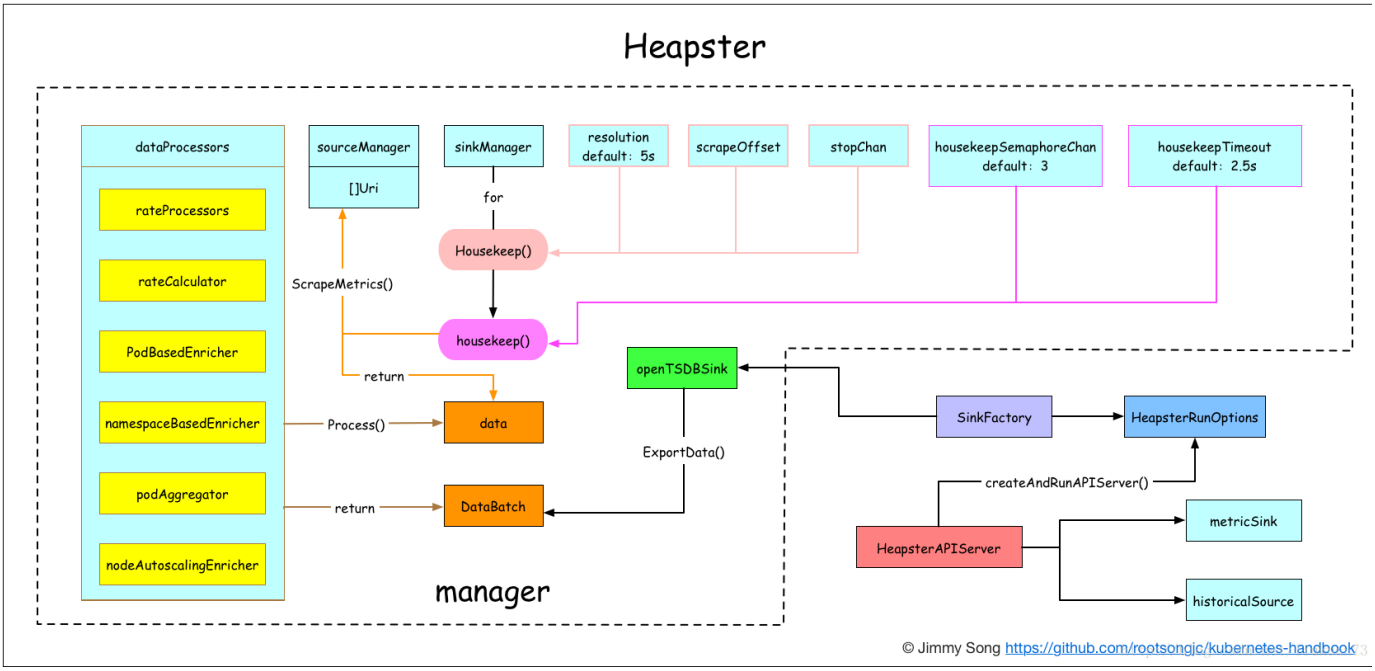
kubernetes监控：heapster+influxdb+grafana

参考：https://blog.csdn.net/liukuan73/article/details/78704395

heapster原理图



1、授权

注:（否则会出现“Failed to list *v1.Node: nodes is forbidden: User”，由于heapster没有集群角色权限导致，无法访问kubernetes node节点数据）

```
$ kubectl create clusterrolebinding heapster-cluster
--clusterrole=cluster-admin --serviceaccount=kube-system:heapster
```

2、安装InfluxDB

influxdb 官方建议使用命令行或 HTTP API 接口来查询数据库，从 v1.1.0 版本开始默认关闭 admin UI，将在后续版本中移除 admin UI 插件。

开启方式：修改配置文件 /etc/config.toml ，修改35行 “enabled = true”

InfluxDB是一个开源的时序数据库，使用GO语言开发，特别适合用于处理和分析资源监控数据这种时序相关数据。而InfluxDB自带的各种特殊函数如求标准差，随机取样数据，统计数据变化比等，使数据统计和实时分析变得十分方便。

详细请参考：https://blog.csdn.net/liukuan73/article/details/79950329

```
$ kubectl  apply  -f  influxdb.yaml
$ cat influxdb.yaml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: monitoring-influxdb     #
  namespace: kube-system
spec:
  replicas: 1
  template:
    metadata:
      labels:
        task: monitoring
        k8s-app: influxdb
    spec:
      containers:
      - name: influxdb
        image: registry.odc.sunline.cn/test/heapster-influxdb-amd64:v1.3.3
## eapsterharbor
        volumeMounts:
        - mountPath: /data
          name: influxdb-storage
      imagePullSecrets:
        - name: registry-user-pass
      volumes:
      - name: influxdb-storage
        emptyDir: {}
---
apiVersion: v1
kind: Service
metadata:
  labels:
    task: monitoring
    # For use as a Cluster add-on
(https://github.com/kubernetes/kubernetes/tree/master/cluster/addons)
    # If you are NOT using this as an addon, you should comment out this
line.
    kubernetes.io/cluster-service: 'true'
    kubernetes.io/name: monitoring-influxdb
  name: monitoring-influxdb
  namespace: kube-system
spec:
  type: NodePort
  ports:
  - port: 8086
    targetPort: 8086   ## heapster
    nodePort: 30004    ##
  selector:
    k8s-app: influxdb
```

3、安装Heapster

什么是Heapster？

Heapster是容器集群监控和性能分析工具，天然的支持Kubernetes和CoreOS。
Kubernetes有个出名的监控agent—cAdvisor。在每个kubernetes
Node上都会运行cAdvisor，它会收集本机以及容器的监控数据(cpu,memory,filesystem,network,uptime)。
在较新的版本中，K8S已经将cAdvisor功能集成到kubelet组件中。每个Node节点可以直接进行web访问。

引用：https://www.kubernetes.org.cn/932.html

```
$ kubectl  apply  -f  heapster.yaml
$ cat heapster.yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: heapster     ## heapster
  namespace: kube-system   ##
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: heapster
  namespace: kube-system
spec:
  replicas: 1
  template:
    metadata:
      labels:
        task: monitoring
        k8s-app: heapster
    spec:
      serviceAccountName: heapster
      containers:
      - name: heapster
        image: registry.odc.sunline.cn/test/heapster-amd64:v1.5.2   ##
heapsterharbor
        imagePullPolicy: IfNotPresent
        command:
        - /heapster
        - --source=kubernetes:https://kubernetes.default     ## kubernetes
        - --sink=influxdb:http://10.101.68.151:8086       ##
influxdbipkubernetesUIip
      imagePullSecrets:
        - name: registry-user-pass           ## harbor
---
apiVersion: v1
kind: Service
metadata:
  labels:
```

```
    task: monitoring
    # For use as a Cluster add-on
(https://github.com/kubernetes/kubernetes/tree/master/cluster/addons)
    # If you are NOT using this as an addon, you should comment out this
line.
    kubernetes.io/cluster-service: 'true'
    kubernetes.io/name: Heapster
  name: heapster
  namespace: kube-system
spec:
  type: NodePort      ## nodePort
  ports:
  - port: 80       ##
    targetPort: 8082     ##
    nodePort: 30003      ##
  selector:
```

```
    k8s-app: heapster
```

4、安装Grafana

Grafana是一个开源的度量分析与可视化套件。经常被用作基础设施的时间序列数据和应用程序分析的可视化，它在其他领域也被广泛的使用包括工业传感器、家庭自动化、天气和过程控制等。

Grafana支持许多不同的数据源。每个数据源都有一个特定的查询编辑器,该编辑器定制的特性和功能是公开的特定数据来源。

官方支持以下数据源:Graphite，InfluxDB，OpenTSDB，Prometheus，Elasticsearch，CloudWatch和KairosDB。

每个数据源的查询语言和能力都是不同的。你可以把来自多个数据源的数据组合到一个仪表板，但每一个面板被绑定到一个特定的数据源,它就属于一个特定的组织。下面我们来一起感受一下grafana。

引用：https://blog.csdn.net/bbwangj/article/details/81109615

```
$ kubectl  apply  -f  grafana.yaml
$ cat  grafana.yaml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: monitoring-grafana  ##
  namespace: kube-system
spec:
  replicas: 1
  template:
    metadata:
      labels:
        task: monitoring
        k8s-app: grafana
    spec:
      containers:
      - name: grafana
        image: registry.odc.sunline.cn/test/heapster-grafana-amd64:v4.4.3
##
        ports:
        - containerPort: 3000
          protocol: TCP
        volumeMounts:
        - mountPath: /etc/ssl/certs
          name: ca-certificates
          readOnly: true
        - mountPath: /var
          name: grafana-storage
        env:
        - name: INFLUXDB_HOST
          value: monitoring-influxdb
        - name: GF_SERVER_HTTP_PORT
          value: "3000"
          # The following env variables are required to make Grafana
accessible via
```

```
          # the kubernetes api-server proxy. On production clusters, we
recommend
          # removing these env variables, setup auth for grafana, and
expose the grafana
          # service using a LoadBalancer or a public IP.
        - name: GF_AUTH_BASIC_ENABLED
          value: "false"
        - name: GF_AUTH_ANONYMOUS_ENABLED
          value: "true"
        - name: GF_AUTH_ANONYMOUS_ORG_ROLE
          value: Admin
        - name: GF_SERVER_ROOT_URL
          # If you're only using the API Server proxy, set this value
instead:
          # value:
/api/v1/namespaces/kube-system/services/monitoring-grafana/proxy
          value: /
      imagePullSecrets:
        - name: registry-user-pass   ##
      volumes:
      - name: ca-certificates
        hostPath:
          path: /etc/ssl/certs
      - name: grafana-storage
        emptyDir: {}
---
apiVersion: v1
kind: Service
metadata:
  labels:
    # For use as a Cluster add-on
(https://github.com/kubernetes/kubernetes/tree/master/cluster/addons)
    # If you are NOT using this as an addon, you should comment out this
line.
    kubernetes.io/cluster-service: 'true'
    kubernetes.io/name: monitoring-grafana
  name: monitoring-grafana
  namespace: kube-system
spec:
  type: NodePort
  # In a production setup, we recommend accessing Grafana through an
external Loadbalancer
  # or through a public IP.
  # type: LoadBalancer
  # You could also use NodePort to expose the service at a
randomly-generated port
  # type: NodePort
  ports:
  - port: 80
    targetPort: 3000
    nodePort: 30002    ## grafana
  selector:
```

```
    k8s-app: grafana
```