





Next Gen Cost Observability

The Ongoing Journey To Trustworthy Cost Metrics

Erik Sommer

Sr. Software Engineer

What you can expect

-  The Problem: When the bill doesn't match your metrics
-  Mental model to attributes costs in k8s
-  How to compose cost PromQL queries
-  Getting the resource rates from your cloud provider



▼ Elastic Compute Cloud		\$553.92
▼ US East (N. Virginia)		\$553.92
Amazon Elastic Compute Cloud NatGateway		\$90.32
\$0.045 per GB Data Processed by NAT Gateways	1,263.028 GB	\$56.84
\$0.045 per NAT Gateway Hour	744 Hrs	\$33.48
Amazon Elastic Compute Cloud running Linux/UNIX		\$342.15
\$0.0052 per On Demand Linux t3.nano Instance Hour	1,488 Hrs	\$7.74
\$0.0116 per On Demand Linux t2.micro Instance Hour	2,976 Hrs	\$34.52
\$0.0464 per On Demand Linux t2.medium Instance Hour	4,923.783 Hrs	\$228.46
\$0.096 per On Demand Linux m5.large Instance Hour	744 Hrs	\$71.42
EBS		\$79.25
\$0.00 for 480 Mbps per m5.large instance-hour (or partial hour)	744 Hrs	\$0.00
\$0.05 per GB-Month of snapshot data stored - US East (Northern Virginia)	254.756 GB-Mo	\$12.74
\$0.10 per GB-month of General Purpose SSD (gp2) provisioned storage - US East (Northern Virginia)	665.162 GB-Mo	\$66.52
Elastic Load Balancing - Classic		\$42.21
\$0.008 per GB Data Processed by the LoadBalancer	625.658 GB	\$5.01
\$0.025 per LoadBalancer-hour (or partial hour)	1,488 Hrs	\$37.20



▼ Elastic Compute Cloud

▼ US East (N. Virginia)

Amazon Monthly costs by service

\$0.045 p

\$0.045 p

Amazon

\$0.0052 p

\$0.0116 p

\$0.0464 p

\$0.096 p

EBS

\$0.00 for

\$0.05 per

\$0.10 per

Elastic L

\$0.008 p

\$0.025 p

Last 6 Months

Monthly

Stack

Group by: Instance Type x Service Linked Account Region Usage Type Tag API Operation Availability Zone More

Costs (\$ in thousands)



Download CSV

Instance Type	Oct 1, 2018	Nov 1, 2018	Dec 1, 2018	Jan 1, 2019
Total cost (\$)	1,312.71	1,328.54	1,125.99	1,129.65
t2.micro (\$)	486.75	475.89	405.63	409.27
c4.2xlarge (\$)	296.11	286.56	296.11	296.11

\$553.92

\$553.92

▲ FILTERS

CLEAR ALL

Service

Include only

EC2-Instances x

1

Linked Account

Include all

Region

Include all

Instance Type

Include all

Usage Type

Include all

Usage Type Group

Include all

Tag

Include All

API Operation

Include all

Charge Type

Include all

More filters

▲ ADVANCED OPTIONS

1

Show costs as

Unblended costs

Include costs related to

☐ Show only untagged resources



▼ Elastic Compute Cloud

\$553.92

▼ US East (N. Virginia)

\$553.92

Amazon Monthly costs by service

\$0.045 p

\$0.045 p

Amazon

\$0.0052 |

\$0.0116 |

\$0.0464 |

\$0.096 p

EBS

\$0.00 for

\$0.05 pe

\$0.10 pe

Elastic L

\$0.008 p

\$0.025 p

Last 6 Months

Monthly

Stack

▲ FILTERS

CLEAR ALL

Service

Include only

1.32

5.84

3.48

Group by: In

Costs (\$ in th

1.4

1.2

1.0

0.8

0.6

0.4

0.2

0.0

0.0

0.0

0.0

0.0

0.0

0.0

0.0

0.0

0.0

0.0

0.0

0.0

0.0

0.0

0.0

0.0

0.0

0.0

0.0

0.0

0.0

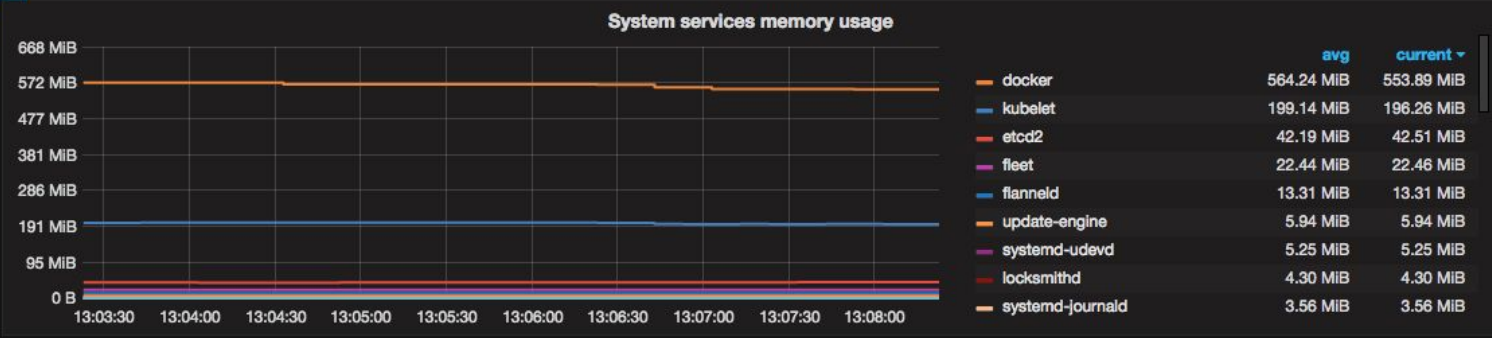
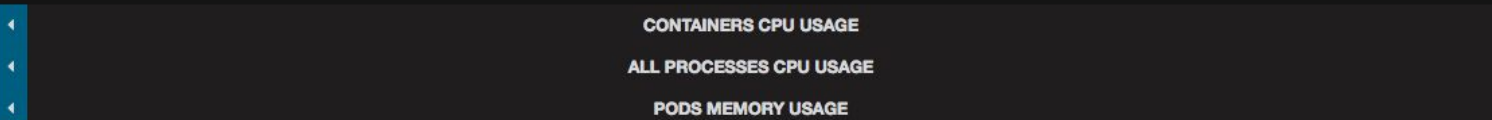
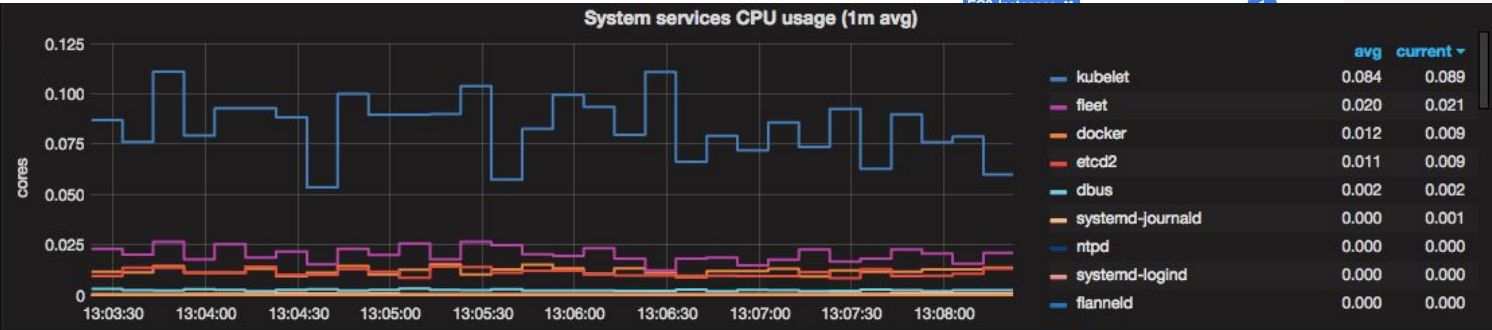
0.0

0.0

0.0

0.0

0.0



▼ Elastic Compute Cloud

\$553.92

▼ US East (N. Virginia)

\$553.92

Amazon Monthly costs by service

\$0.045 per

\$0.045 per

Amazon

\$0.0052 per

\$0.0116 per

\$0.0464 per

\$0.096 per

EBS

\$0.00 for

\$0.05 per

\$0.10 per

Elastic L

\$0.008 per

\$0.025 per

Last 6 Months

Monthly

Stack

FILTERS

CLEAR ALL

Service

Include only

1.32

5.84

3.48

Group by: In

Costs (\$ in th

1.4

1.2

1.0

0.8

0.6

0.4

0.2

0.0

0.0

0.0

0.0

0.0

0.0

0.0

0.0

0.0

0.0

0.0

0.0

0.0

0.0

0.0

0.0

0.0

0.0

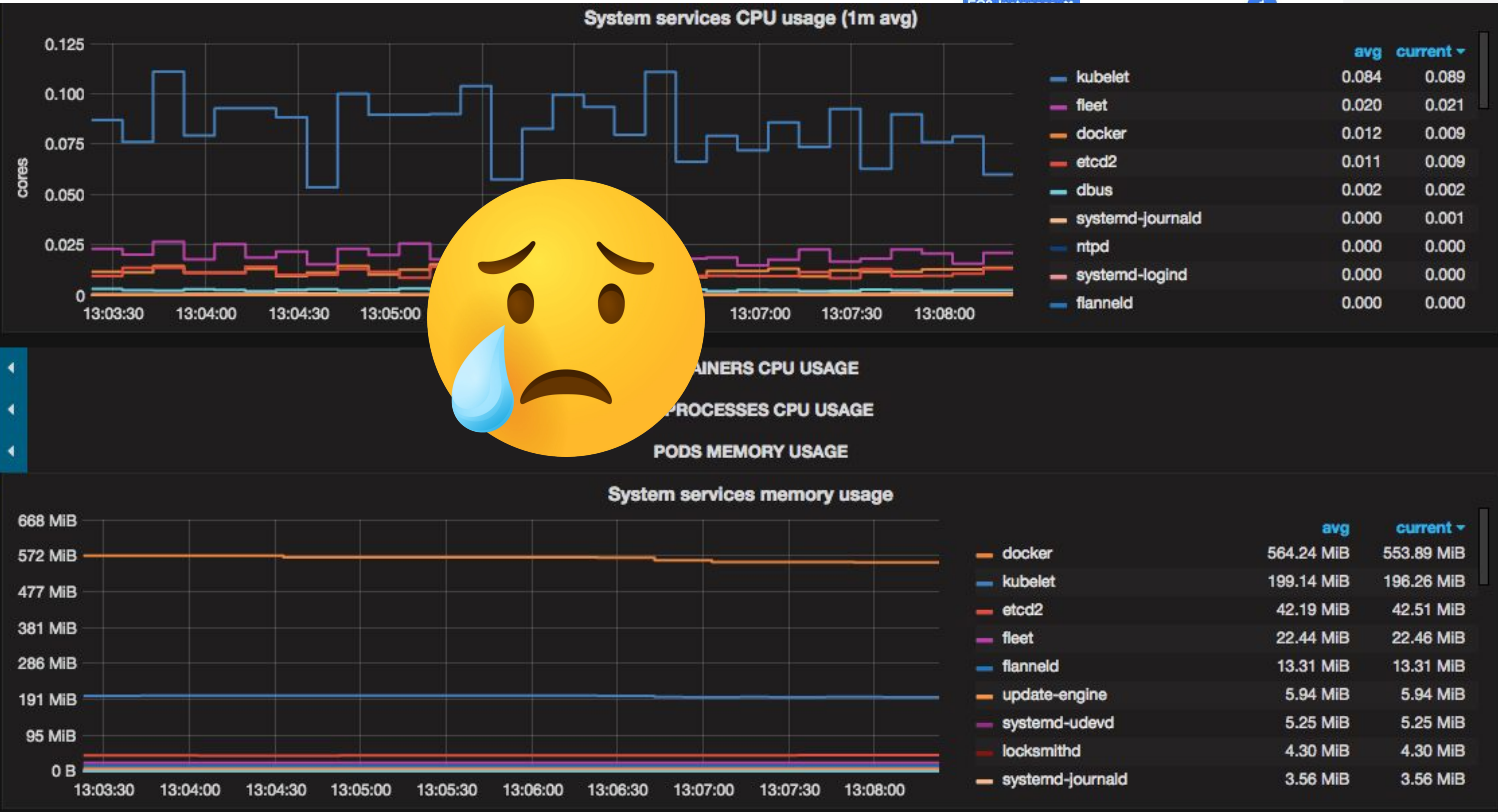
0.0

0.0

0.0

0.0

0.0



Problems

- Disconnect between billing statement and metrics
- Need to attribute costs of workloads running in Kubernetes
- Difficult to accurately measure Total Cost of Ownership (TCO) of our services




A simple formula with profound implications ...

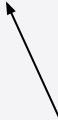
$$\text{Spend} = \text{Usage} \times \text{Rate}$$



A simple formula with profound implications ...

$$\text{Spend} = \text{Usage} \times \text{Rate}$$

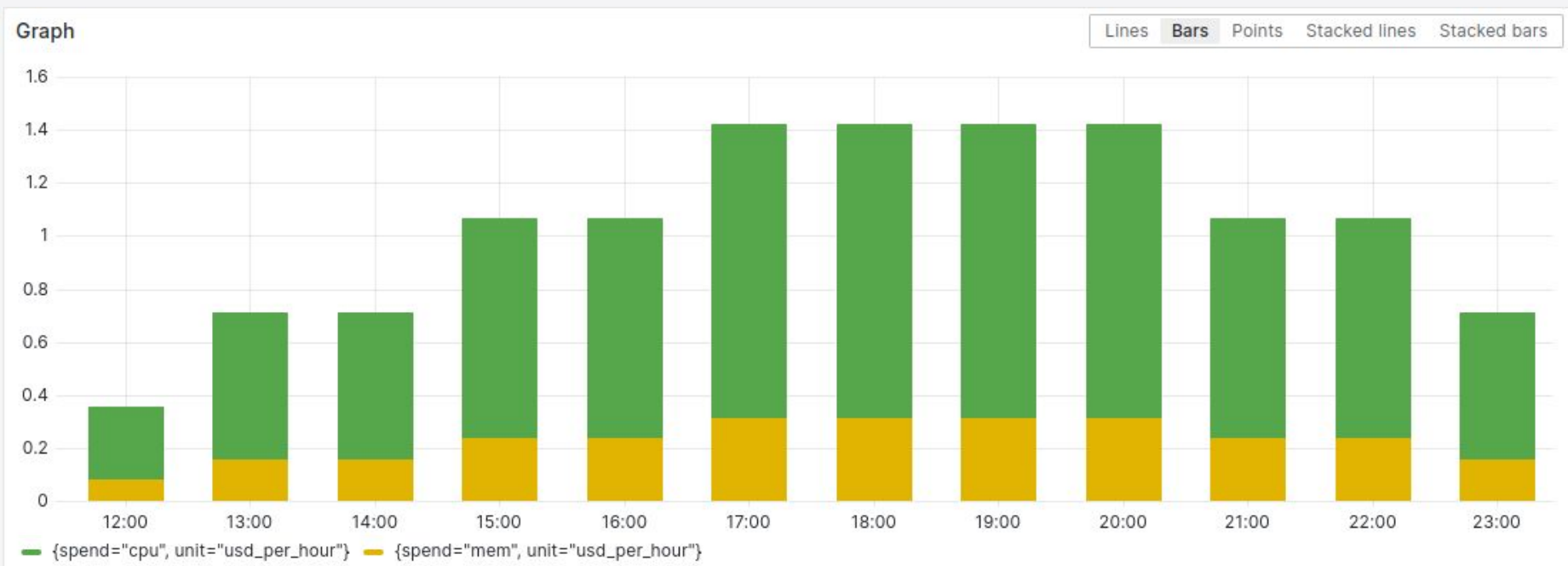
- 
- cpu cores
 - memory
 - transmitted network traffic
 - stored data

- 
- \$/hour
 - \$/GB



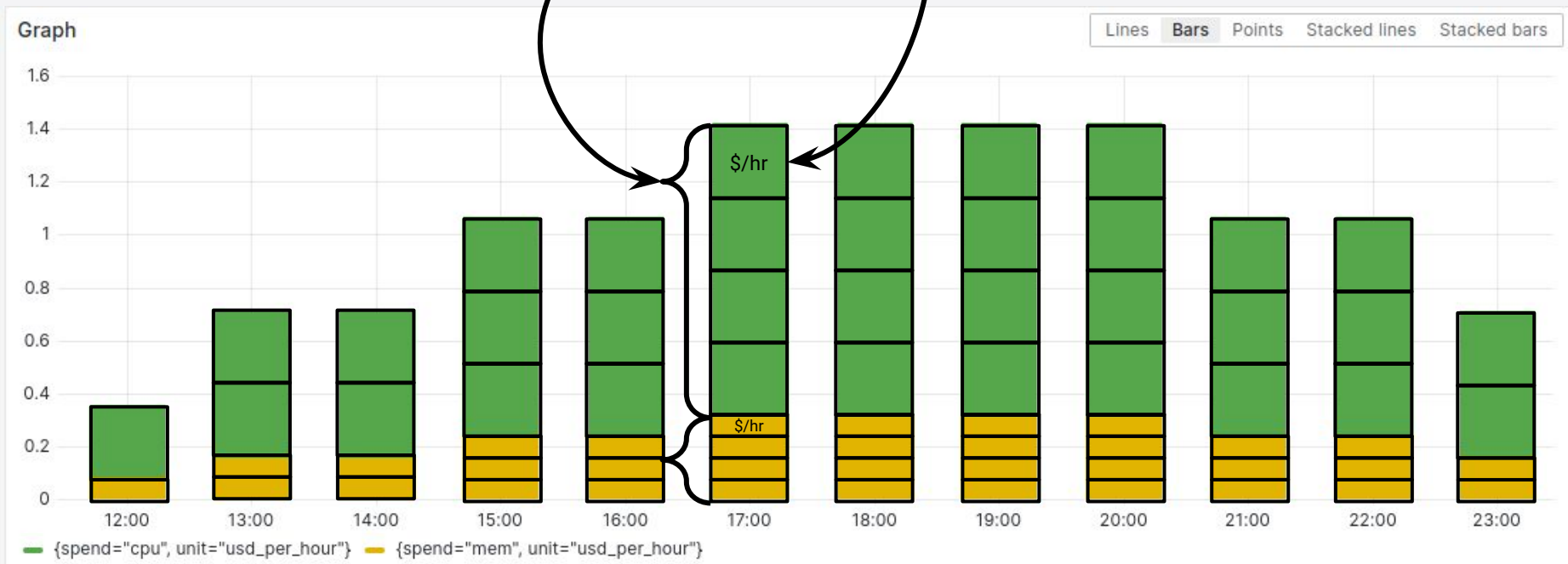
A simple formula with profound implications ...

$$\text{Spend} = \text{Usage} \times \text{Rate}$$



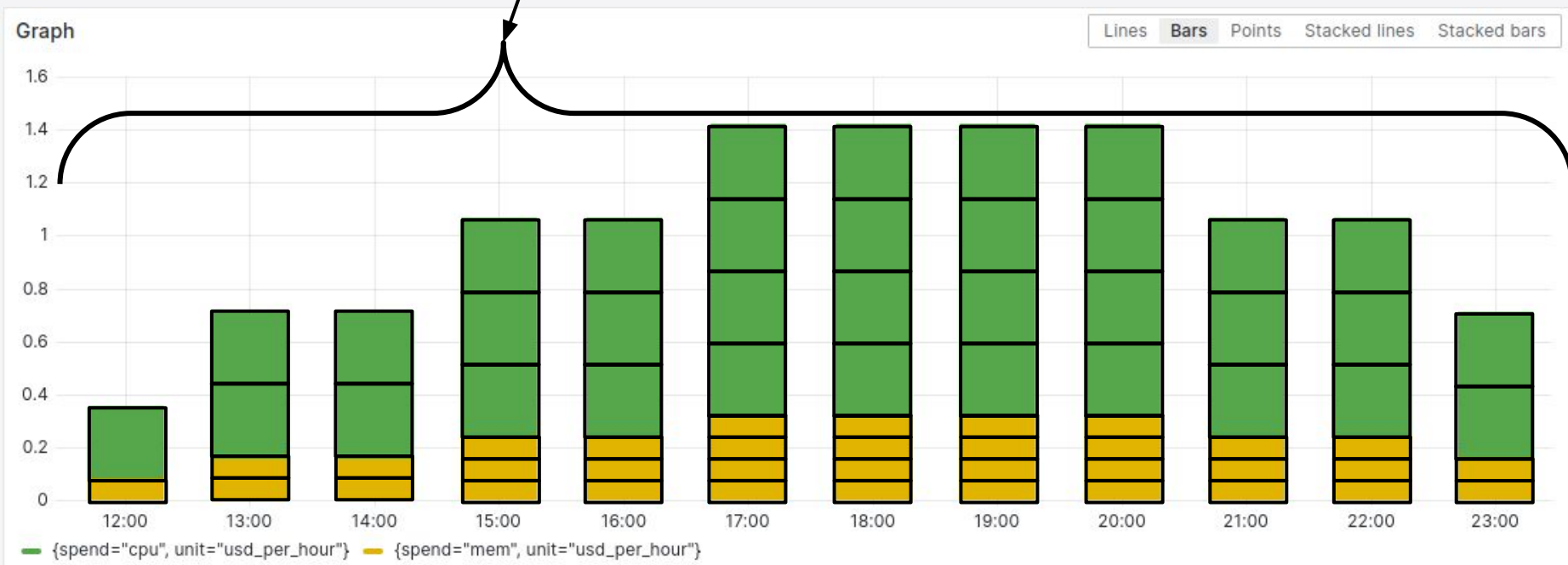
A simple formula with profound implications ...

$$\text{Spend} = \text{Usage} \times \text{Rate}$$



A simple formula with profound implications ...

$$\text{Spend} = \sum \text{Usage} \times \text{Rate} \Delta t$$

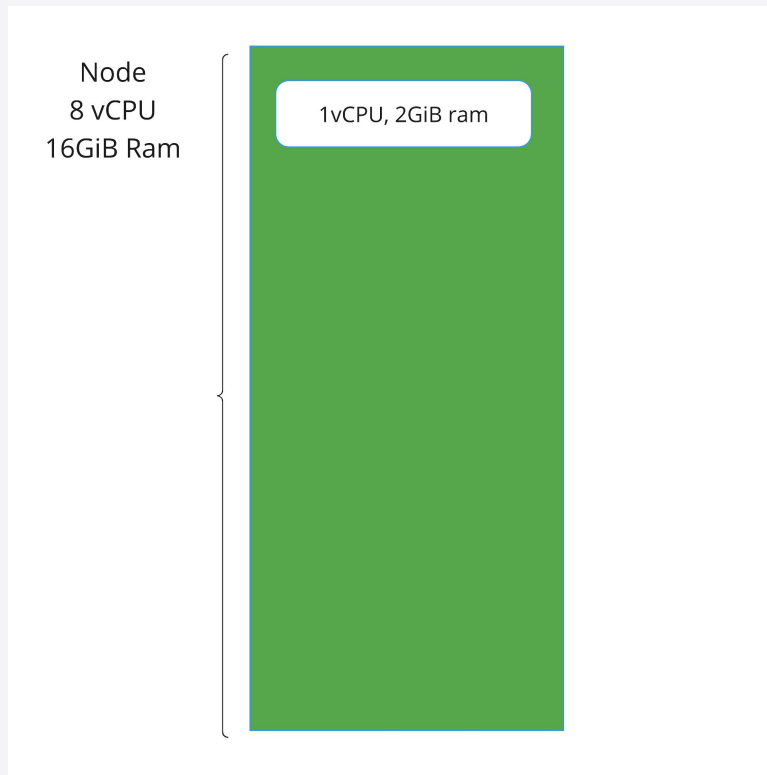


What drives k8s costs

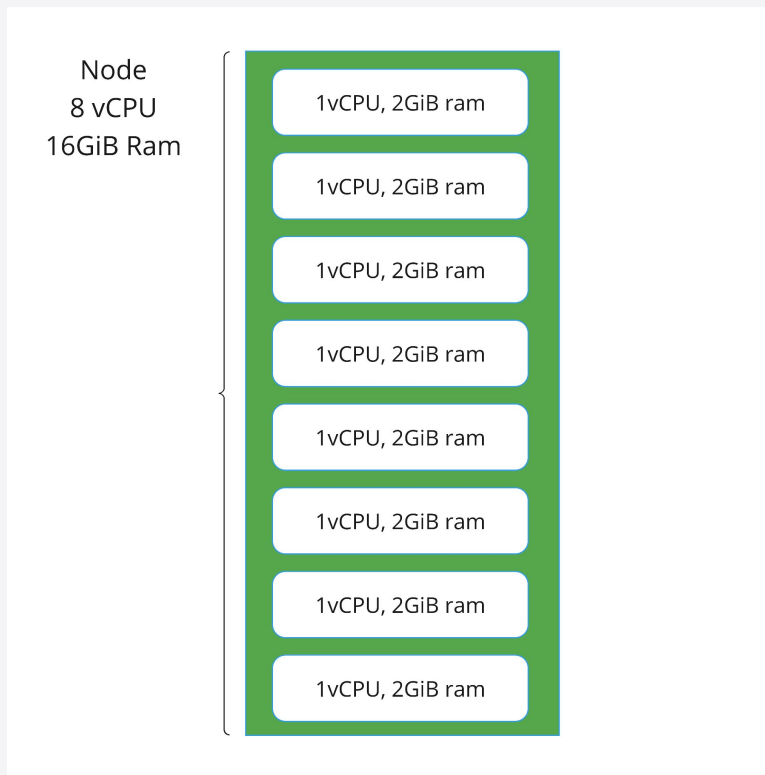
Web service

1vCPU, 2GiB ram

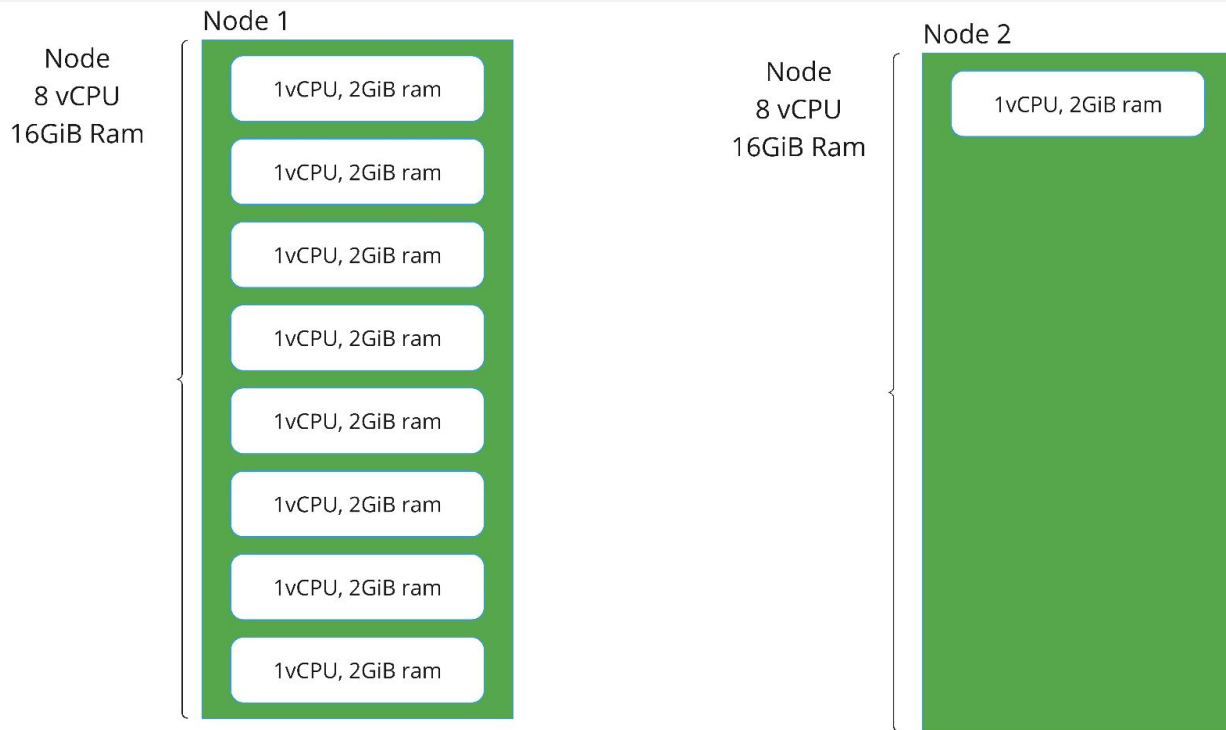
What drives k8s costs



What drives k8s costs



What drives k8s costs



How to measure usage



cpu|memory of nodes



cpu|memory requests of workloads



How to measure usage



cpu|memory of nodes

`kube_node_status_capacity{cluster, resource, node} = float`



cpu|memory requests of workloads

`kube_pod_container_resource_requests{cluster, resource, node, namespace} = float`



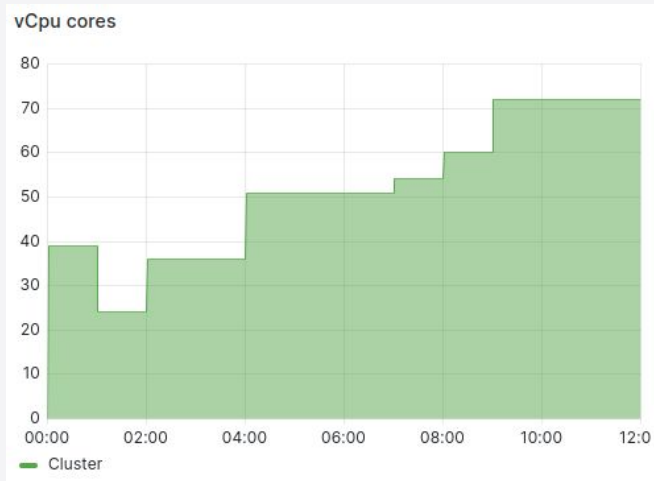
How to measure your nodes

```
sum (  
  usage  
  *  
  rate  
)
```



How to measure your nodes

```
sum (  
  kube_node_status_capacity(resource="cpu")  
  *  
  rate  
)
```



How to measure your nodes

N2 machine types

South Carolina (us-east1) ▼

Hourly ☒ Monthly

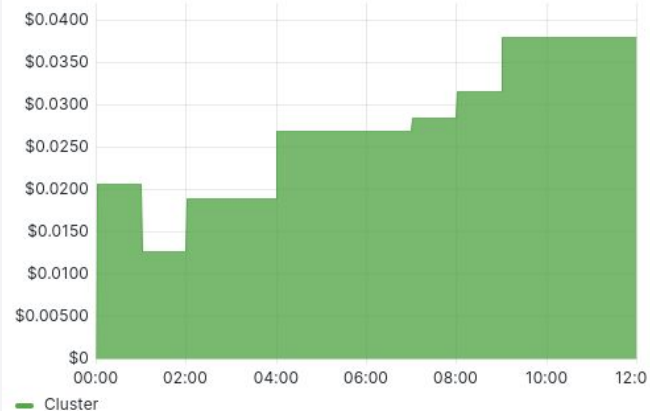
Item	On-demand price (USD)	Spot price* (USD)
Predefined vCPUs	\$0.031611 / vCPU hour	\$0.00836 / vCPU hour
Predefined Memory	\$0.004237 / GB hour	\$0.00112 / GB hour



How to measure your nodes

```
sum (  
  kube_node_status_capacity(resource="cpu")  
  *  
  (0.031611 / 60)  
)
```

Cost in \$ per minute

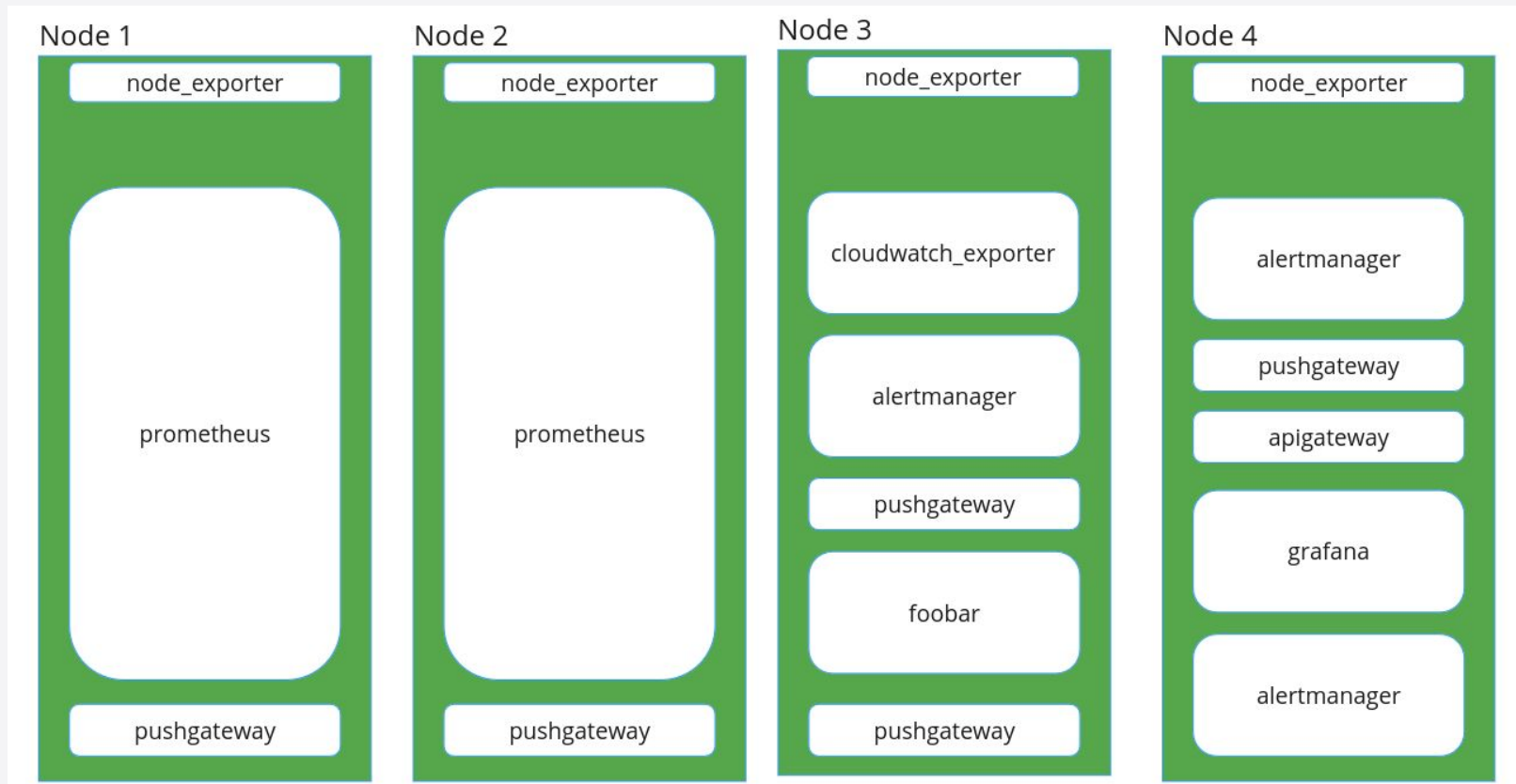


How to measure your nodes

```
- record: cluster:cost_per_minute:sum
  expr: |
    sum by (cluster) (
      kube_node_status_capacity(resource="cpu")
      *
      (0.031611 / 60)
    )
  labels:
    resource: "cpu"
```



What drives k8s costs (or who)



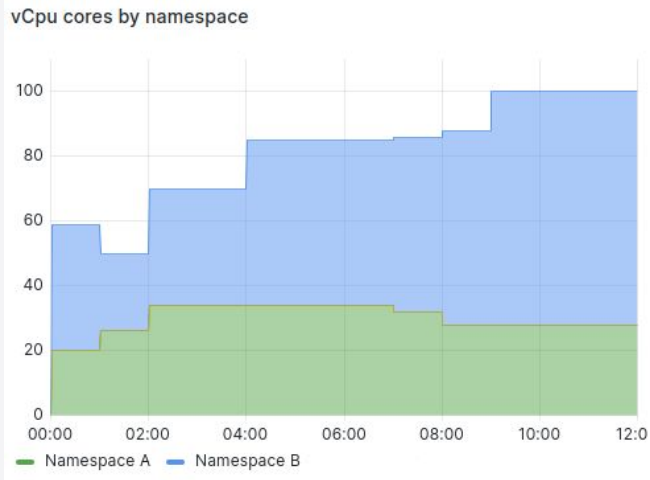
How to measure your workloads impact

```
sum by (namespace) (  
    requests  
    *  
    rate  
)
```



How to measure your workloads impact

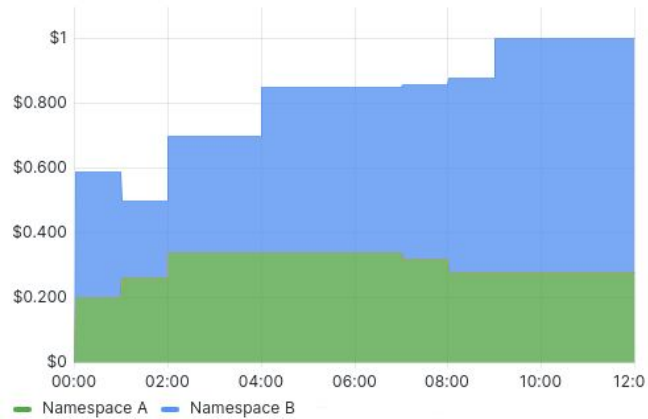
```
sum by (namespace) (  
  kube_pod_container_resource_requests{resource="cpu"}  
  *  
  rate  
)
```



How to measure your workloads impact

```
sum by (namespace) (  
    kube_pod_container_resource_requests{resource="cpu"}  
    *  
    (0.031611 / 60)  
)
```

Cost in \$ per minute by namespace

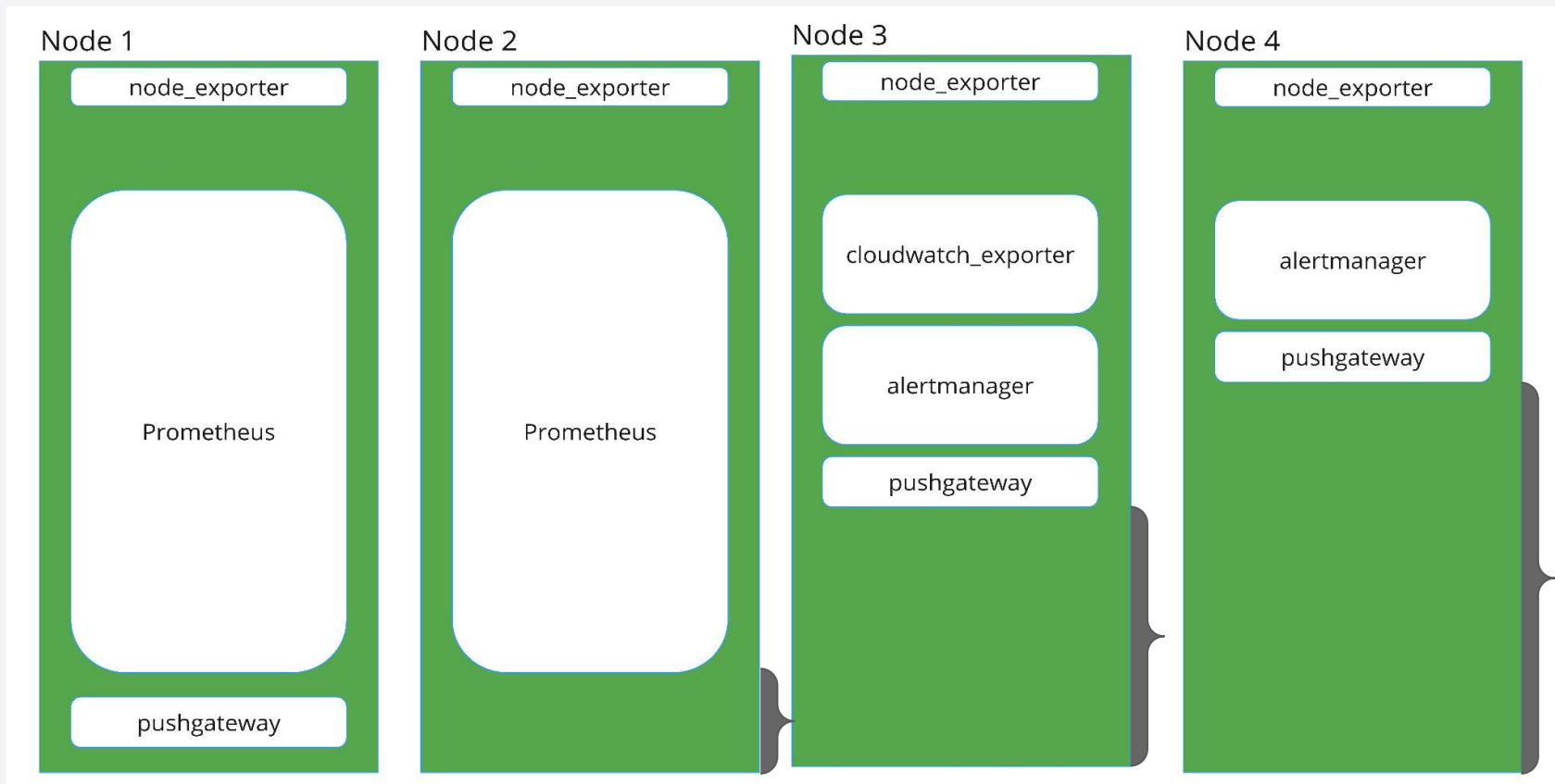


How to measure your workloads impact

```
- record: cluster_namespace:cost_per_minute:sum
  expr: |
    sum by (cluster, namespace) (
      kube_pod_container_resource_requests{resource="cpu"}
      *
      (0.031611 / 60)
    )
  labels:
    resource: "cpu"
```



What drives k8s costs (realistic)



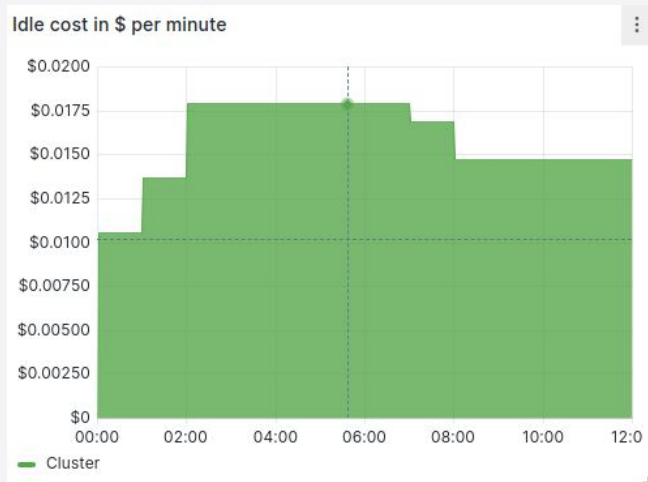
How to measure idle resources

```
sum (  
  (  
    capacity  
    -  
    requests  
  )  
  *  
  rate  
)
```



How to measure idle resources

```
sum (
  (
    sum by (node) (
      kube_node_status_capacity{resource="cpu"}
    )
    -
    sum by (node) (
      kube_pod_container_resource_requests{resource="cpu"}
    )
  )
  *
  (0.031611 / 60)
)
```







How to measure idle resources

```
- record: cluster_namespace:cost_per_minute:sum
  expr: |
    sum by (cluster) (
      (
        sum by (cluster, node) (
          kube_node_status_capacity{resource="cpu"}
        )
        -
        sum by (cluster, node) (
          kube_pod_container_resource_requests{resource="cpu"}
        )
      )
      *
      (0.031611 / 60)
    )
  labels:
    resource: "cpu"
    namespace: "__idle__"
```



Shortcomings

-  This approach only works for homogeneous clusters
-  Takes only compute resources into account
-  Not all CSPs will give you the breakdown on compute resources costs
-  Doesn't reflect savings plans



How to get the rate of your resources

- All major CSPs have pricing APIs
- Not designed for simplicity and ease of use
- Not a metric
- Tools exist to use these APIs
- We looked for a plain metric exporter



We need a plain metric

```
- record: cluster_namespace:cost_per_minute:sum
  expr: |
    sum by (cluster, namespace) (
      kube_pod_container_resource_requests{resource="cpu"}
      *
      rate
    )
  labels:
    resource: "cpu"
```



cloudcost-exporter Public

Edit Pins

Unwatch 105

Fork 0

Starred 16

main 19 Branches 28 Tags

Go to file

Add file

<> Code

About

Prometheus Exporter for Cloud Provider agnostic cost metrics

Readme
Apache-2.0 license
Code of conduct
Security policy
Activity
Custom properties
16 stars
105 watching
0 forks
Report repository















Releases

28 tags

[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

 dependabot[bot]	chore(deps): Bump github.com/prometheus/client_golang f...	ad5c30a · last week	180 Commits
 .github	chore(go): Upgrade to latest version of go (#113)		7 months ago
 cmd/exporter	initial stab into a readiness check (#237)		2 months ago
 docs	feat(ec2): Add name label to exported metrics (#286)		2 weeks ago
 mocks/pkg	Add EBS cost metrics (#240)		2 months ago
 pkg	chore(deps): Bump github.com/prometheus/client_golang...		last week
 scripts	feat(eks): Implement base eks module (#169)		4 months ago
 .gitignore	Azure Provider Skeleton (#199)		3 months ago
 .golangci.yml	Azure: Reconfigure Collect method in AKS to prevent dea...		2 months ago
 .mockery.yaml	refactor(GCS): Migrate to using mockery/testify (#25)		10 months ago
 CODEOWNERS	ci(CODEOWNERS): Add default CODEOWNERS file (#265)		2 months ago
 CODE_OF_CONDUCT.md	docs: Add necessary files to go opensource (#133)		6 months ago
 CONTRIBUTING.md	docs: Add necessary files to go opensource (#133)		6 months ago
 Dockerfile	chore(deps): Bump golang from 1.22.5 to 1.23.0 (#279)		3 weeks ago



Design

$$\text{Spend} = \text{Usage} \times \text{Rate}$$

- Stackdriver exporter
- YACE
- ...

- Cloudcost Exporter provides metrics per provider and resources
 - GCP
 - GKE
 - `cloudcost_gcp_gke_instance_cpu_usd_per_core_hour`
 - `cloudcost_gcp_gke_compute_instance_memory_usd_per_gib_hour`
 - AWS
 - EC2
 - `cloudcost_aws_ec2_instance_cpu_usd_per_core_hour`
 - `cloudcost_aws_ec2_instance_memory_usd_per_gib_hour`
 - S3
 - `cloudcost_aws_s3_storage_by_location_usd_per_gibyte_hour`
 - `cloudcost_aws_s3_operation_by_location_usd_per_krequest`
 - Azure
 - AKS
 - `cloudcost_azure_aks_instance_cpu_usd_per_core_hour`
 - `cloudcost_azure_aks_instance_memory_usd_per_gib_hour`



Final result

```
- record: cluster_namespace:cost_per_minute:sum
  expr: |
    sum by (cluster, namespace) (
      kube_pod_container_resource_requests{resource="cpu"}
      *
      on (node) group_left instance_cpu_usd_per_core_hour / 60
    )
  labels:
    resource: "cpu"
```



Thank you

Slides: <https://github.com/the-it/talks>

Cloudcost Export



Meetup Grafana & Friends Hamburg



