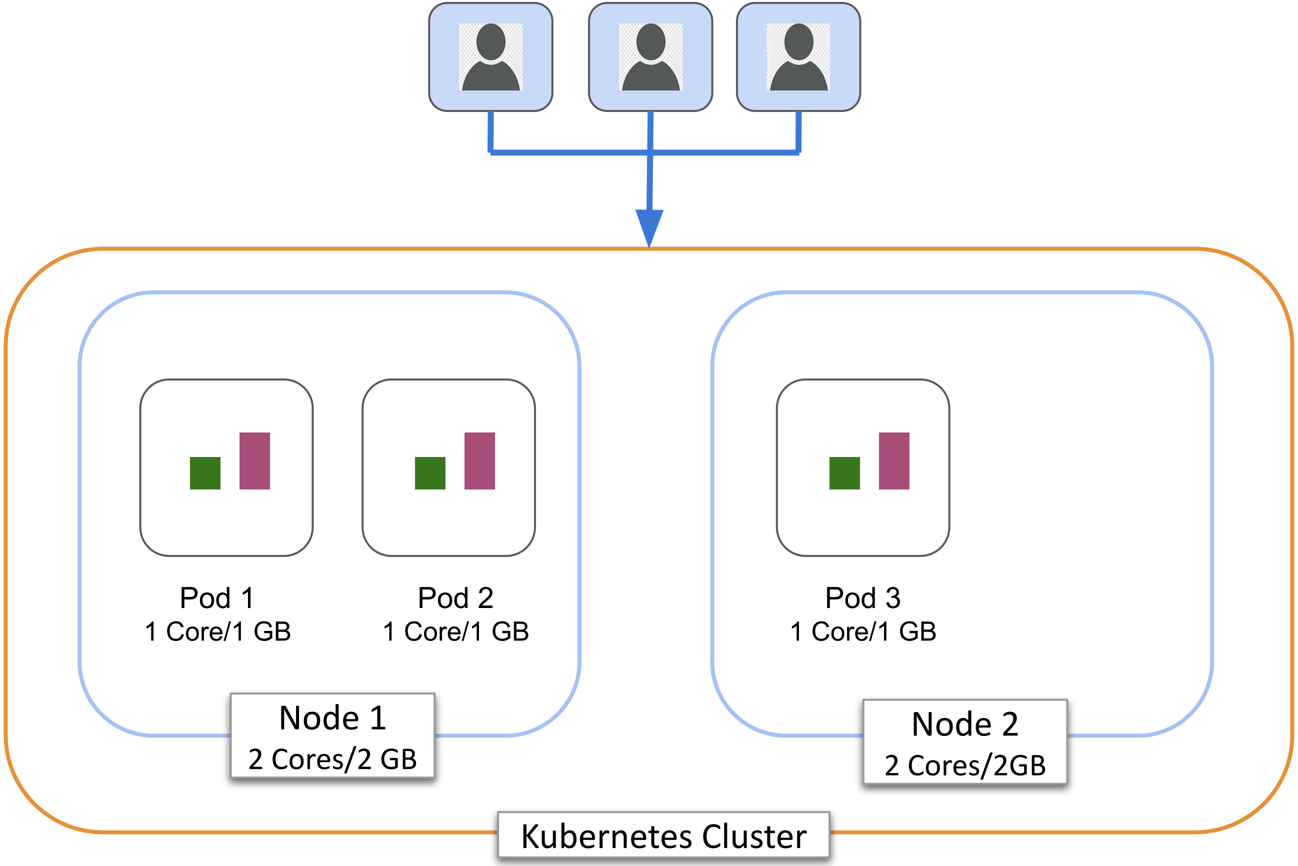
Node 1
2 Cores/2 GB

Node 2
2 Cores/2GB

Kubernetes Cluster

Container spec

```
resources:
  requests:
    memory: 1 GB
    cpu: 1 Core
  limits:
    memory: 1 GB
    cpu: 1 Core
```

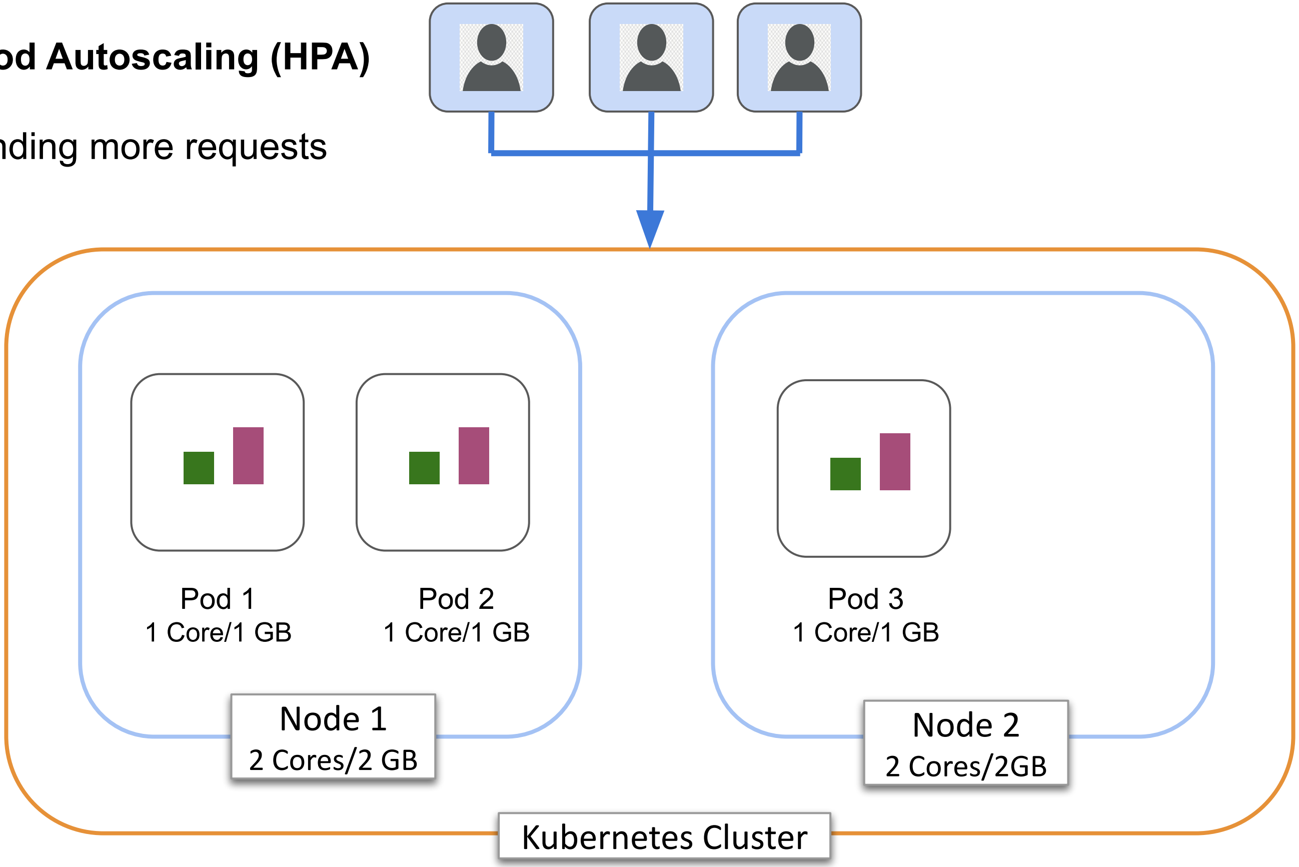


Horizontal Pod Autoscaling (HPA)

- great for handing more requests

Container spec

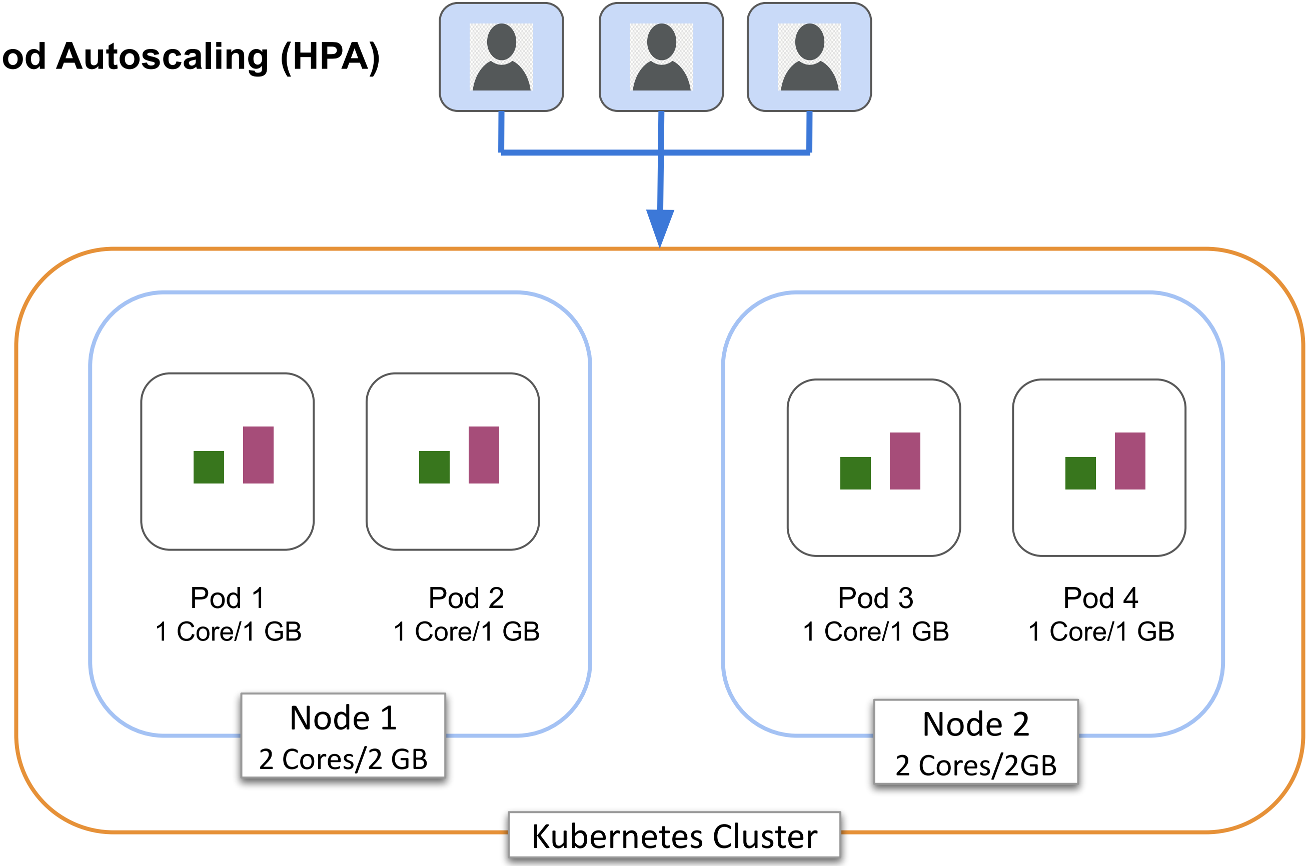
```
resources:
  requests:
    memory: 1 GB
    cpu: 1 Core
  limits:
    memory: 1 GB
    cpu: 1 Core
```

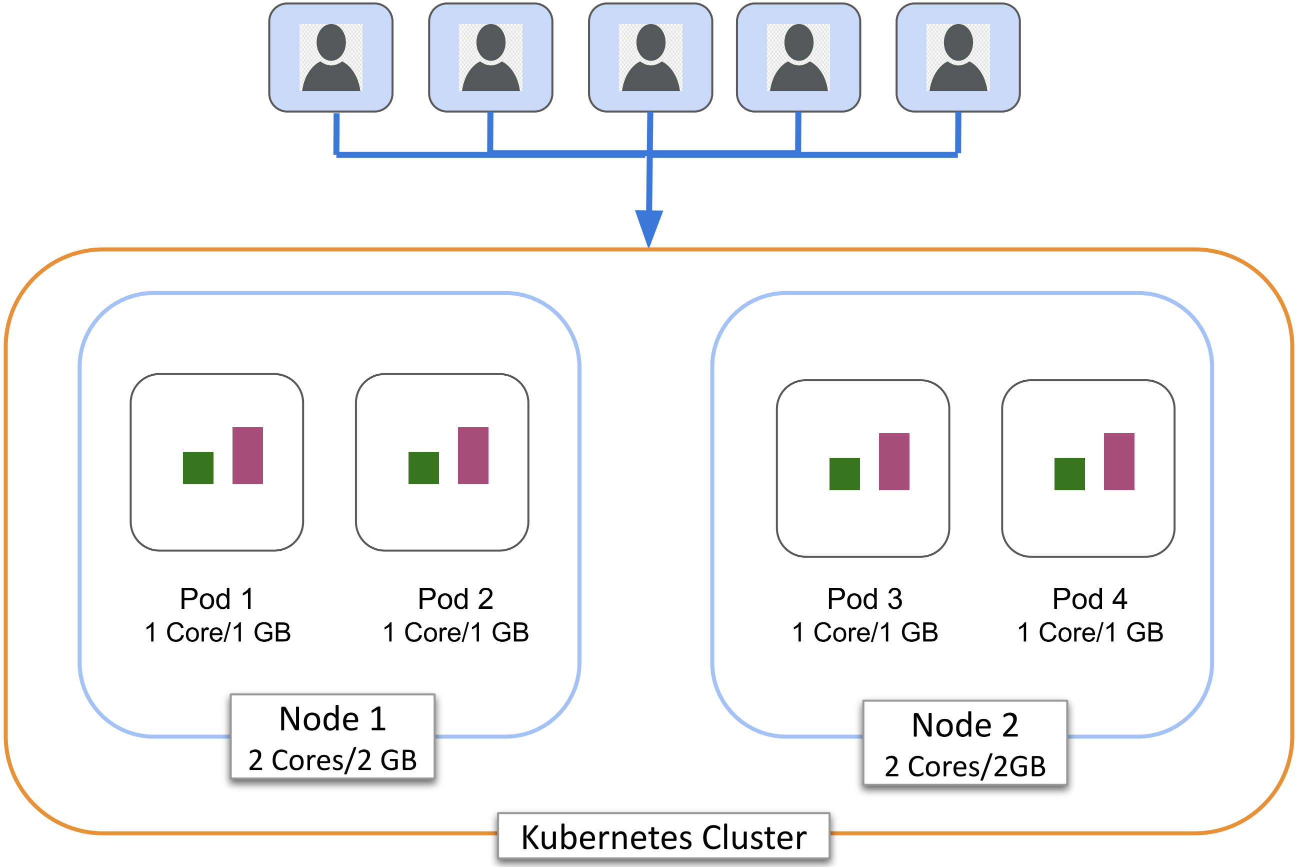


Horizontal Pod Autoscaling (HPA)

Container spec

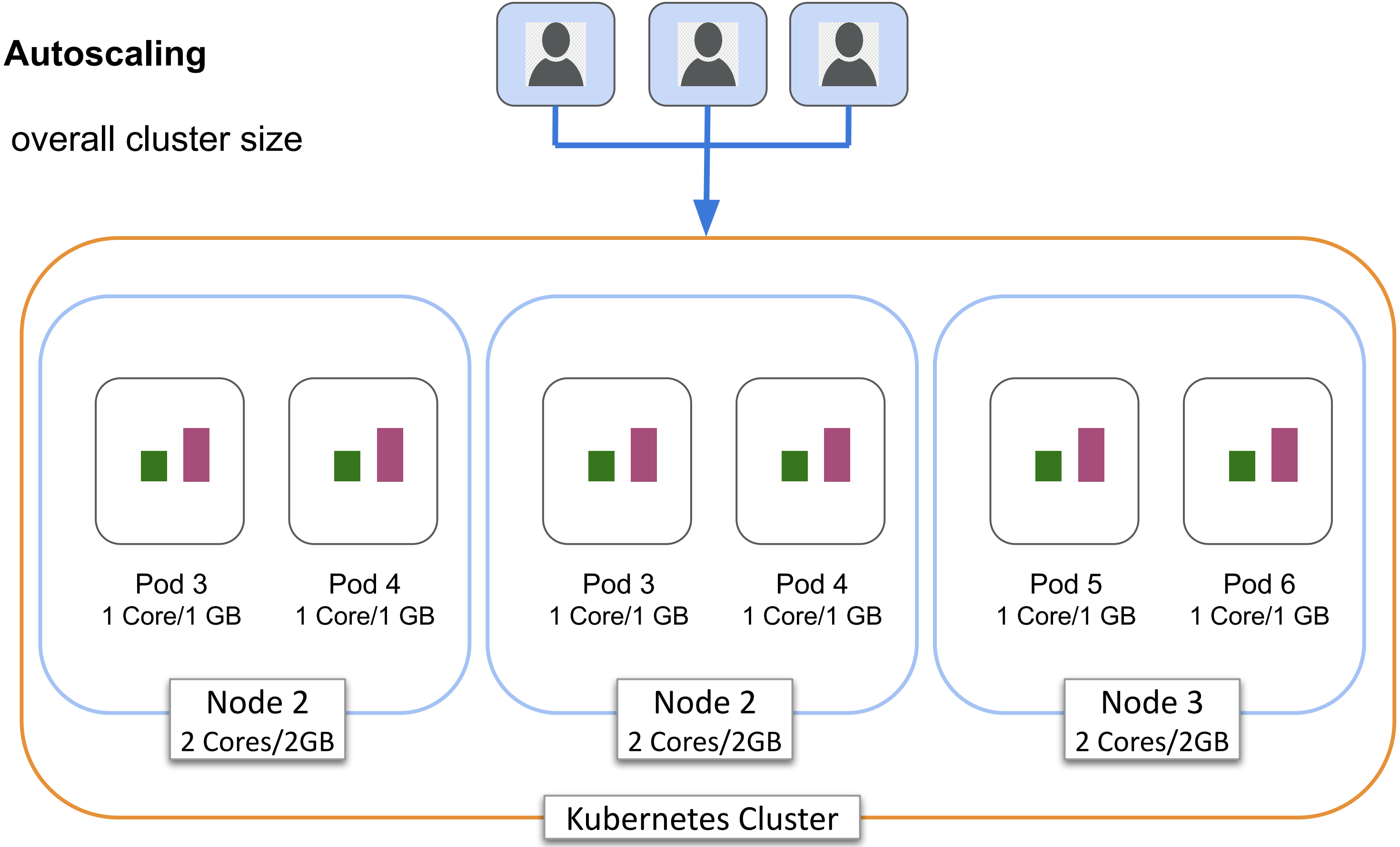
```
resources:
  requests:
    memory: 1 GB
    cpu: 1 Core
  limits:
    memory: 1 GB
    cpu: 1 Core
```

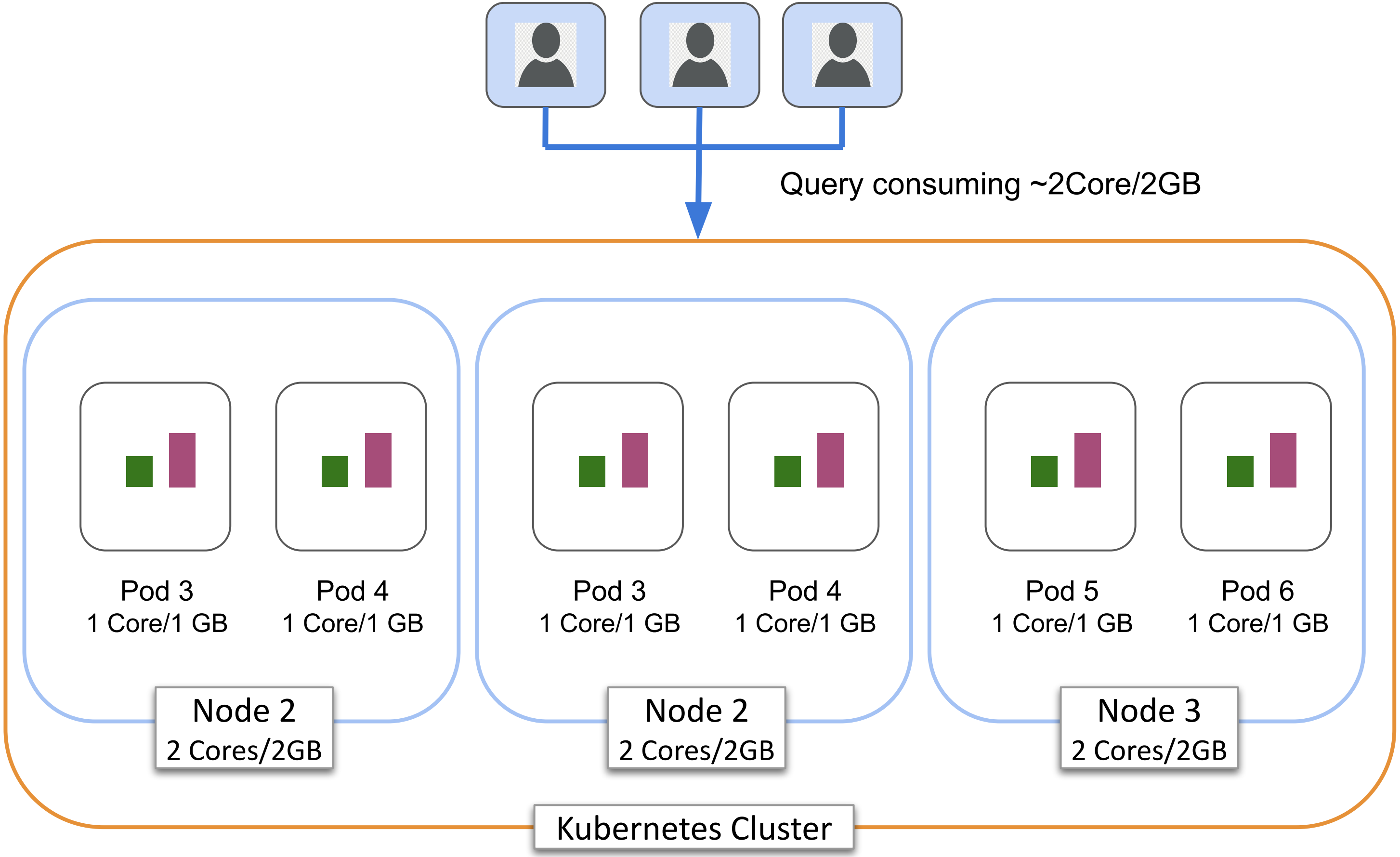



Pod 1
1 Core/1 GB
Pod 2
1 Core/1 GB
Pod 3
1 Core/1 GB
Pod 4
1 Core/1 GB
Node 1
2 Cores/2 GB
Node 2
2 Cores/2GB
Kubernetes Cluster

Cluster Autoscaling

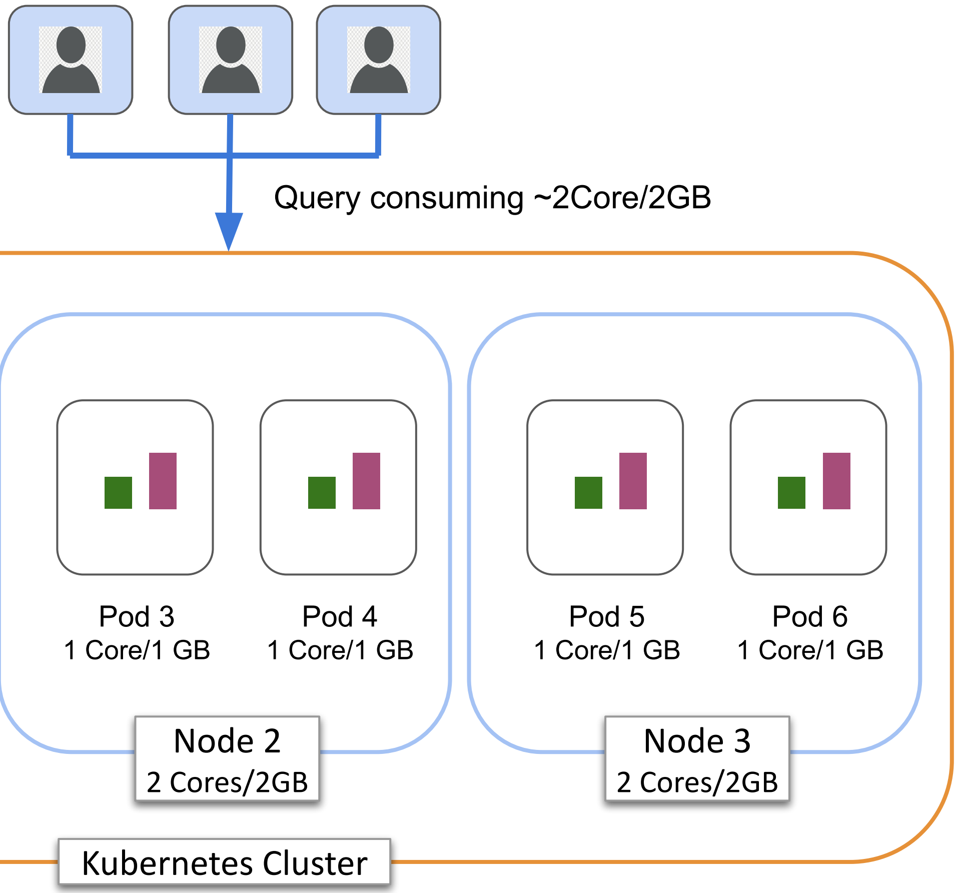
- Scales overall cluster size





Query consuming ~2Core/2GB
Pod 3
1 Core/1 GB
Pod 4
1 Core/1 GB
Node 2
2 Cores/2GB
Pod 3
1 Core/1 GB
Pod 4
1 Core/1 GB
Node 2
2 Cores/2GB
Pod 5
1 Core/1 GB
Pod 6
1 Core/1 GB
Node 3
2 Cores/2GB
Kubernetes Cluster

Vertical Pod Autoscaling (VPA)



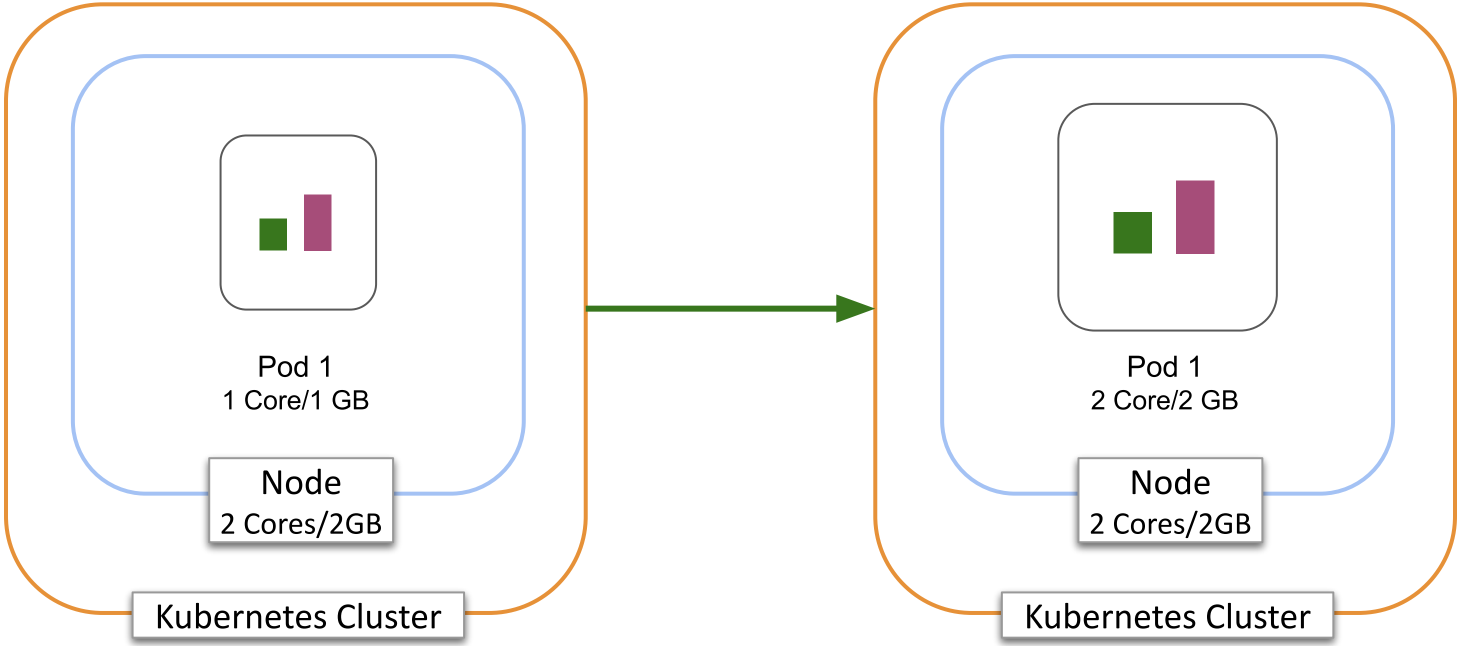
VPA Recommendation

```
resources:
  requests:
    memory: 2 GB
    cpu: 2 Core
  limits:
    memory: 2 GB
    cpu: 2 Core
```

Vertical Pod Autoscaling (VPA)

VPA Recommendation

```
resources:
  requests:
    memory: 2 GB
    cpu: 2 Core
  limits:
    memory: 2 GB
    cpu: 2 Core
```



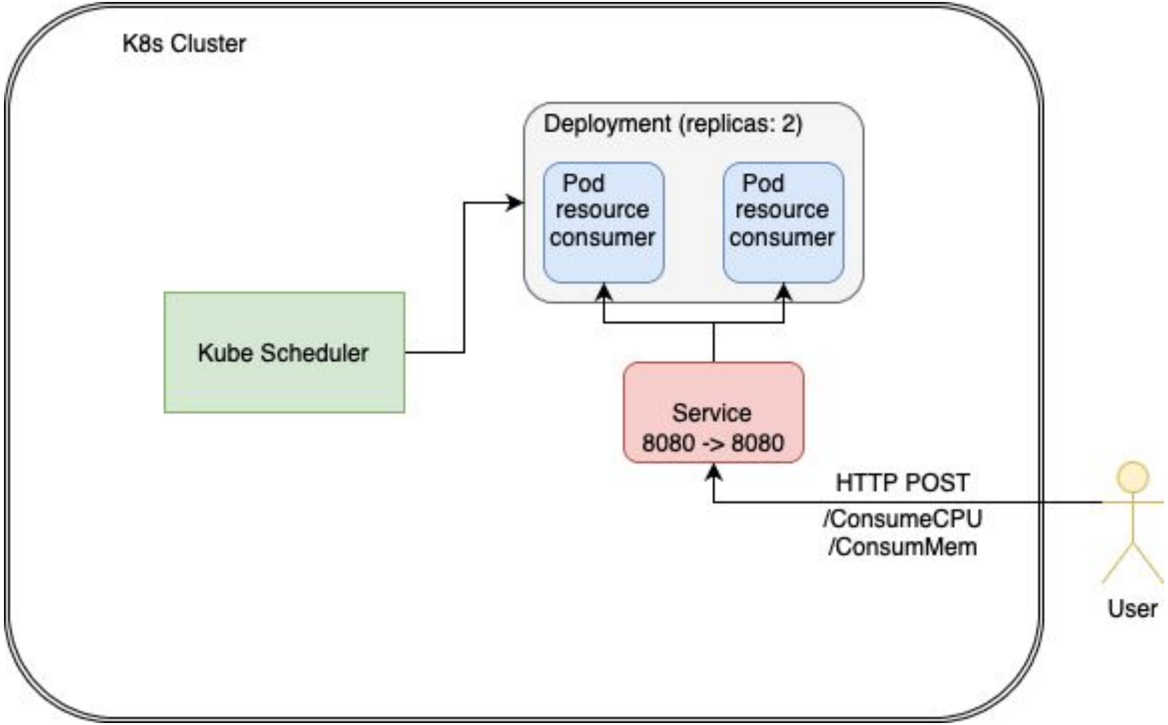
What we achieved in last sprint

(Sprint chicago)

- Installed VPA and Grafana operators in both MOC and Operate First Production cluster(Smaug).
- Deployed Resource Consumer(RC).
- Automated requests to the RC using Cron Jobs.
- Created a Grafana instance.

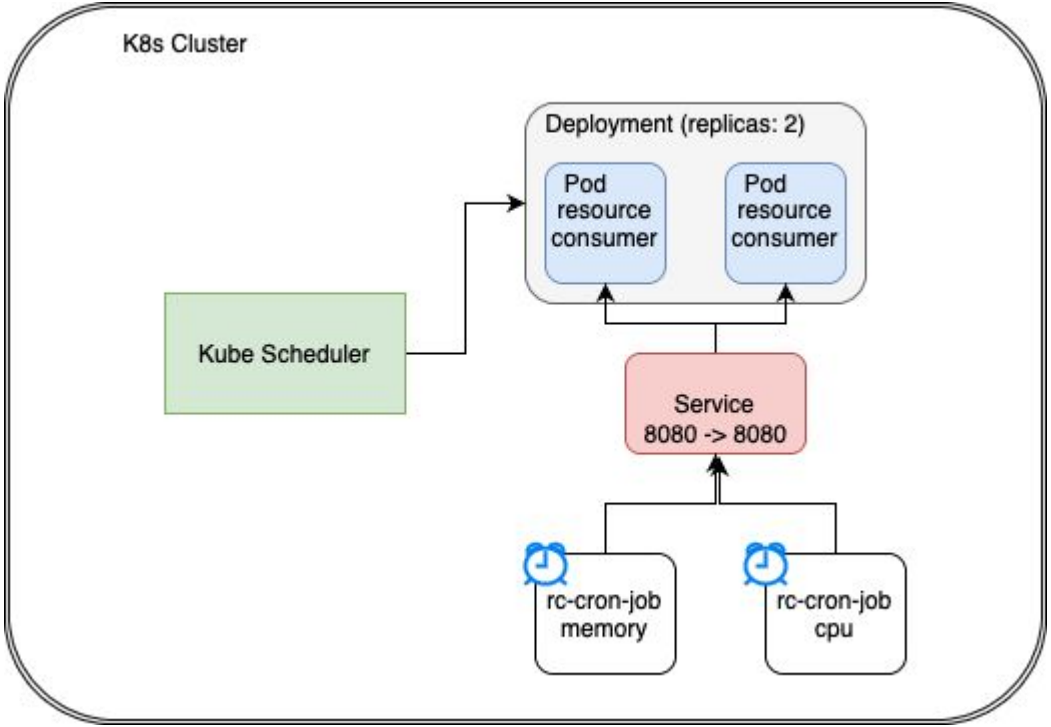
AUTOMATED REQUESTS TO RESOURCE CONSUMER

Manual requests via CLI



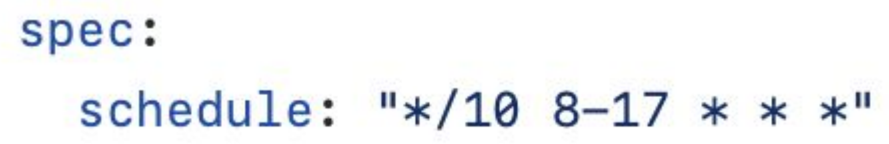
AUTOMATED REQUESTS TO RESOURCE CONSUMER

Automated requests using CronJob



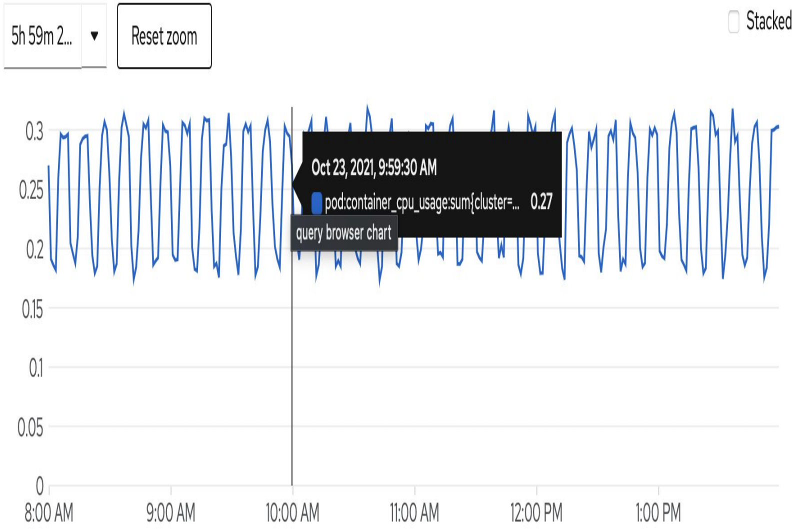
AUTOMATED REQUESTS TO RESOURCE CONSUMER

- Scheduled to send HTTP requests to /ConsumeMem and /ConsumeCPU every 10 min in the working hours (8AM - 5PM) daily.

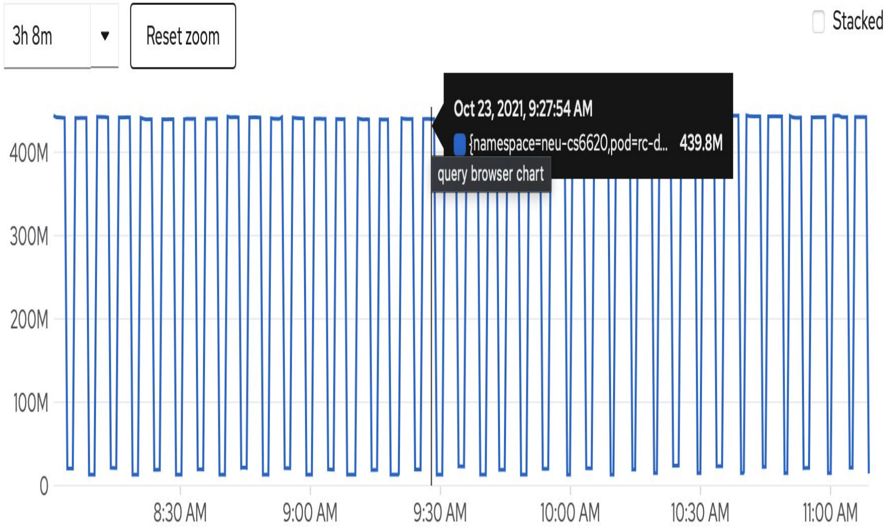


```
spec:
  schedule: "*/10 8-17 * * *"
```

AUTOMATED REQUESTS TO RESOURCE CONSUMER

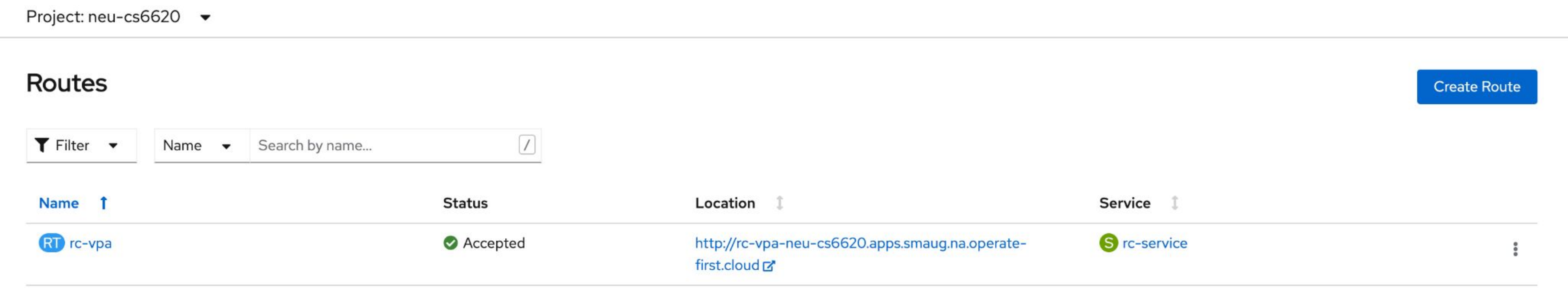


Requests from CronJob to consume CPU



Requests from CronJob to consume Memory

AUTOMATE ROUTE CONFIGURATION

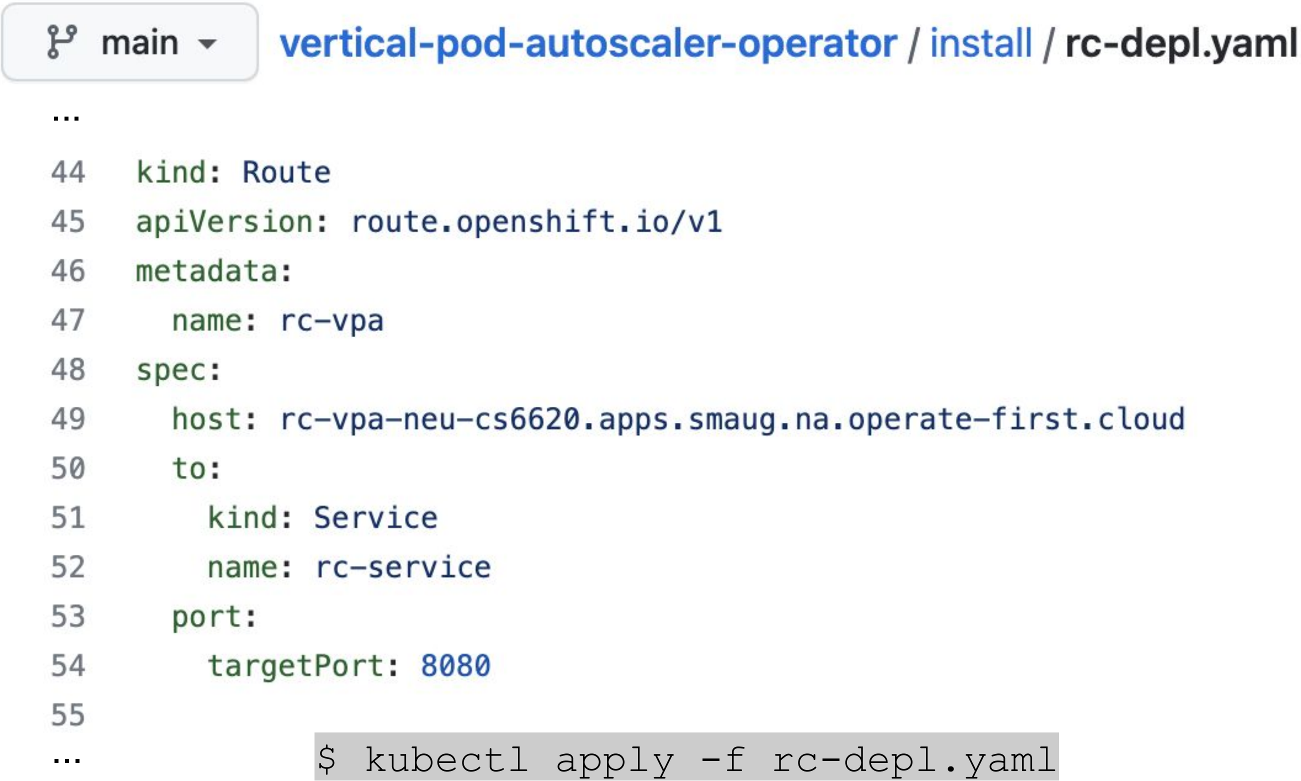


Project: neu-cs6620

Routes

Name	Status	Location	Service
RT rc-vpa	✓ Accepted	http://rc-vpa-neu-cs6620.apps.smaug.na.operate-first.cloud	S rc-service

Github



main

vertical-pod-autoscaler-operator / install / rc-depl.yaml

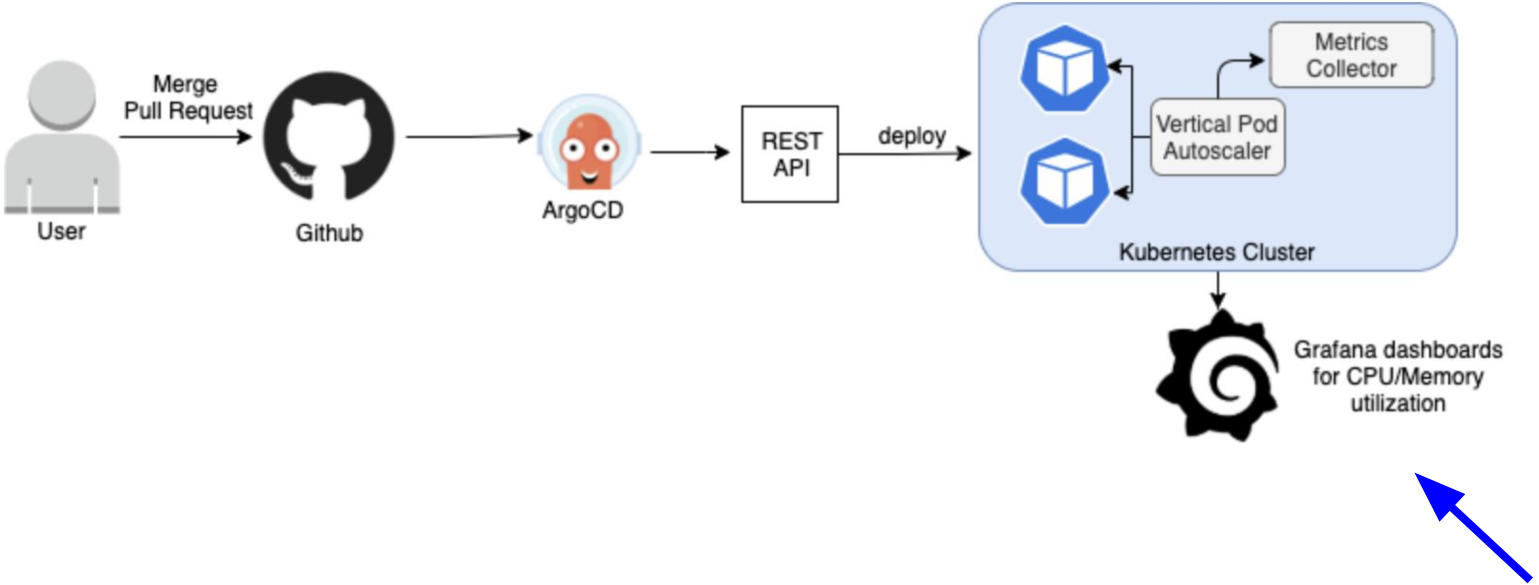
```
...
kind: Route
apiVersion: route.openshift.io/v1
metadata:
  name: rc-vpa
spec:
  host: rc-vpa-neu-cs6620.apps.smaug.na.operate-first.cloud
  to:
    kind: Service
    name: rc-service
  port:
    targetPort: 8080

...
```

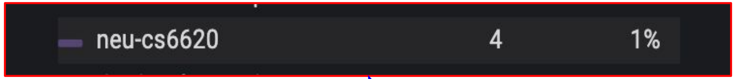
```
$ kubectl apply -f rc-depl.yaml
```

```
$ curl --insecure --data "milicores=900&durationSec=600"
http://rc-vpa-neu-cs6620.apps.smaug.na.operate-first.cloud:8080/ConsumeCPU
```

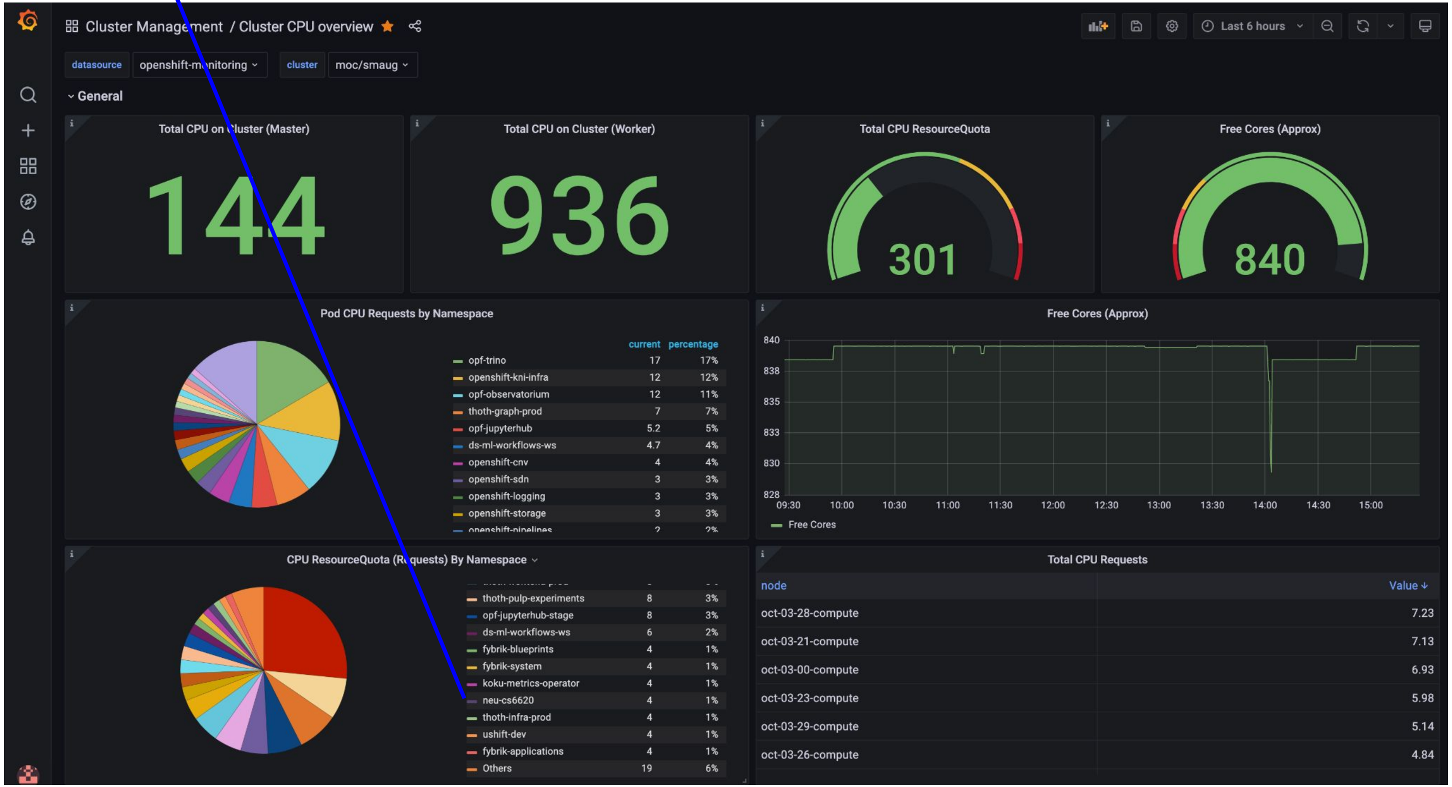
GRAFANA


User
Merge
Pull Request
Github
ArgoCD
REST
API
deploy
Metrics
Collector
Vertical Pod
Autoscaler
Kubernetes Cluster
Grafana dashboards
for CPU/Memory
utilization

GRAFANA



neu-cs6620 4 1%



Cluster Management / Cluster CPU overview

datasource: openshift-monitoring

cluster: moc/smaug

General

Total CPU on Cluster (Master): 144

Total CPU on Cluster (Worker): 936

Total CPU ResourceQuota: 301

Free Cores (Approx): 840

Pod CPU Requests by Namespace

	current	percentage
opf-trino	17	17%
openshift-kni-infra	12	12%
opf-observatorium	12	11%
thoth-graph-prod	7	7%
opf-jupyterhub	5.2	5%
ds-ml-workflows-ws	4.7	4%
openshift-cnv	4	4%
openshift-sdn	3	3%
openshift-logging	3	3%
openshift-storage	3	3%
openshift-pipelines	2	2%

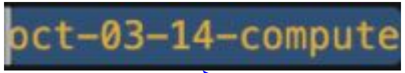
Free Cores (Approx)

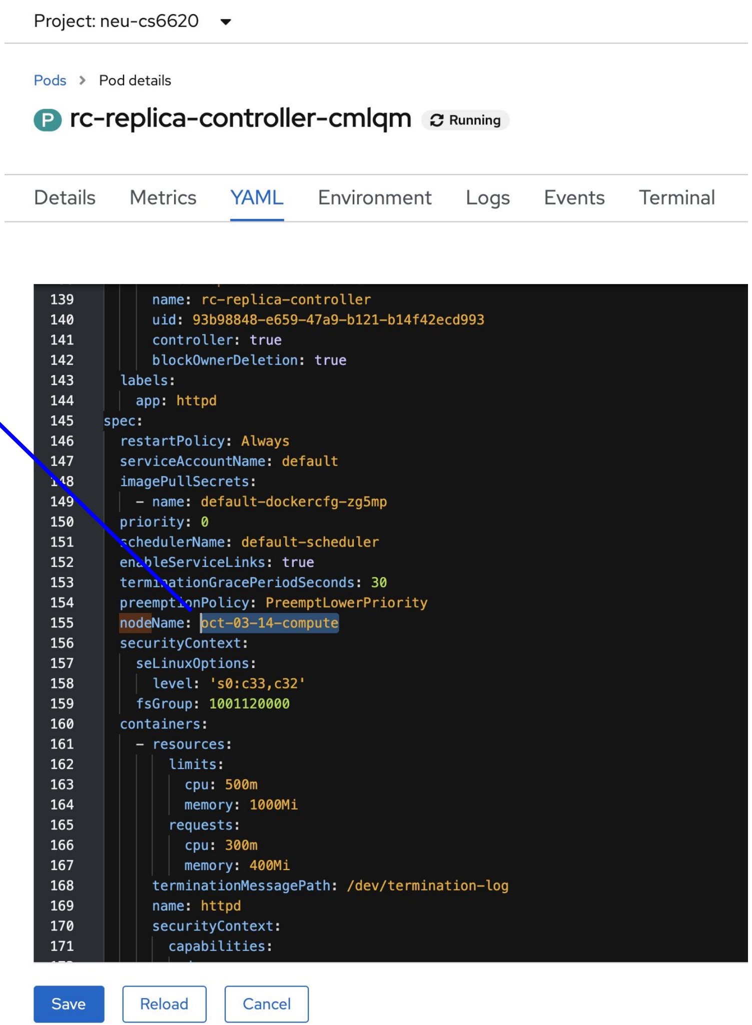
CPU ResourceQuota (Requests) By Namespace

thoth-pulp-experiments	8	3%
opf-jupyterhub-stage	8	3%
ds-ml-workflows-ws	6	2%
fybrik-blueprints	4	1%
fybrik-system	4	1%
koku-metrics-operator	4	1%
neu-cs6620	4	1%
thoth-infra-prod	4	1%
ushift-dev	4	1%
fybrik-applications	4	1%
Others	19	6%

Total CPU Requests

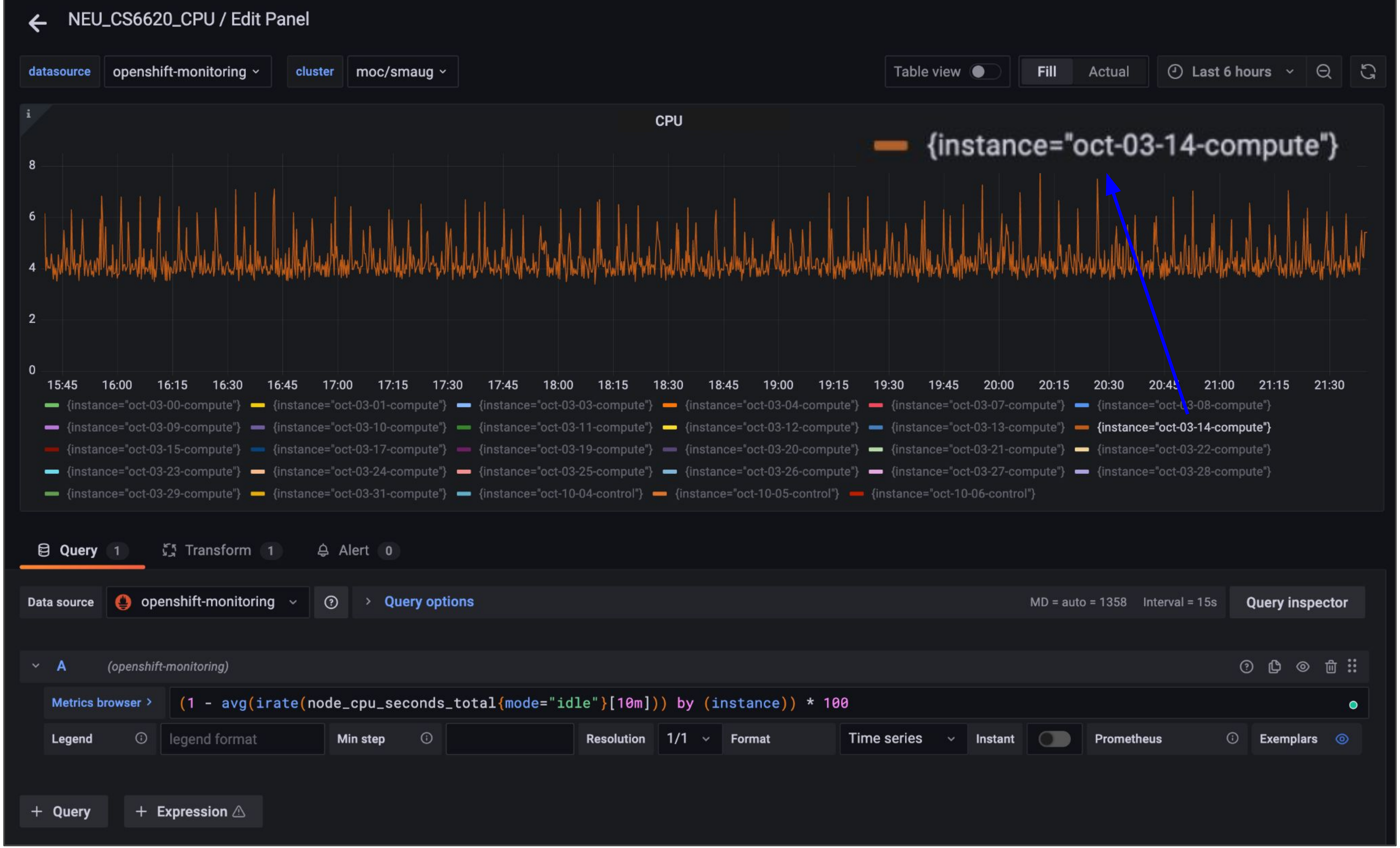
node	Value ↓
oct-03-28-compute	7.23
oct-03-21-compute	7.13
oct-03-00-compute	6.93
oct-03-23-compute	5.98
oct-03-29-compute	5.14
oct-03-26-compute	4.84





rc-replica-controller-cmlqm

```
      name: rc-replica-controller
      uid: 93b98848-e659-47a9-b121-b14f42ecd993
      controller: true
      blockOwnerDeletion: true
  labels:
    app: httpd
spec:
  restartPolicy: Always
  serviceAccountName: default
  imagePullSecrets:
    - name: default-dockercfg-zg5mp
  priority: 0
  schedulerName: default-scheduler
  enableServiceLinks: true
  terminationGracePeriodSeconds: 30
  preemptionPolicy: PreemptLowerPriority
  nodeName: pct-03-14-compute
  securityContext:
    seLinuxOptions:
      level: 's0:c33,c32'
    fsGroup: 1001120000
  containers:
    - resources:
        limits:
          cpu: 500m
          memory: 1000Mi
        requests:
          cpu: 300m
          memory: 400Mi
      terminationMessagePath: /dev/termination-log
      name: httpd
      securityContext:
        capabilities:
```



NEU_CS6620_CPU / Edit Panel

datasource: openshift-monitoring

cluster: moc/smaug

Table view

Fill | Actual

Last 6 hours

CPU
{instance="oct-03-14-compute"}
8
6
4
2
0
15:45
16:00
16:15
16:30
16:45
17:00
17:15
17:30
17:45
18:00
18:15
18:30
18:45
19:00
19:15
19:30
19:45
20:00
20:15
20:30
20:45
21:00
21:15
21:30
{instance="oct-03-00-compute"}
{instance="oct-03-01-compute"}
{instance="oct-03-03-compute"}
{instance="oct-03-04-compute"}
{instance="oct-03-07-compute"}
{instance="oct-03-08-compute"}
{instance="oct-03-09-compute"}
{instance="oct-03-10-compute"}
{instance="oct-03-11-compute"}
{instance="oct-03-12-compute"}
{instance="oct-03-13-compute"}
{instance="oct-03-14-compute"}
{instance="oct-03-15-compute"}
{instance="oct-03-17-compute"}
{instance="oct-03-19-compute"}
{instance="oct-03-20-compute"}
{instance="oct-03-21-compute"}
{instance="oct-03-22-compute"}
{instance="oct-03-23-compute"}
{instance="oct-03-24-compute"}
{instance="oct-03-25-compute"}
{instance="oct-03-26-compute"}
{instance="oct-03-27-compute"}
{instance="oct-03-28-compute"}
{instance="oct-03-29-compute"}
{instance="oct-03-31-compute"}
{instance="oct-10-04-control"}
{instance="oct-10-05-control"}
{instance="oct-10-06-control"}
Query 1
Transform 1
Alert 0
Query options
Query inspector
+ Query
+ Expression

Data source: openshift-monitoring

MD = auto = 1358 Interval = 15s

A (openshift-monitoring)

Metrics browser >

```
(1 - avg(irate(node_cpu_seconds_total{mode="idle"}[10m])) by (instance)) * 100
```

Legend: legend format

Min step:

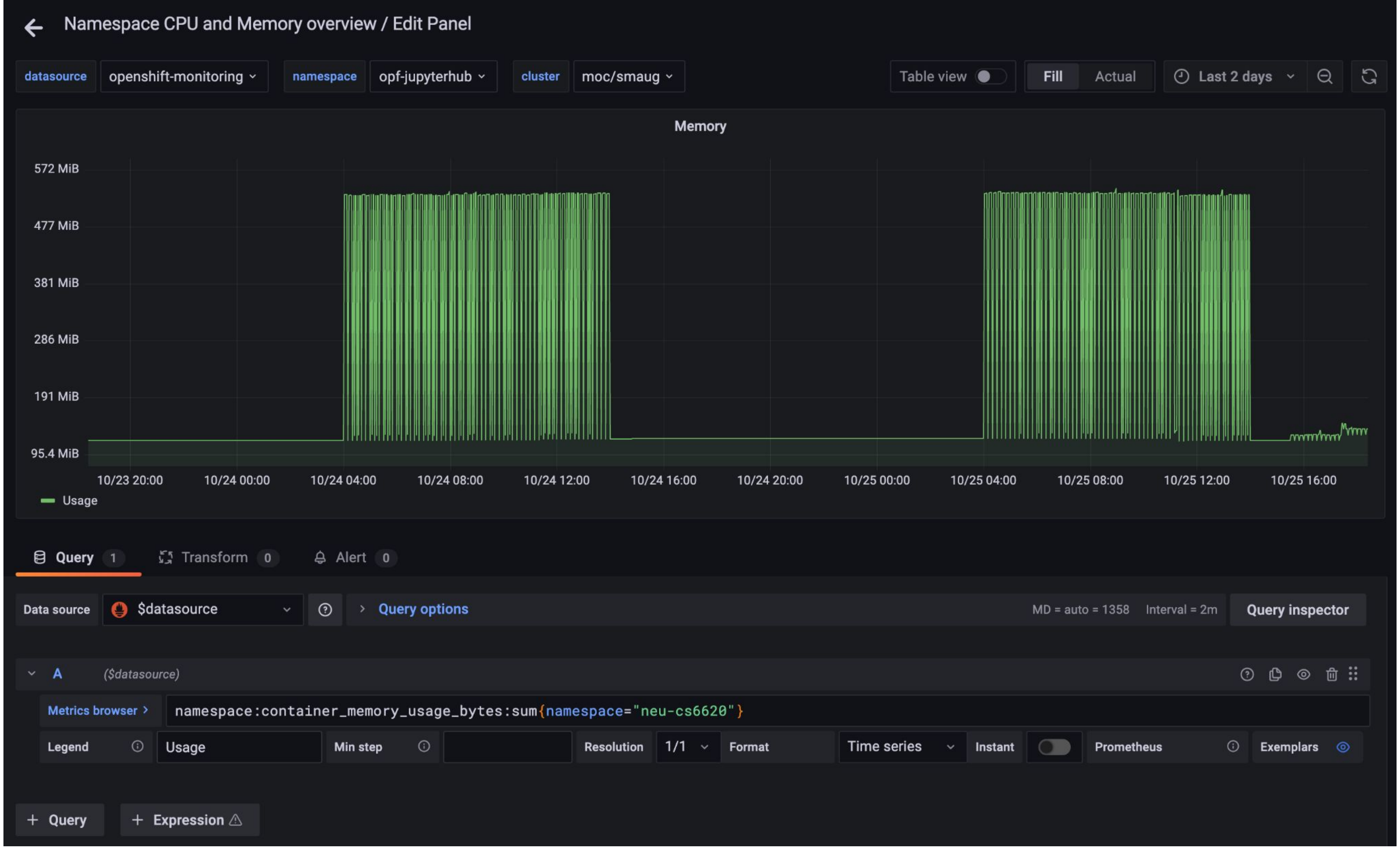
Resolution: 1/1

Format: Time series

Instant

Prometheus

Exemplars



datasource: openshift-monitoring

namespace: opf-jupyterhub

cluster: moc/smaug

Data source: $datasource

MD = auto = 1358 Interval = 2m

A ($datasource)

```
namespace:container_memory_usage_bytes:sum{namespace="neu-cs6620"}
```

Legend: Usage

Min step:

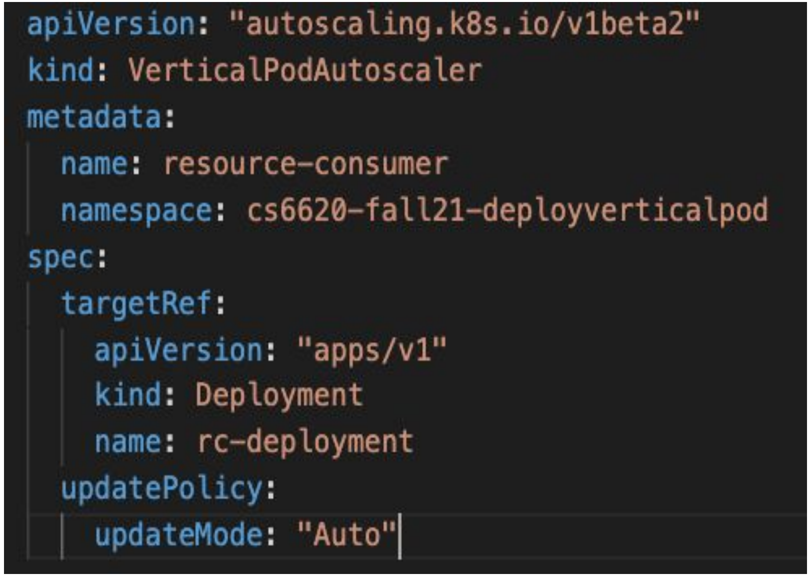
Resolution: 1/1

Format: Time series

Instant: off

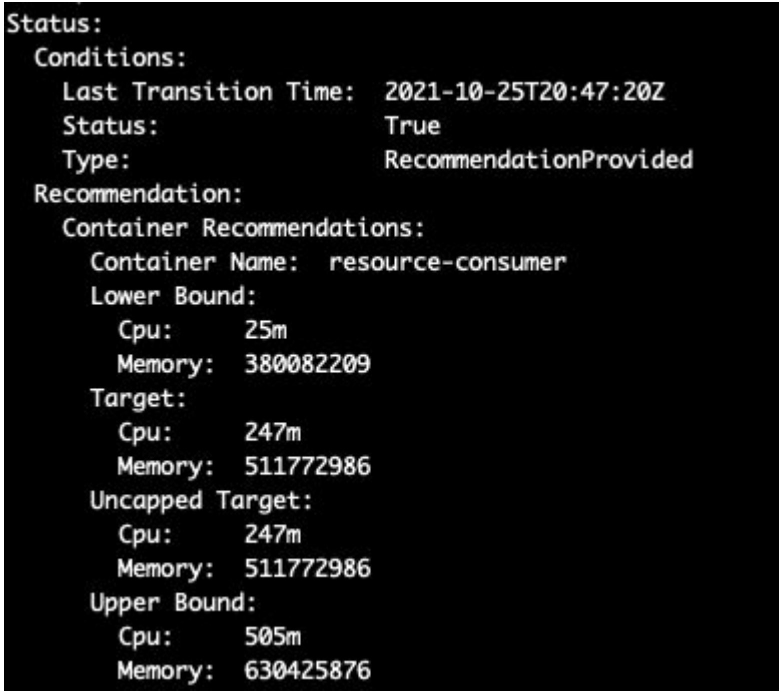
VPA Recommendation

VPA Custom Resource



```yaml
apiVersion: "autoscaling.k8s.io/v1beta2"
kind: VerticalPodAutoscaler
metadata:
  name: resource-consumer
  namespace: cs6620-fall21-deployverticalpod
spec:
  targetRef:
    apiVersion: "apps/v1"
    kind: Deployment
    name: rc-deployment
  updatePolicy:
    updateMode: "Auto"
```

Recommendation:



```
Status:
  Conditions:
    Last Transition Time:  2021-10-25T20:47:20Z
    Status:                True
    Type:                  RecommendationProvided
  Recommendation:
    Container Recommendations:
      Container Name:  resource-consumer
      Lower Bound:
        Cpu:     25m
        Memory:  380082209
      Target:
        Cpu:     247m
        Memory:  511772986
      Uncapped Target:
        Cpu:     247m
        Memory:  511772986
      Upper Bound:
        Cpu:     505m
        Memory:  630425876
```

Resource metrics

```
resources:
 requests:
  memory: 1 GB
  cpu: 1 Core
 limits:
  memory: 1 GB
  cpu: 1 Core
```

Limit

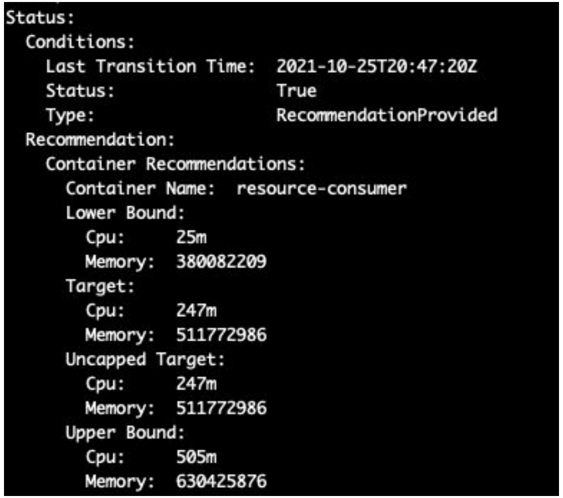
The kubelet enforces those ***limits*** so that the running container is not allowed to use more of that resource than the limit you set.

Request

The kubelet also reserves at least the ***request*** amount of that system resource specifically for that container to use

VPA Recommendation Terms

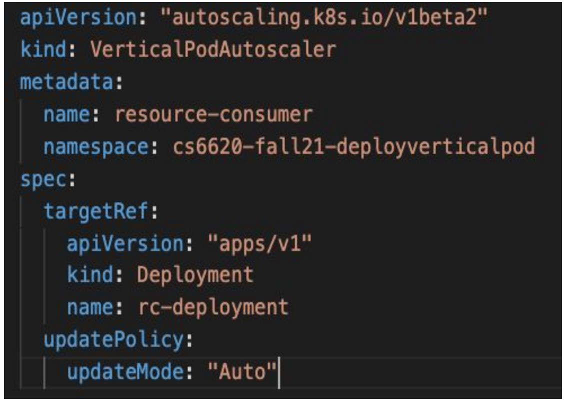
- **lowerBound** - the minimum recommended resource levels.
- **target** - the recommended resource levels.
- **upperBound** - the highest recommended resource levels.
- **uncappedTarget** - the most recent resource recommendations.



```
Status:
  Conditions:
    Last Transition Time:  2021-10-25T20:47:20Z
    Status:                True
    Type:                  RecommendationProvided
  Recommendation:
    Container Recommendations:
      Container Name:  resource-consumer
      Lower Bound:
        Cpu:     25m
        Memory:  380082209
      Target:
        Cpu:     247m
        Memory:  511772986
      Uncapped Target:
        Cpu:     247m
        Memory:  511772986
      Upper Bound:
        Cpu:     505m
        Memory:  630425876
```

VPA Update policy

- **Auto** - Updates the pod based on its recommendation by deleting and recreating it.
- **Initial** - Applies recommendations only during pod creation.
- **Off** - Just analyzes and records the recommendations.

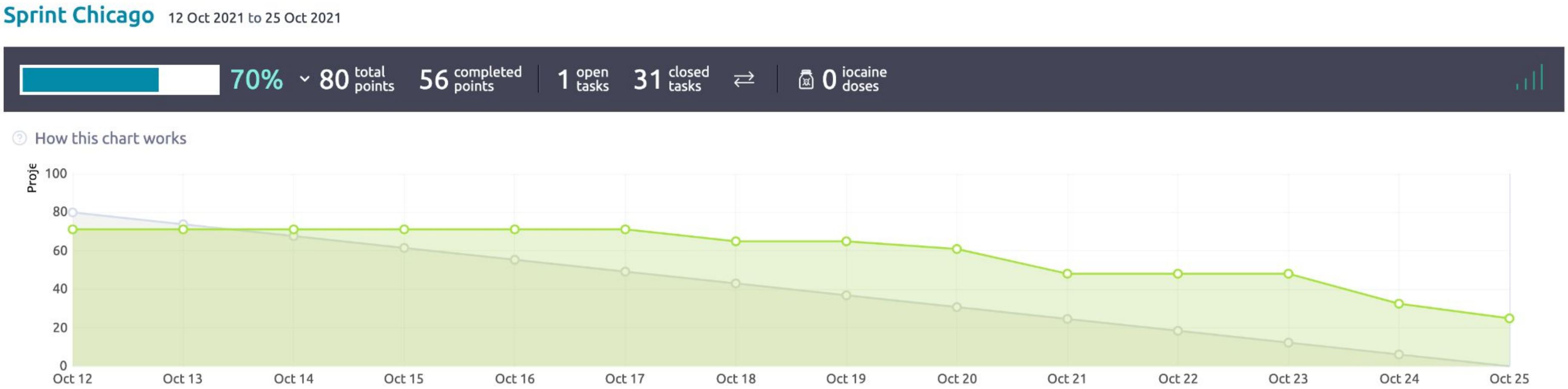


```yaml
apiVersion: "autoscaling.k8s.io/v1beta2"
kind: VerticalPodAutoscaler
metadata:
  name: resource-consumer
  namespace: cs6620-fall21-deployverticalpod
spec:
  targetRef:
    apiVersion: "apps/v1"
    kind: Deployment
    name: rc-deployment
  updatePolicy:
    updateMode: "Auto"
```

VPA Test cases

1. Pod over-utilization within upper bound of VPA recommendation.
2. Pod under-utilization within lower bound of VPA recommendations
3. VPA automatically deletes and creating a pod with its new recommendation for upperBound case (over-utilizing) within the limit
4. VPA automatically deletes and creating a pod with its new recommendation for lowerBound case (under-utilizing)

VPA Demo



Sprint Chicago 12 Oct 2021 to 25 Oct 2021

70% ⌄ 80 total points 56 completed points 1 open tasks 31 closed tasks ⇄ 0 iocaine doses

How this chart works

What we plan for Next sprint

(Sprint Dallas)

- Experiment VPA scenarios
  - Auto updatePolicy of VPA
  - Underutilization and Overutilization of resources
- Grafana Live Dashboard
  - Create Grafana dashboard with our metrics of interest
- Setup POC application through GitOps workflow