KubeCon | CloudNativeCon

North America 2018

Fine-grained Cost Allocation in Multi-tenant Kubernetes Clusters

Yang Guan (Google) *yangguan@google.com*

Multi-tenant cluster has benefits

- Example tenant
  - Team/department within a company
  - Customer of a Saas
  - Application
- Eases cluster management
- Lowers control plane overhead
- Reduces resource fragmentation
- Allows quick provisioning for new tenant

Tenants share common resources
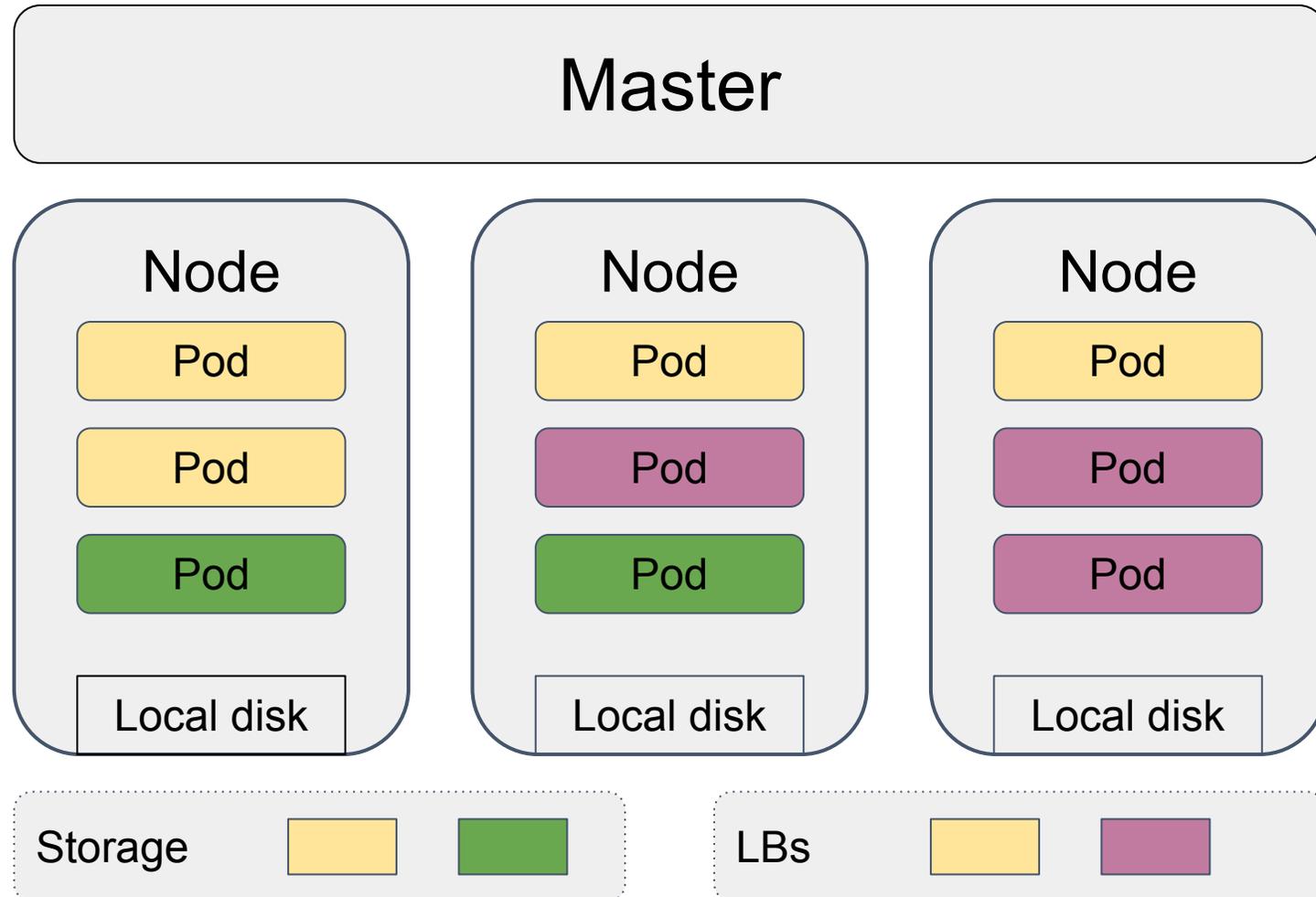
Visibility into usage is challenging, especially in multi-tenant clusters

As a result,

- Hard to know how much of each resource is used, and by whom
- Difficult to understand and predict per-tenant usage/cost trend
- Challenging to pinpoint costs to a specific tenant

Who needs visibility into resource usage?

Finance department	SREs/ Admins	IT Director/ Product manager
Have basis for planning, budgeting and forecasting	Catch unexpected usage/bugs	Understand usage trends and drivers
Perform cost allocation	Monitor network traffic	Optimize resource utilization

Usage metering provides visibility

- Monitor and report resource usage in a Kubernetes cluster
  - CPU, memory, GPU, storage, network traffic, *etc*

- Tie resource usage to Kubernetes concepts
  - pods, persistentVolumeClaims, namespaces, labels, *etc*

Measure usage in Kubernetes
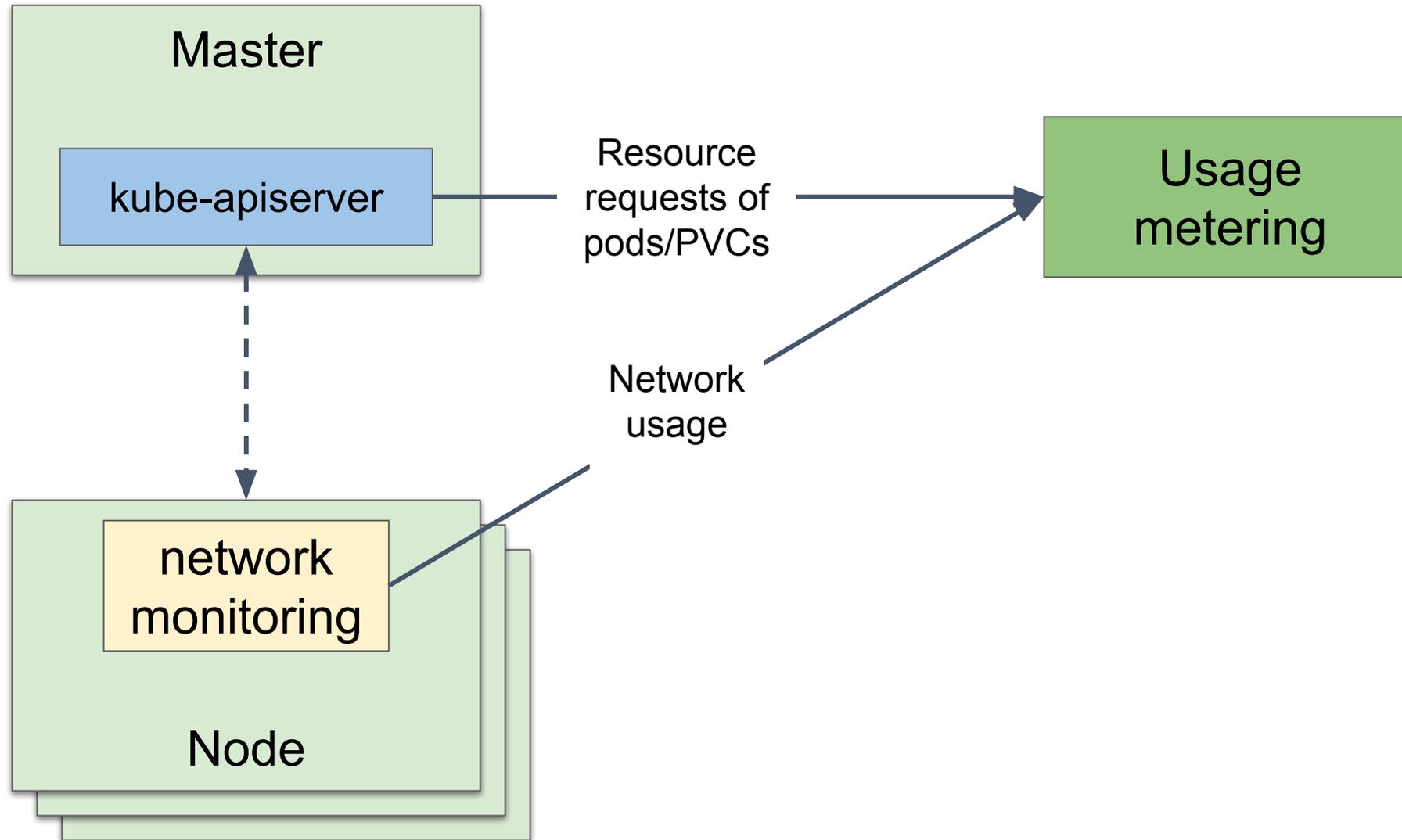
Define "usage" as "request" for CPU/Memory

- kube-scheduler
  - Uses request to decide whether pod fits on node
- cluster autoscaler
  - Adds nodes when there are pending pods
  - Removes nodes when possible to re-pack pods
- ResourceQuota
  - Constrains the aggregated resource requests

Utilization is still useful

- Tracks resources used by BestEffort/Burstable pods
- Allows understanding application behavior
  - For example, changes in tenant usage patterns
- Accounts network usage
  - Kubernetes does not (yet) support network bandwidth as a schedulable resource

Convert usage into cost

- Dedicated resource (e.g., dynamically provisioned PDs)
  - Perform a simple lookup
- Shared resource (e.g., VM instances)
  - Allocate cost proportionally
  - Example:
    - Pod requests 200 milli-CPU and runs 60 minutes
    - Cost of a single-core VM is $1 per hour
    - CPU cost for the pod == $0.20

GKE Usage Metering

- Currently in Beta
- Enable/disable via `gcloud`
- Features
  - CPU, memory, dynamically provisioned disks, GPUs, and network egress
  - Reports usage by Kubernetes namespaces + labels

System Architecture
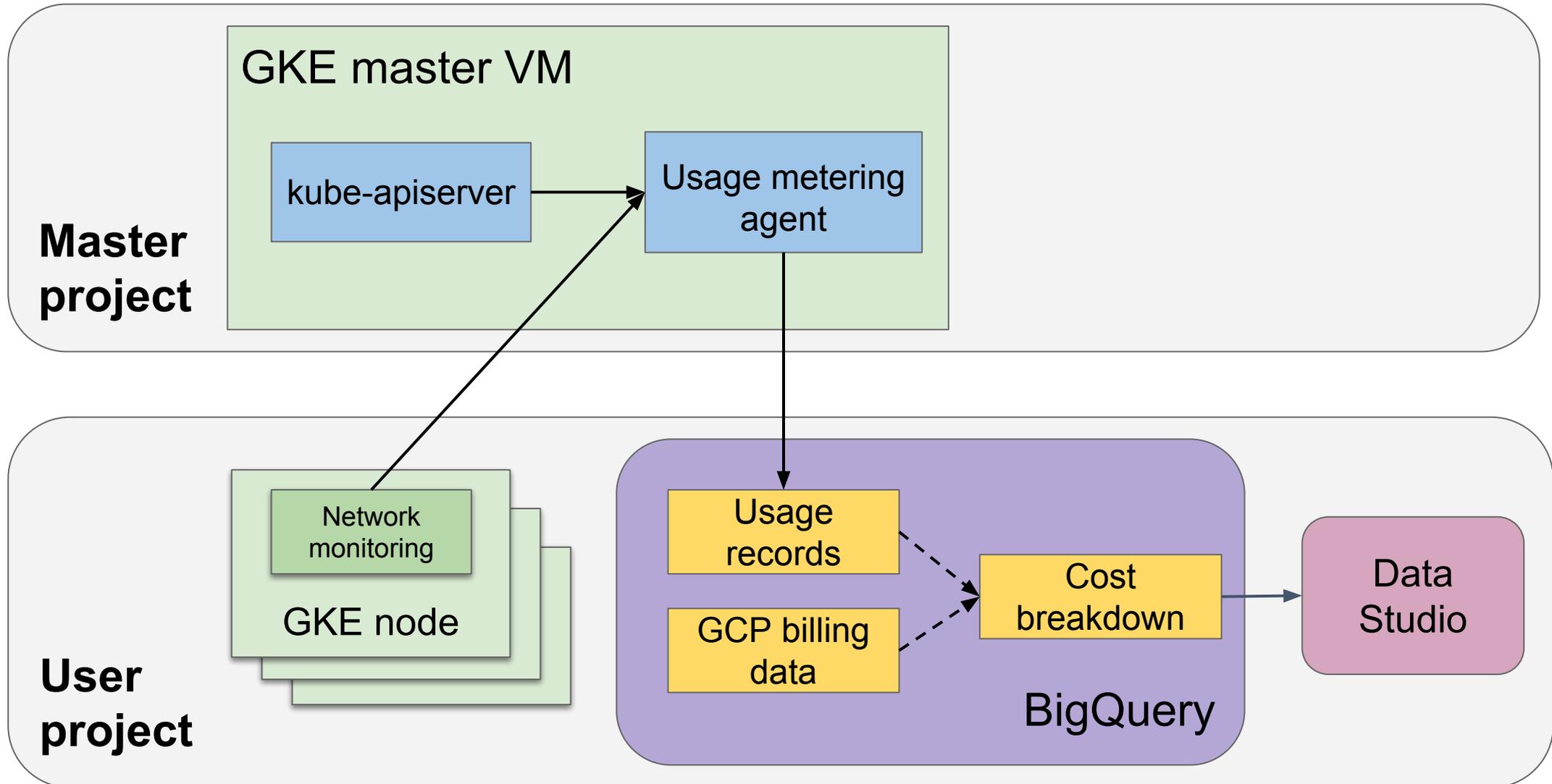
Demo

Conclusion

- Multi-tenancy has many benefits
  - Eases cluster management and lowers cost
- Per-tenant usage metering has many use cases
  - Resource usage monitoring/forecasting
  - Anomalous tenant behavior detection
  - Cost allocation
- Available in Beta on GKE
  - [Documentation](#)

KubeCon | CloudNativeCon

North America 2018