Greenplum on Kubernetes

容器化MPP数据库

AGENDA

云数据库背景
云数据库实现方案
Greenplum on Kubernetes
Greenplum Operator
总结

云数据库背景

云数据库背景

- 资源变化
  - 本地资源 → 云
  - 静态资源 → 弹性需求

- 数据变化
  - 内部数据 → 多数据源
  - 数据规模 → 不易预测
  - 数据格式 → 半结构化/无模式
  - 数据隔离 → 数据共享

STAMFORD, Conn., July 1, 2019

GREENPLUM DATABASE

Gartner Says the Future of the Database Market Is the Cloud

On-Premises DBMS Revenue Continues to Decrease as DBMS Market Shifts to the Cloud

By 2022, 75% of all databases will be deployed or migrated to a cloud platform, with only 5% ever considered for repatriation to on-premises, according to Gartner, Inc. This trend will largely be due to databases used for analytics, and the SaaS model.

"According to inquiries with Gartner clients, organizations are developing and deploying new applications in the cloud and moving existing assets at an increasing rate, and we believe this will continue to increase," said Donald Feinberg, distinguished research vice president at Gartner. "We also believe this begins with systems for data management solutions for analytics (DMSA) use cases — such as data warehousing, data lakes and other use cases where data is used for analytics, artificial intelligence (AI) and machine learning (ML). Increasingly, operational systems are also moving to the cloud, especially with conversion to the SaaS application model."

Gartner research shows that 2018 worldwide database management system (DBMS) revenue grew 18.4% to $46 billion. Cloud DBMS revenue accounts for 68% of that 18.4% growth — and Microsoft and Amazon Web Services (AWS) account for 75.5% of the total market growth. This trend reinforces that cloud service provider (CSP) infrastructures and the services that run on them are becoming the new data management platform.

Ecosystems are forming around CSPs that both integrate services within a single CSP and provide early steps toward intercloud data management. This is in distinct contrast to the on-premises approach, where individual products often serve multiple roles but rarely offer their own built-in capabilities to support integration with adjacent products within the on-premises deployment environment. While there is some growth in on-premises systems, this growth is rarely from new on-premises deployments; it is generally due to price increases and forced upgrades undertaken to avoid risk.

"Ultimately what this shows is that the prominence of the CSP infrastructure, its native offerings, and the third-party offerings that run on them is assured," said Mr. Feinberg. "A recent Gartner cloud adoption survey showed that of those on the public cloud, 81% were using more than one CSP. The cloud ecosystem is expanding beyond the scope of a single CSP — to multiple CSPs — for most cloud consumers."

- 云数据库市场巨大
- 云数据库增速巨大
- DBasS的需求
- 跨云的需求

云数据库实现方案

云数据库需求

- DBasS
  - 自动化运维
  - 自动化调优
- 弹性资源管理
  - 存储资源
  - 计算资源
- 安全
  - 用户数据
  - 临时文件
  - 网络传输
  - 权限控制
- 跨云
  - 公有云
  - 私有云

云数据库实现方案

GREENPLUM DATABASE

- 全新数据库
  - Snowflake

- 原有数据库架构升级
  - Vertica Eon Mode

- 容器化数据库+Kubernetes
  - Apache Spark
  - CockroachDB
  - Apache HAWQ

云数据库存储方案

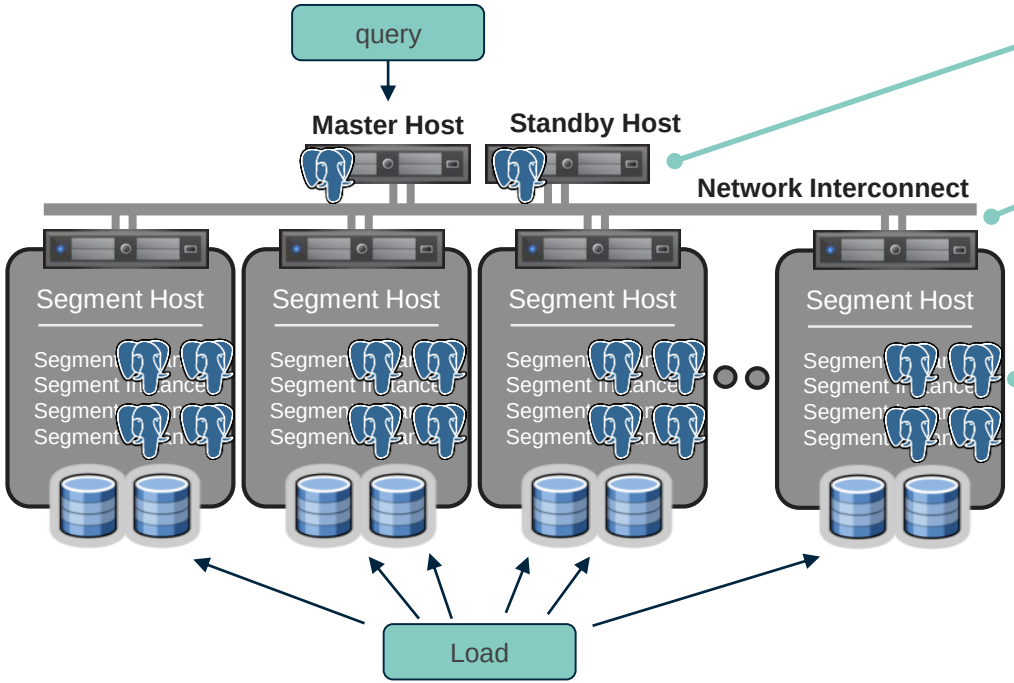
GREENPLUM DATABASE

- 块存储
  - **文件系**统接口

- 对象存储
  - **成本低**
  - 扩展性强
  - 访问延迟高

Greenplum on Kubernetes

Greenplum 架构



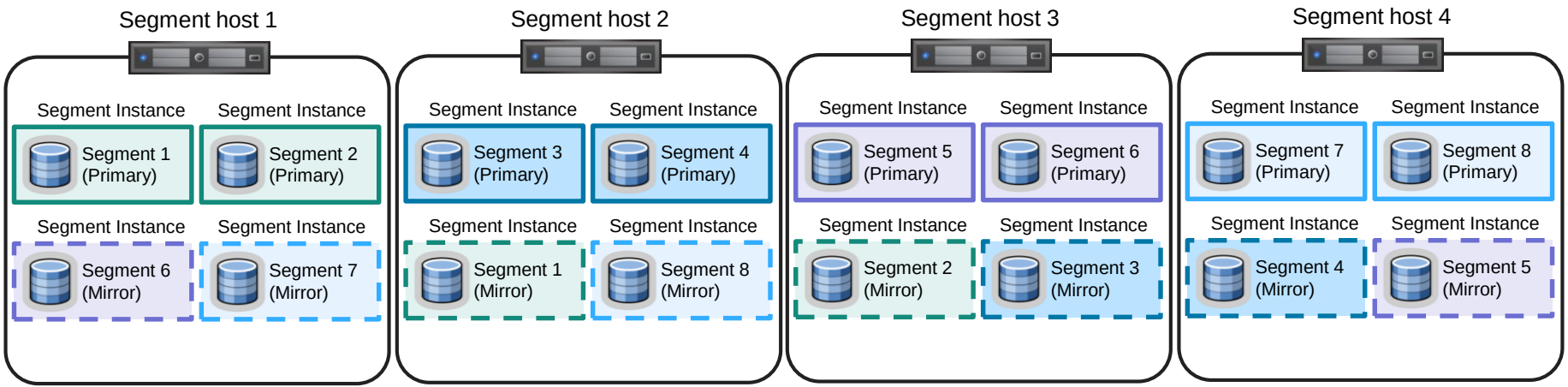
Master节点和Standby Master节点

Interconnect高速网络

Segment主机部署多个Segment实例
（Primary Segment和Mirror Segment）

Greenplum 部署方案

- Master节点和Standby Master分机部署
- Primary Segment节点和Mirror Segment节点分机部署
- Mirroring部署方案



容器化Greenplum

GREENPLUM DATABASE®

GREENPLUM DATABASE®

+

docker

=

?

容器化Greenplum

- 容器粒度
  - Segment主机 VS. Segment实例
- 容器资源分配
  - CPU
  - 内存
  - 磁盘
- 容器间网络互联
  - 本机网络
  - 跨机网络

- 容器化Greenplum部署策略
  - Master部署策略
  - Primary Segment部署策略
  - Mirror Segment部署策略
- 容器化Greenplum运维管理
  - 故障检测及恢复
  - 升级扩容
- 容器化Greenplum存储管理
  - 容器本地存储易失性

容器化Greenplum

GREENPLUM DATABASE

- 容器粒度
  - Segment主机 VS. Segment实例
- 容器资源分配
  - CPU
  - 内存
  - 磁盘
- 容器间网络互联
  - 本机网络
  - 跨机网络

容器镜像管理

容器调度

容器资源管理

容器监控及自定义操作

容器网络管理

容器存储管理

- 容器化Greenplum部署策略
  - Master部署策略
  - Primary Segment部署策略
  - Mirror Segment部署策略
- 容器化Greenplum运维管理
  - 故障检测及恢复
  - 升级扩容
- 容器化Greenplum存储管理
  - 容器本地存储易失性
  - 容器外部存储关联性

容器化分布式应用程序公共问题

容器化Greenplum

- 容器粒度
  - Segment主机 VS. Segment实例
- 容器资源分配
  - CPU
  - 内存
  - 磁盘
- 容器间网络互联
  - 本机网络
  - 跨机网络

容器镜像管理

容器调度

容器资源管理

容器监控及自定义操作

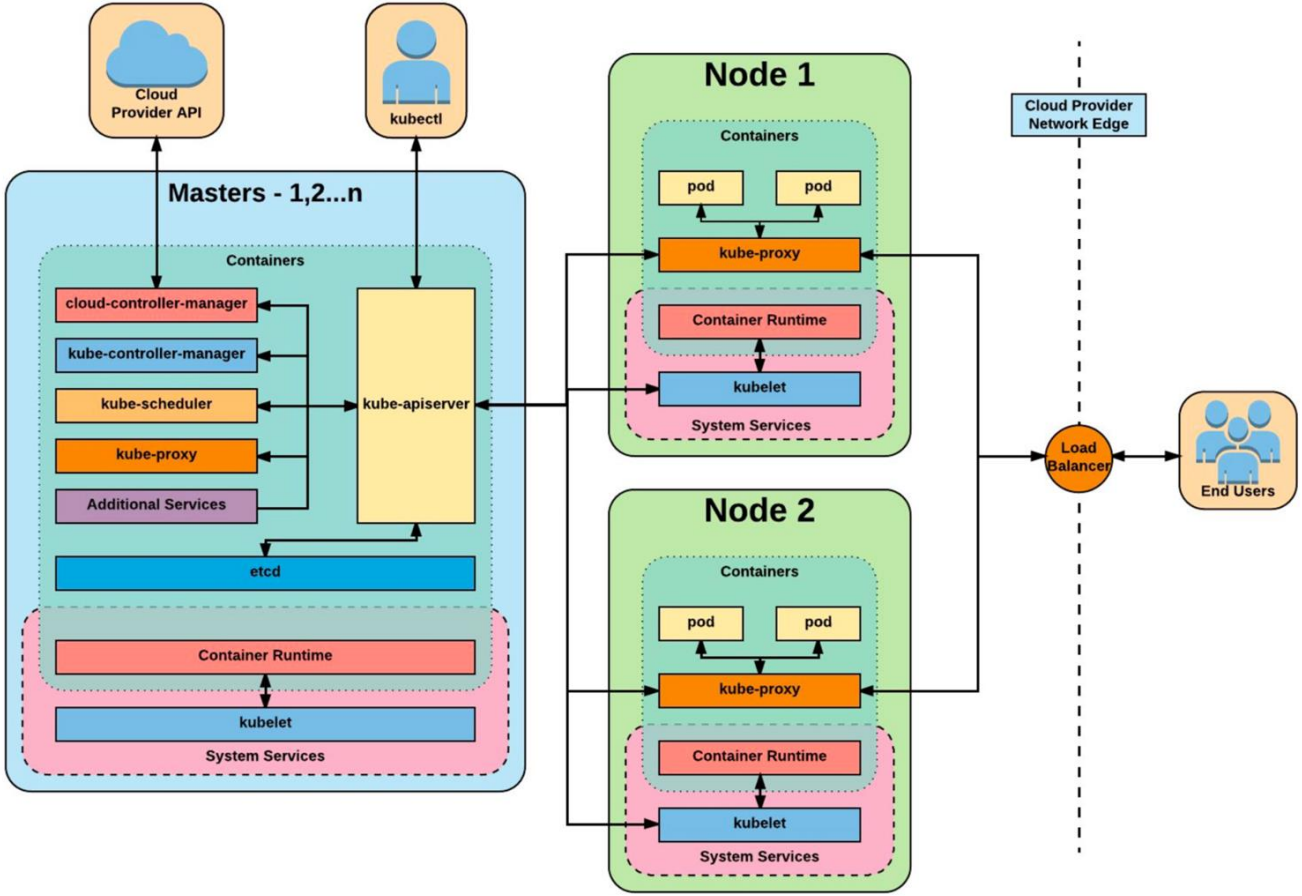
容器网络管理

容器存储管理

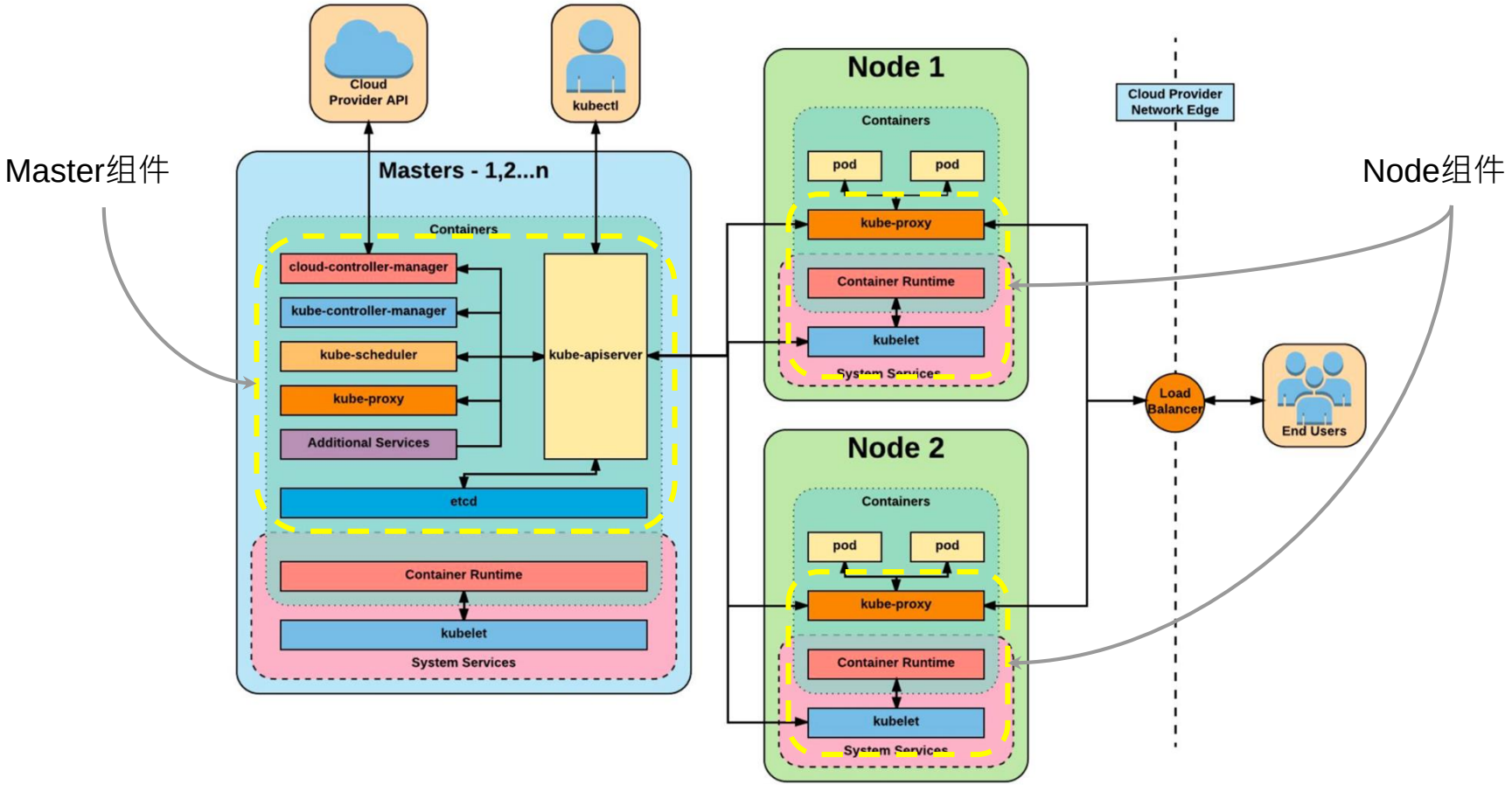
- 容器化Greenplum部署策略
  - Master部署策略
  - Primary Segment部署策略
  - Mirror Segment部署策略
- 容器化Greenplum运维管理
  - 故障检测及恢复
  - 升级扩容
- 容器化Greenplum存储管理
  - 容器本地存储易失性
  - 容器外部存储关联性

Kubernetes

Kubernetes 101



Kubernetes 101



Kubernetes存储资源 PV

- PersistentVolume
  - 表示一种存储资源，独立容器生命周期
  - AWSElasticBlockStore,
    AzureDisk,
    AzureFile,
    GCEPersistentDisk,
    CephFS,
    Glusterfs,
    NFS,
    Cinder,
    VsphereVolume
- PersistentVolumeClaim
  - 申请存储资源

```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv0003
spec:
  capacity:
    storage: 5Gi
  volumeMode: Filesystem
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Recycle
  storageClassName: slow
  mountOptions:
    - hard
    - nfsvers=4.1
  nfs:
    path: /tmp
    server: 172.17.0.2
```

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: myclaim
spec:
  accessModes:
    - ReadWriteOnce
  volumeMode: Filesystem
  resources:
    requests:
      storage: 8Gi
  storageClassName: slow
  selector:
    matchLabels:
      release: "stable"
    matchExpressions:
      - {key: environment, operator: In, values: [dev]}
```

Kubernetes 计算资源 Pod

- Pod
  - 计算任务 → 容器 → Pod
  - 资源分配：CPU， 内存， 磁盘
  - 资源调度：Pod → Node
- Pod管理
  - 无状态计算资源组：Deployment
  - 有状态计算资源组：StatefulSet
- Pod持久存储
  - 通过PVC申请PV存储资源
- StatefulSet
  - Pod网络地址不变
  - Pod与PV映射关系不变

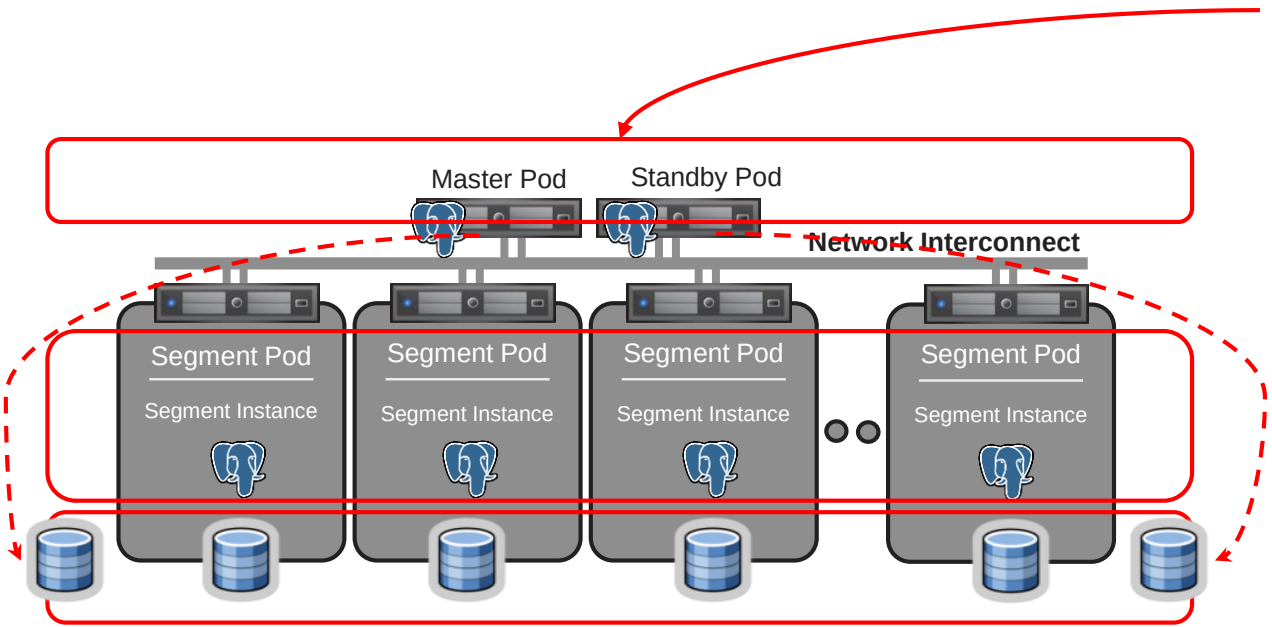
```yaml
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: web
spec:
  selector:
    matchLabels:
      app: nginx # has to match .spec.template.metadata.labels
  serviceName: "nginx"
  replicas: 3 # by default is 1
  template:
    metadata:
      labels:
        app: nginx # has to match .spec.selector.matchLabels
    spec:
      terminationGracePeriodSeconds: 10
      containers:
      - name: nginx
        image: k8s.gcr.io/nginx-slim:0.8
        ports:
        - containerPort: 80
          name: web
        volumeMounts:
        - name: www
          mountPath: /usr/share/nginx/html
  volumeClaimTemplates:
  - metadata:
      name: www
    spec:
      accessModes: [ "ReadWriteOnce" ]
      storageClassName: "my-storage-class"
      resources:
        requests:
          storage: 1Gi
```

Kubernetes 网络资源 Service

- Service
  - **定**义统一网络地址
  - **分布式**应用程序路由映射
  - 负载均衡器

```yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  ports:
  - port: 80
    name: web
  clusterIP: None
  selector:
    app: nginx
```

Greenplum on Kubernetes



Database Service (Active Master)

Master StatefulSet (2)

Primary Segment StatefulSet (N)
Mirror Segment StatefulSet (N)

Cluster PV (N)

Greenplum on Kubernetes

GREENPLUM DATABASE

Master节点示例

```
Name:           master-0
Namespace:      default
Node:           minikube/192.168.64.2
Labels:         app=greenplum
                greenplum-cluster=my-greenplum
                statefulset.kubernetes.io/pod-name=master-0
                type=master
Status:         Running
IP:             172.17.0.8
Controlled By:  StatefulSet/master
Containers:
  greenplum:
    Container ID:   docker://9d7bc3bd31823ba0357b13b7b92aae346340c87460878ef001044ad0a0859183
    Image:          greenplum-for-kubernetes:v1.4.0
    Image ID:       docker://sha256:8a0c551661e51bd0e6fd6e7a59f7b214a692e47f022dcf6c5ef0627da31d8443
    Port:           22/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Tue, 13 Aug 2019 17:07:12 +0800
    Ready:          True
    Limits:
      cpu:     500m
      memory:  800Mi
    Requests:
      cpu:     500m
      memory:  800Mi
    Environment:
      MASTER_DATA_DIRECTORY:  /greenplum/data-1
    Mounts:
      /etc/config from config-volume (rw)
      /etc/ssh-key from ssh-key-volume (rw)
      /greenplum from my-greenplum-pgdata (rw)
Volumes:
  my-greenplum-pgdata:
    Type:       PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName:  my-greenplum-pgdata-master-0
    ReadOnly:   false
  ssh-key-volume:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  ssh-secrets
  config-volume:
    Type:      ConfigMap (a volume populated by a ConfigMap)
    Name:      greenplum-config
```

Segment节点示例

```
Name:           segment-a-0
Namespace:      default
Node:           minikube/192.168.64.2
Labels:         app=greenplum
                greenplum-cluster=my-greenplum
                statefulset.kubernetes.io/pod-name=segment-a-0
                type=segment-a
Status:         Running
IP:             172.17.0.6
Controlled By:  StatefulSet/segment-a
Containers:
  greenplum:
    Container ID:   docker://aad0ef3200a3e8115c5672707cedcc3b4a2a0d12cf345bfe829273829c9e10c1
    Image:          greenplum-for-kubernetes:v1.4.0
    Image ID:       docker://sha256:8a0c551661e51bd0e6fd6e7a59f7b214a692e47f022dcf6c5ef0627da31d8443
    Port:           22/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Tue, 13 Aug 2019 17:07:11 +0800
    Ready:          True
    Limits:
      cpu:     500m
      memory:  800Mi
    Requests:
      cpu:     500m
      memory:  800Mi
    Environment:
      MASTER_DATA_DIRECTORY:  /greenplum/data-1
    Mounts:
      /etc/config from config-volume (rw)
      /etc/ssh-key from ssh-key-volume (rw)
      /greenplum from my-greenplum-pgdata (rw)
Volumes:
  my-greenplum-pgdata:
    Type:       PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName:  my-greenplum-pgdata-segment-a-0
    ReadOnly:   false
  ssh-key-volume:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  ssh-secrets
  config-volume:
    Type:      ConfigMap (a volume populated by a ConfigMap)
    Name:      greenplum-config
```

Greenplum on Kubernetes

```
iweng@Ivan-Wengs-MacBook-Pro  ~/workspace  kubectl get service  greenplum -o wide
NAME        TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE   SELECTOR
greenplum   LoadBalancer   10.99.192.32   <pending>     5432:32551/TCP   34m   statefulset.kubernetes.io/pod-name=master-0
```

```
iweng@Ivan-Wengs-MacBook-Pro  ~/workspace  kubectl get statefulsets -o wide
NAME        READY   AGE   CONTAINERS   IMAGES
master      2/2     35m   greenplum    greenplum-for-kubernetes:v1.4.0
segment-a   1/1     35m   greenplum    greenplum-for-kubernetes:v1.4.0
segment-b   1/1     35m   greenplum    greenplum-for-kubernetes:v1.4.0
```

```
iweng@Ivan-Wengs-MacBook-Pro  ~/workspace  kubectl get pods -o wide
NAME                                  READY   STATUS    RESTARTS   AGE    IP           NODE       NOMINATED NODE   READINESS GATES
greenplum-operator-58fdd6dbd-s6brq    1/1     Running   0          120m   172.17.0.5   minikube   <none>           <none>
master-0                              1/1     Running   0          35m    172.17.0.8   minikube   <none>           <none>
master-1                              1/1     Running   0          35m    172.17.0.7   minikube   <none>           <none>
segment-a-0                           1/1     Running   0          35m    172.17.0.6   minikube   <none>           <none>
segment-b-0                           1/1     Running   0          35m    172.17.0.9   minikube   <none>           <none>
```

```
iweng@Ivan-Wengs-MacBook-Pro  ~/workspace  kubectl get pvc -o wide
NAME                                  STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE    VOLUMEMODE
my-greenplum-pgdata-master-0          Bound    pvc-13b79a0c-2b37-403c-a2f2-6212d0861120   1G         RWO            standard       112m   Filesystem
my-greenplum-pgdata-master-1          Bound    pvc-fe5d4905-44c6-452e-b055-48d82a1dda7b   1G         RWO            standard       112m   Filesystem
my-greenplum-pgdata-segment-a-0       Bound    pvc-5b93fddc-1bf5-4253-9430-a0837d58fad3   2G         RWO            standard       112m   Filesystem
my-greenplum-pgdata-segment-b-0       Bound    pvc-444c77ec-d402-4552-82be-423f90862a68   2G         RWO            standard       112m   Filesystem
```

```
iweng@Ivan-Wengs-MacBook-Pro  ~/workspace  kubectl get pv -o wide
NAME                                       CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM                                          STORAGECLASS   REASON   AGE    VOLUMEMODE
pvc-13b79a0c-2b37-403c-a2f2-6212d0861120   1G         RWO            Delete           Bound    default/my-greenplum-pgdata-master-0           standard                110m   Filesystem
pvc-32bae801-e134-4ac0-b172-4b60554fc27b   2G         RWO            Delete           Bound    gpinstance-1/my-greenplum-pgdata-segment-a-0   standard                112m   Filesystem
pvc-444c77ec-d402-4552-82be-423f90862a68   2G         RWO            Delete           Bound    default/my-greenplum-pgdata-segment-b-0        standard                110m   Filesystem
pvc-5b93fddc-1bf5-4253-9430-a0837d58fad3   2G         RWO            Delete           Bound    default/my-greenplum-pgdata-segment-a-0        standard                110m   Filesystem
```

Greenplum on Kubernetes

- 存储计算分离
  - PV持久化存储资源
  - StatefulSet/Pod弹性扩展计算资源
- 数据库服务层
  - Service统一Master & Standby Master地址
- 服务发现机制
  - 所有节点地址名不变
- 跨云能力
  - 容器应用对基础设施透明

Greenplum Operator

Kubernetes Operator

GREENPLUM DATABASE

- 自定义资源类型
  - Custom Resource Definition (CRD)
  - 和内置资源用法一致

- 自定义控制器
  - Custom Controller (Operator)
  - 根据资源状态封装对资源的自定义操作

Greenplum CRD

```yaml
apiVersion: "greenplum.pivotal.io/v1"
kind: "GreenplumCluster"
metadata:
  name: my-greenplum
spec:
  masterAndStandby:
    hostBasedAuthentication: |
      # host   all   gpadmin   1.2.3.4/32   trust
      # host   all   gpuser    0.0.0.0/0   md5
    memory: "800Mi"
    cpu: "0.5"
    storageClassName: standard
    storage: 1G
    antiAffinity: "yes"
    workerSelector: {}
  segments:
    primarySegmentCount: 1
    memory: "800Mi"
    cpu: "0.5"
    storageClassName: standard
    storage: 2G
    antiAffinity: "yes"
    workerSelector: {}
    mirrors: yes
```

```
$ kubectl create -f my-gp-instance.yaml
```

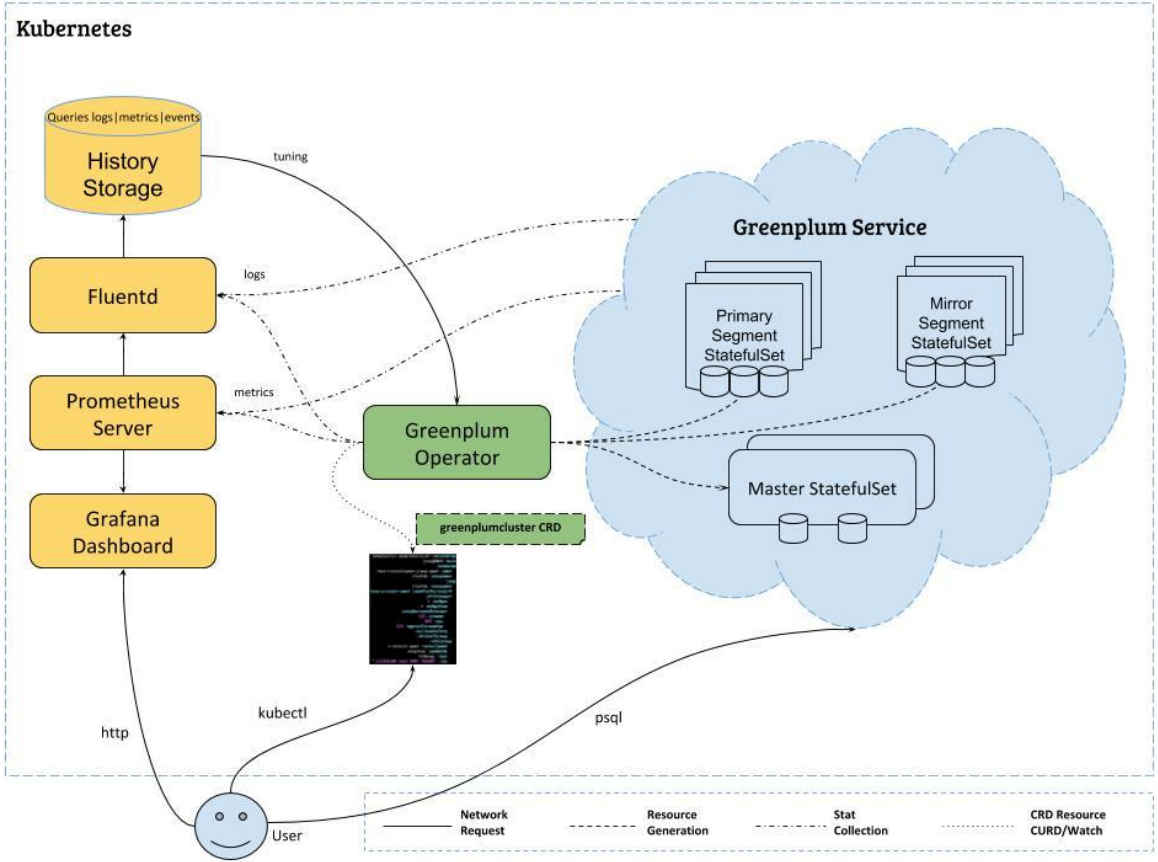
```
iweng@Ivan-Wengs-MacBook-Pro  ~/workspace/gp4k/tools/greenplum-for-kubernetes-v1.4.0/workspace  kubectl describe greenplumcluster my-greenplum
Name:         my-greenplum
Namespace:    default
Labels:       <none>
Annotations:  kubectl.kubernetes.io/last-applied-configuration:
                {"apiVersion":"greenplum.pivotal.io/v1","kind":"GreenplumCluster","metadata":{"annotations":{},"name":"my-greenplum","namespace":"default"...
API Version:  greenplum.pivotal.io/v1
Kind:         GreenplumCluster
Metadata:
  Creation Timestamp:  2019-08-13T09:07:09Z
  Finalizers:
    stopcluster.greenplumcluster.pivotal.io
  Generation:        3
  Resource Version:  10371
  Self Link:         /apis/greenplum.pivotal.io/v1/namespaces/default/greenplumclusters/my-greenplum
  UID:               8335129d-7b6e-4151-a613-fc530ffd3afe
Spec:
  Master And Standby:
    Anti Affinity:              no
    Cpu:                        0.5
    Host Based Authentication:  # host   all   gpadmin   1.2.3.4/32   trust
# host   all   gpuser    0.0.0.0/0   md5

    Memory:              800Mi
    Storage:             1G
    Storage Class Name:  standard
    Worker Selector:
  Segments:
    Anti Affinity:          no
    Cpu:                    0.5
    Memory:                 800Mi
    Primary Segment Count:  1
    Storage:                2G
    Storage Class Name:     standard
    Worker Selector:
Status:
  Instance Image:    greenplum-for-kubernetes:v1.4.0
  Operator Version:  greenplum-operator:v1.4.0
  Phase:             Running
Events:              <none>
```

Greenplum Operator

- Kubernetes Deployment维护Greenplum Operator

- 集群自动部署
  - kube-scheduler → 自定义部署策略
- 集群动态扩容
  - GreenplumCluster → primarySegmentCount参数
- Master节点检测及修复
  - 自动执行gpactivatestandby
  - 重定向Service路由
- Segment节点检查及修复
  - 自动执行gprecoverseg
  - 无需Segment Rebalance
- Node失效
  - Node节点上Master/Segment节点自动修复

Kubernetes 生态集成



- 日志收集
  - Fluentd
- 监控及Metrics收集
  - Prometheus
- 可视化
  - Grafana
- ……

总结

GREENPLUM DATABASE

Greenplum ➔ Kubernetes Native Database

GREENPLUM DATABASE®

扫描加入 Greenplum 技术讨论群

微信群组

QQ 群组

欢迎订阅 Greenplum中文社区：greenplum.cn