Pivotal

When Distributed Database Meets Cloud Lessons Learned

Yanqing Weng

iweng@pivotal.io

© Copyright 2017 Pivotal Software, Inc. All rights Reserved. Version 1.0

About me

- Principal Software Engineer in Pivotal
- Apache HAWQ Committer
- Apache HAWQ PMC Member

Agenda

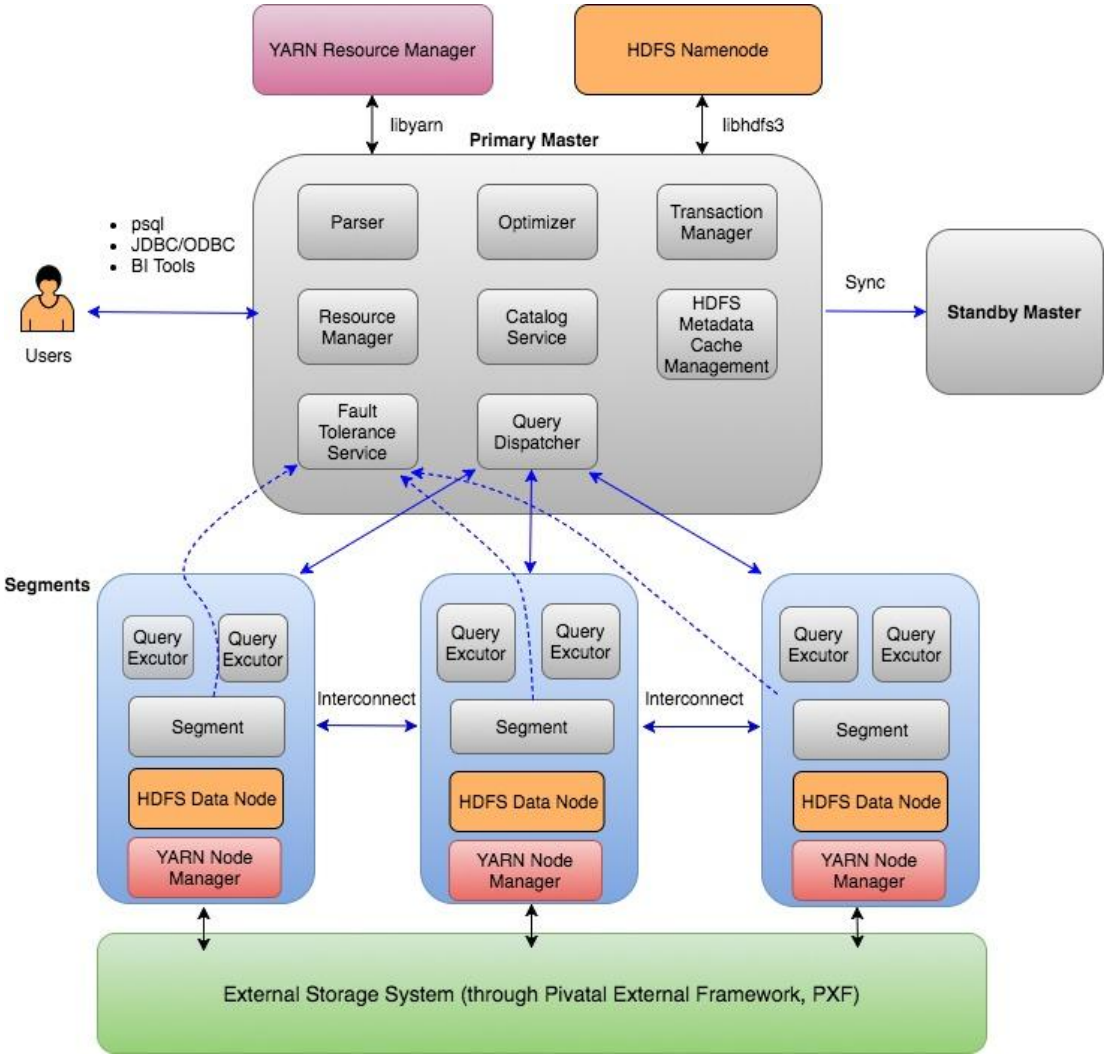
- Introduction to Distributed Database
- Distributed Database on Cloud
- Lessons Learned
- Q & A

Distributed Database

Apache HAWQ

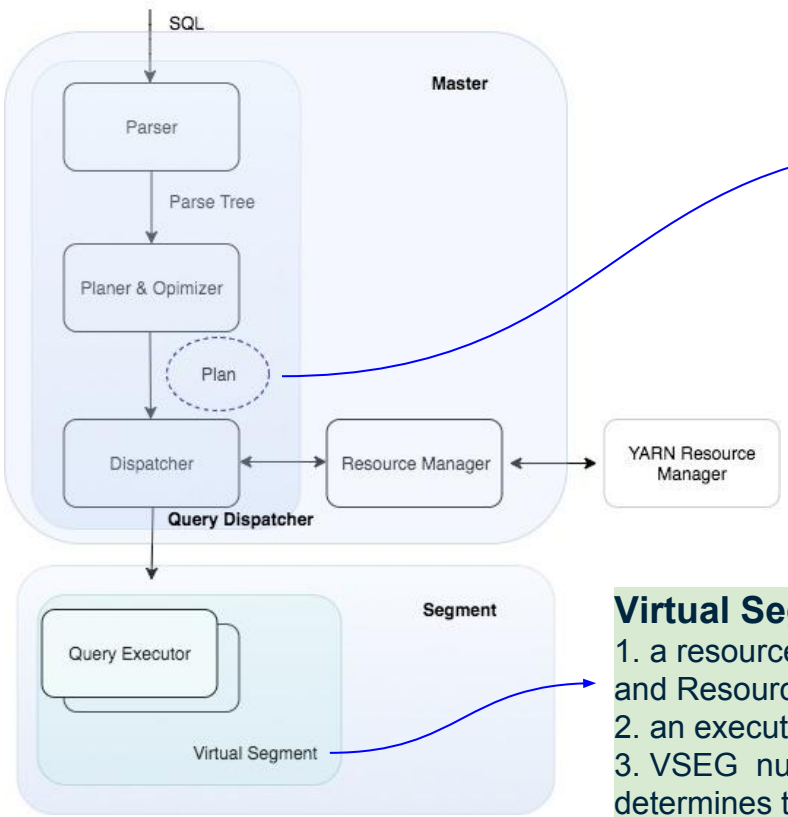
- Apache Hadoop Native SQL, Advanced, MPP, Elastic Query Engine.
- Apache Top Level Project in 2018.8

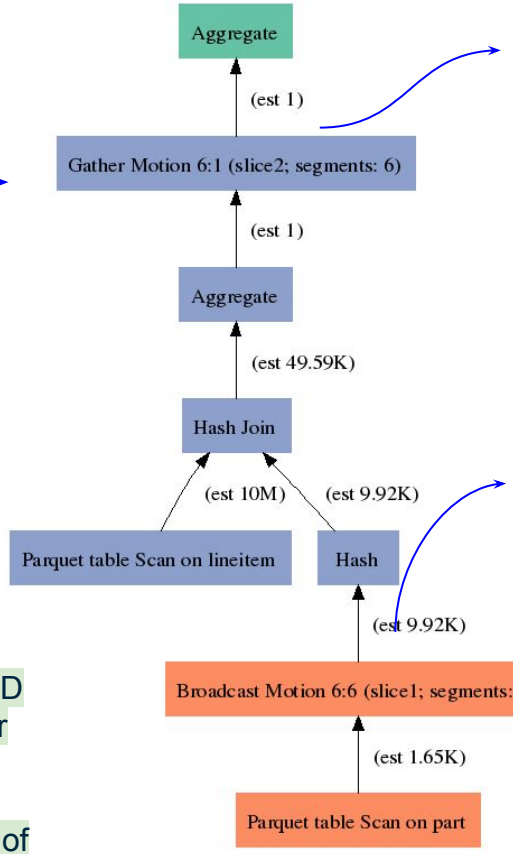
Apache HAWQ Architecture



Pivotal

Apache HAWQ Query Processing





Slice:
1. a portion of the plan that segments can work on independently.
2. a query plan is sliced wherever a motion operation occurs in the plan.

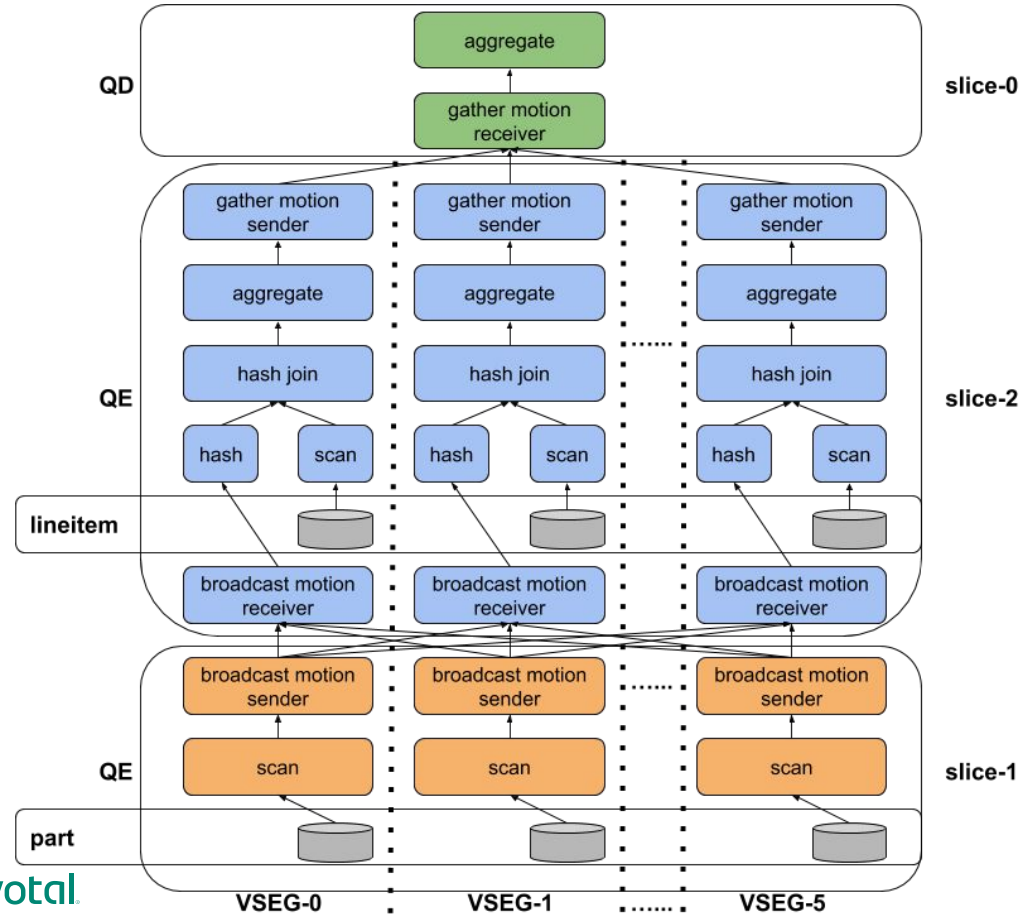
Motion:
1. an operation involves moving tuples between the segments during query processing.
2. three types: redistribution, broadcast, gather motion.

Virtual Segment:
1. a resource unit for QD and Resource Manager
2. an execution unit
3. VSEG number determines the degree of parallelism of a query dynamically.

SELECT COUNT(*) FROM lineitem, part WHERE p_partkey=l_partkey AND p_brand = 'Brand#23'

Pivotal

Virtual Segment



- Resource allocation unit
- Query execution unit
- Variable virtual segment number
- Place on any physical segment

Summary

- High Performance
- Storage computing separation
- Fine-grained resource management
- Elastic query execution engine
- Stateless segment

Cloud Database

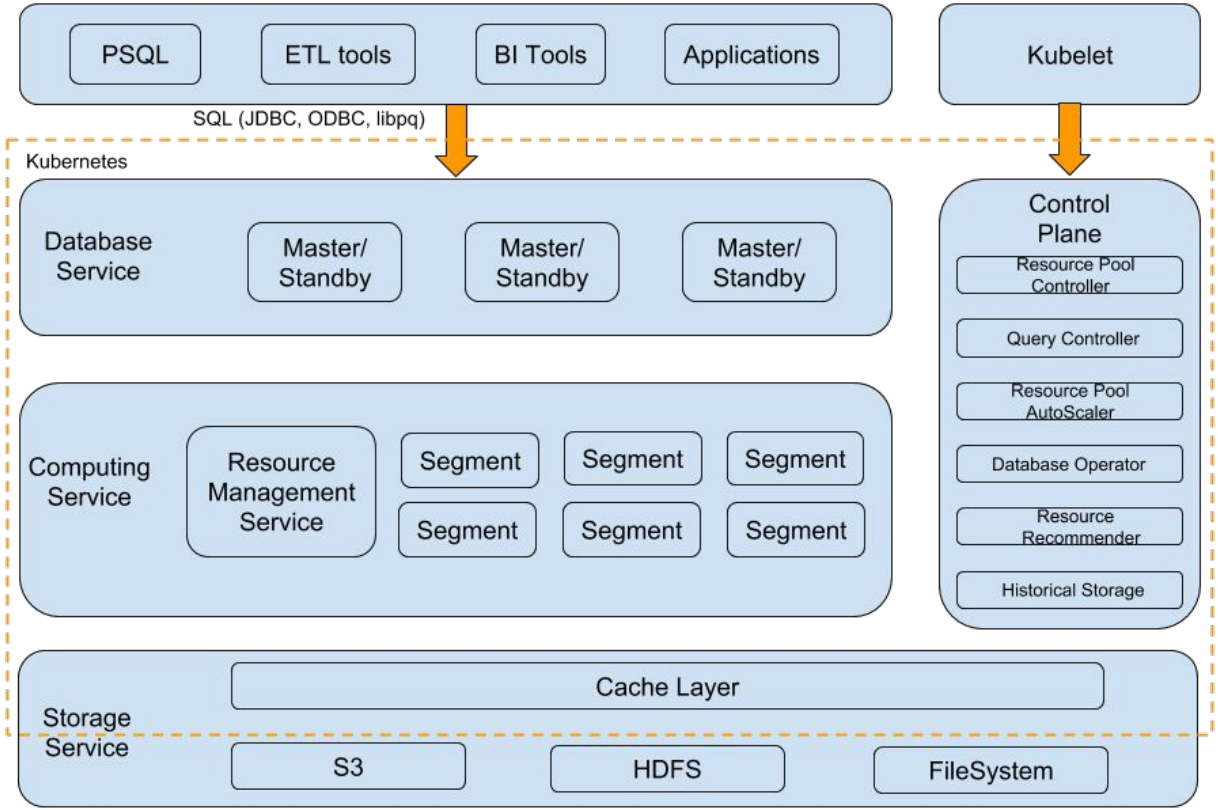
Requirement

- Database as Service
- Efficient Resource Management
- Infrastructure Agnostic
- DBA Free

Deployment & Operation

- Container VS. Virtual Machine
- Kubernetes
  - Service discovery
  - Load balancing
  - Horizontal and Vertical auto scaling
  - Rolling upgrade
  - Monitor and metrics collection
  - …...

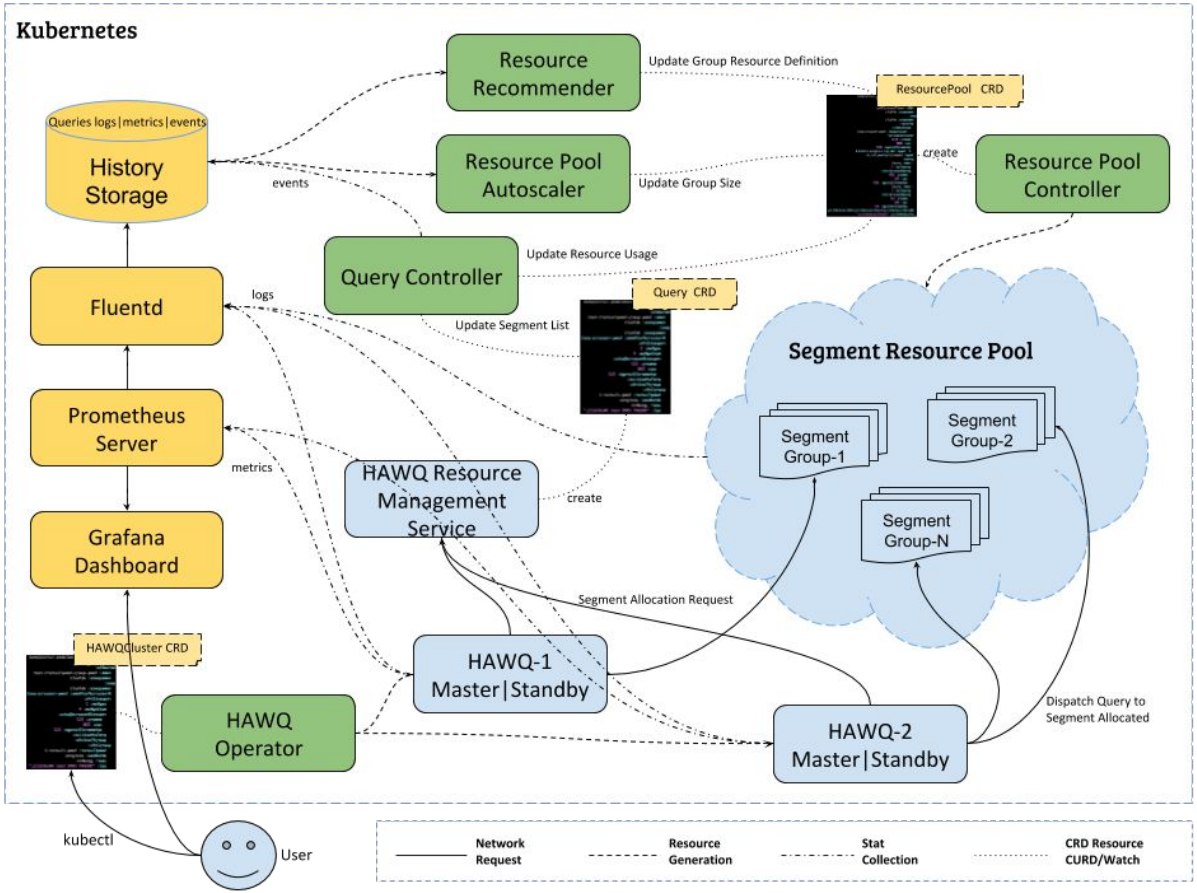
Architecture



Architecture

- Storage Service
  - Cloud Storage, Amazon S3, Hadoop……
  - Unified Cache Lever by Alluxio
- Computing Service
  - Shared Segment Pool
  - Global Resource Management Service
- Database Service
  - Master/Standby as Database
  - Get Segments for Query on Demand
- Control Plane
  - Operator/Controllers as DBA

Apache HAWQ on Kubernetes



Custom Resource

```yaml
apiVersion: pivotaldata.io/v1alpha1
kind: HAWQCluster
metadata:
  name: hawq-cluster-demo
spec:
  clusterPort: 30001
  userGUCs:
    hawq_master_address_port: 5432
  masterResourceLimit:
    cpu: 250
    memory: 1024M
  standbyResourceLimit:
    cpu: 250
    memory: 1024M
status:
  clusterStatus: Ready
```

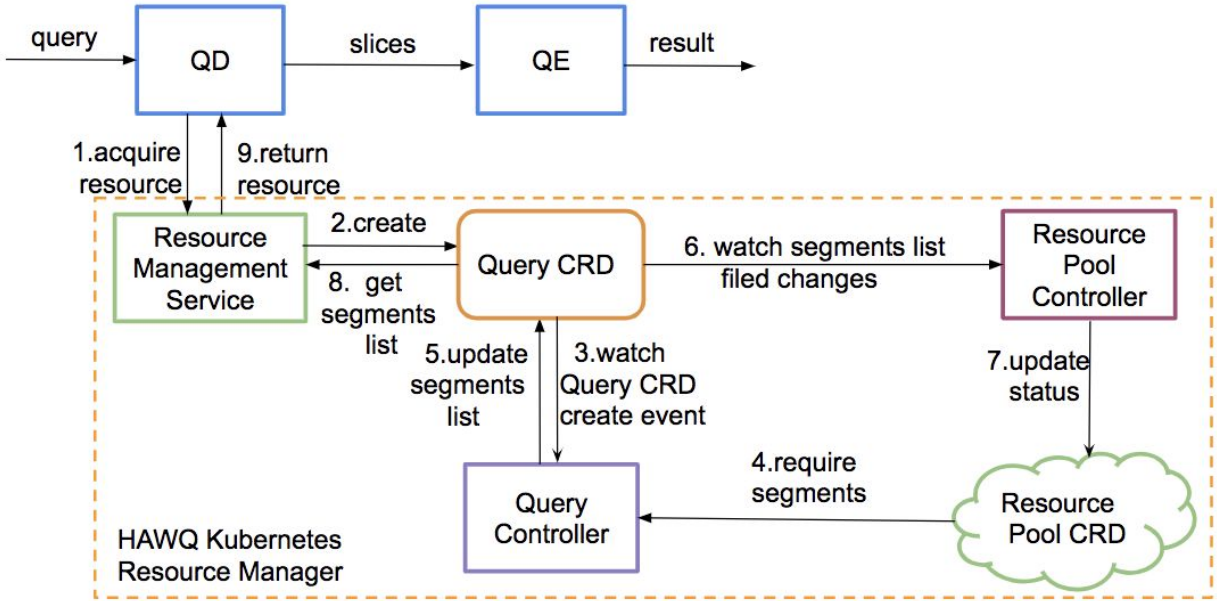
```yaml
apiVersion: pivotaldata.pivotaldata.io/v1alpha1
kind: HAWQQuery
metadata:
  name: q-10.4.1.95-074367
spec:
  queryInfo:
    dataBase: "demo"
    hawqCluster: 10.4.1.95
    sql: "SELECT COUNT(*) FROM lineitem;"
    user: "hawq"
  requestInfo:
    maxSegNum: 1
    preferHostList: ""
    queryPlanInfo: ""
    requestResourceQuota:
      cpu: 200
      ephemeralStorage: 512
      memory: 512
    segNum: 1
  resourcePoolName: hawq-resource-pool
status:
  actualUsedResource: null
  events: null
  queryExecuteStatus: "SUCCESS"
  segmentList:
  - ip: 10.4.1.93
    name: group1-0
  startTimeStamp: 2018-11-09T10:00:17.284390568Z
  stopTimeStamp: 2018-11-09T10:00:25.193212063Z
```

```yaml
apiVersion: pivotaldata.pivotaldata.io/v1alpha1
kind: HAWQResourcePool
metadata:
  name: hawq-resource-pool
spec:
  allocatePolicy: FastAllocatePolicy
  autoScale:
    maxSegmentNum: 4
    scaleDownInterval: 30
    scaleDownRatio: 0.25
    scaleUpRatio: 0.75
  groups:
  - groupResourceLimit:
      cpu: 250
      ephemeralStorage: 1024
      memory: 1024
    groupSize: 1
    name: group1
  - groupResourceLimit:
      cpu: 250
      ephemeralStorage: 512
      memory: 512
    groupSize: 1
    name: group2
  image: hawqbeijing/hawq:demo
  resourceCapacity:
    cpu: 4000
    ephemeralStorage: 3072
    memory: 6144
status:
  availableResourceStatus:
    cpu: 500
    ephemeralStorage: 1536
    memory: 1536
  resourceGroupStatus:
  - allocatedPodNum: 0
    availablePodNum: 1
    name: group1
  - allocatedPodNum: 0
    availablePodNum: 1
    name: group2
  resourcePoolStatus: ready
```

Controller

- HAWQ Operator
- Resource Pool Controller
- Resource Pool AutoScaler
- Resource Recommender
- Query Controller

Apache HAWQ on Kubernetes



Lesson Learned

Architecture

- Service Oriented Architecture
  - Monolithic → Micro Service
- Resource Centric
  - Abstract Component as Resource
  - Service for Resource Usage
  - Controller for Resource Management

Containerization

- Container != Image
- Container != VM
- Container = Fine-grained Resource

Resource Management

- Traditional Database
  - Fixed resources
  - Balance resource usage among queries
- Cloud Database
  - Dynamic resources
  - Maximize resource sharing
  - Maximize resource utilization for each query

Resource Monitoring & Tuning

- Database
  - Variant Query Workload
  - Data Size
  - ……
- Query Similarity and Classification
- Query Resource Monitoring
  - Pod Runtime Metrics
  - Application Logs
  - Kubernetes Events
  - ……
- Intelligent Resource Tuning
  - Resource Pool Definition
  - Horizontal & Vertical

Kubernetes Ecosystem

- Log Collection
  - Fluentd
- Monitoring and Metrics Collection
  - Prometheus
- Visualization
  - Grafana
- ……

Others

- Management Utilities
- Imperative VS. Declarative
- Pod Priority
- ……

Pivotal