Kubernetes 使得管理复杂环境变得更简单，但是对 kubernetes 本身的各种组件还有运行在 kubernetes 集群上的各种应用程序做到很好的洞察就很难了。Kubernetes 本身对应用程序的做了很多抽象，在生产环境下对这些不同的抽象组件的健康就是迫在眉睫的事情。

我们在安装 kubernetes 集群的时候，默认安装了 kubernetes 官方提供的 heapster 插件，可以对 kubernetes 集群上的应用进行简单的监控，获取 pod 级别的**内存**、**CPU**和**网络**监控信息，同时还能够通过 API 监控 kubernetes 中的基本资源监控指标。

Heapster作为kubernetes安装过程中默认安装的一个插件。这对于集群监控十分有用，同时在Horizontal Pod Autoscaling中也用到了，HPA将Heapster作为Resource Metrics API，向其获取metric，做法是在kube-controller-manager 中配置--api-server指向kube-aggregator，也可以使用heapster来实现，通过在启动heapster的时候指定--api-server=true。

Heapster可以收集Node节点上的cAdvisor数据，还可以按照kubernetes的资源类型来集合资源，比如Pod、Namespace域，可以分别获取它们的CPU、内存、网络和磁盘的metric。默认的metric数据聚合时间间隔是1分钟。

安装heapster插件

到 https://github.com/kubernetes/heapster/releases下载最新版本的 heapster。

```
[root@vlnx251101 ~]# tar xf heapster-v1.5.4.tar.gz
[root@vlnx251101 ~]# cd heapster-1.5.4/deploy/kube-config/influxdb/
```

```
[root@vlnx251101 influxdb]# ls
grafana.yaml  heapster.yaml  influxdb.yaml

[root@vlnx251101 influxdb]# kubectl create -f .
deployment.extensions/monitoring-grafana created
service/monitoring-grafana created
serviceaccount/heapster created
clusterrolebinding.rbac.authorization.k8s.io/heapster
created
deployment.extensions/heapster created
service/heapster created
configmap/influxdb-config created
deployment.extensions/monitoring-influxdb created
service/monitoring-influxdb created
```

检查执行结果

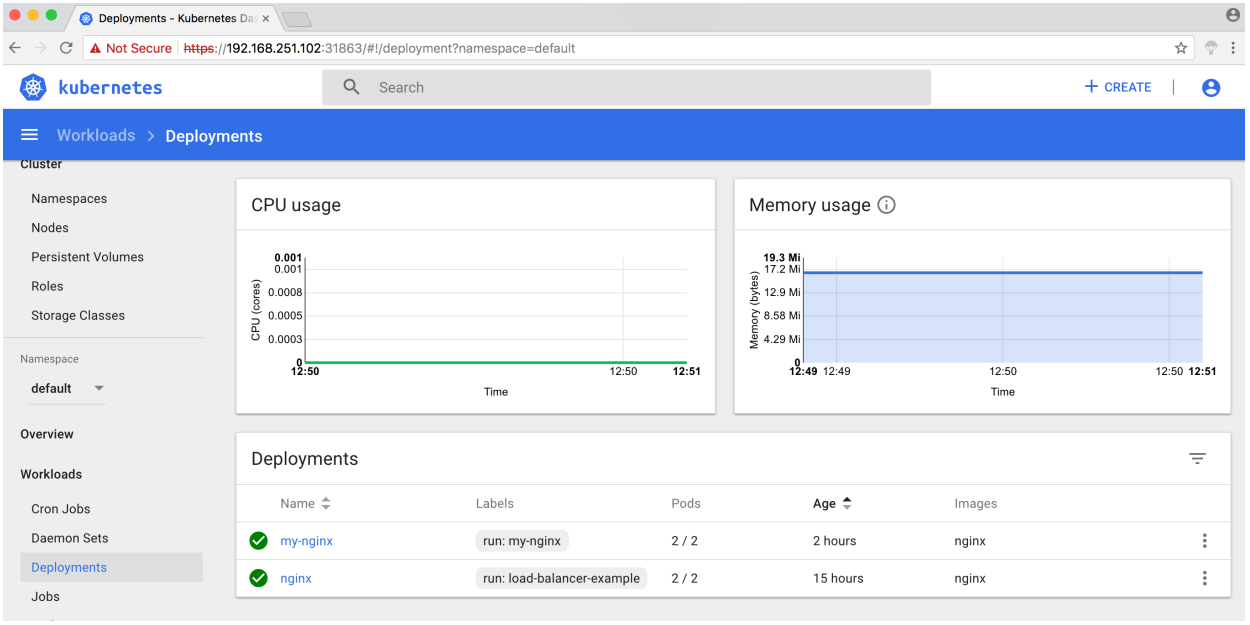
检查 Deployment

```
[root@vlnx251101 influxdb]# kubectl get deployments -n kube-system |
grep -E 'heapster|monitoring'
heapster               1         1         1            1           7s
monitoring-grafana     1         1         1            1           7s
monitoring-influxdb    1         1         1            1           7s
```

检查 Pods

```
[root@vlnx251101 influxdb]# kubectl get pods -n kube-system | grep -E
'heapster|monitoring'
heapster-55884f49b6-rr8p7                1/1       Running   0
1m
monitoring-grafana-84fd47f8c9-7gvvd      1/1       Running   0
1m
monitoring-influxdb-64b7644788-2xjvp     1/1       Running   0
1m
```

检查 kubernets dashboard 界面，看是显示各 Nodes、Pods 的 CPU、内存、负载等利用率曲线图；



图片 - dashboard-heapster

访问 grafana

1.通过 kube-apiserver 访问：

获取 monitoring-grafana 服务 URL

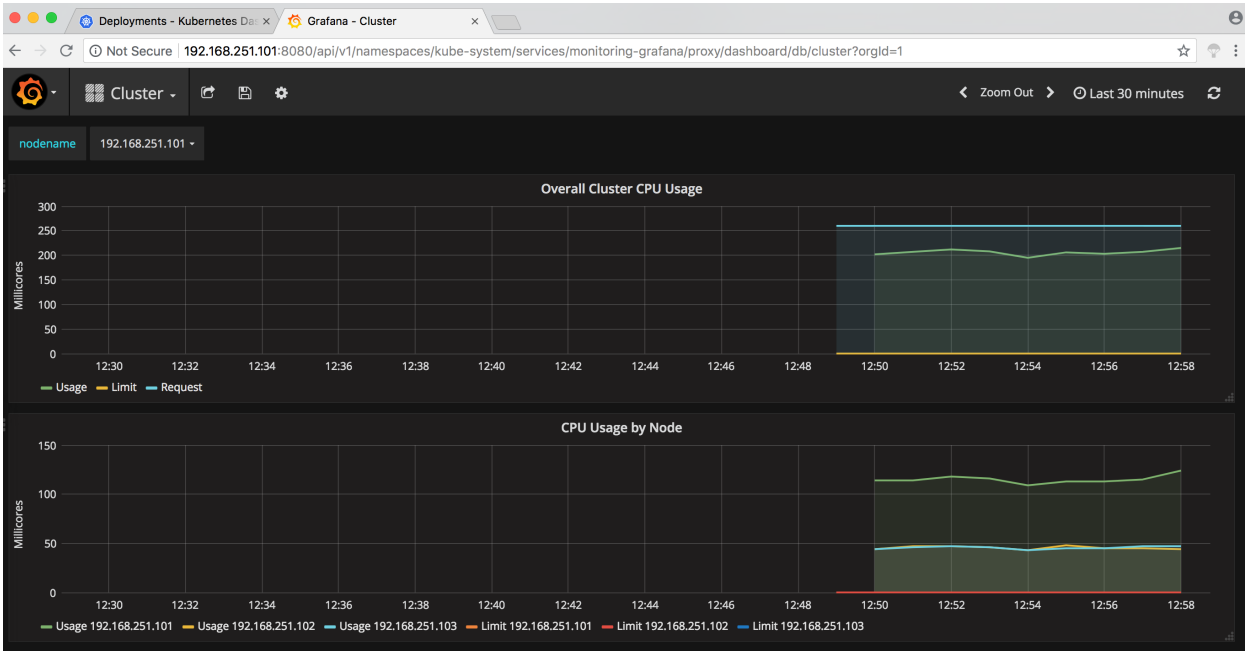
```
[root@vlnx251101 influxdb]# kubectl cluster-info
Kubernetes master is running at
https://192.168.251.101:6443
Heapster is running at
https://192.168.251.101:6443/api/v1/namespaces/kube-system/services/heapster/proxy
KubeDNS is running at
https://192.168.251.101:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
monitoring-grafana is running at
https://192.168.251.101:6443/api/v1/namespaces/kube-
```

system/services/monitoring-grafana/proxy

monitoring-influxdb is running at

https://192.168.251.101:6443/api/v1/namespaces/kube-system/services/monitoring-influxdb:http/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.



浏览器访问

URL： http://192.168.251.101:8080/api/v1/namespaces/kube-system/services/monitoring-grafana/proxy

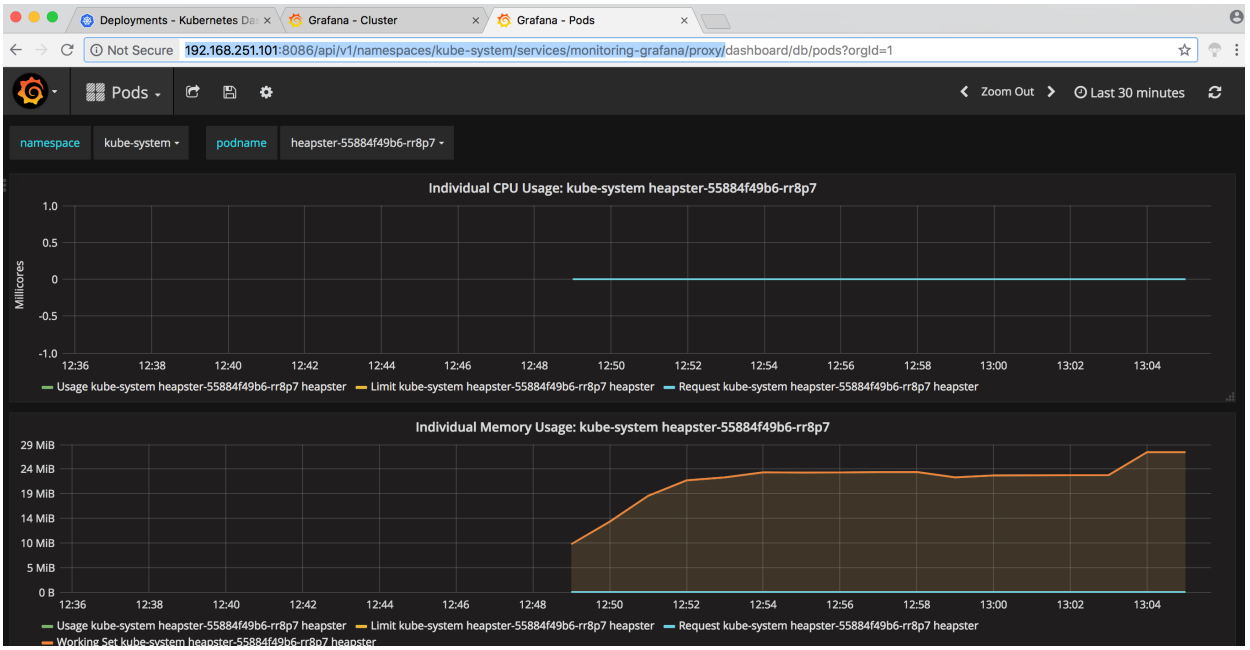
2.通过 kubectl proxy 访问：

创建代理

```
[root@vlnx251101 influxdb]# kubectl proxy --address='192.168.251.101' --port=8086 --accept-hosts='^*$'
Starting to serve on 192.168.251.101:8086
```

浏览器访问 URL：

http://192.168.251.101:8086/api/v1/namespaces/kube-system/services/monitoring-grafana/proxy/



图片 - grafana

```
  Warning  Failed     5s (x3 over 21s)  kubelet, 192.168.251.101
Error: failed to start container "influxdb": Error response from
daemon: oci runtime error: container_linux.go:247: starting
container process caused "container init exited prematurely"
```

Metrics-server

```
[root@vlnx251101 ~ (⎈ |kubernetes:kube-system)]# git clone
https://github.com/kubernetes-incubator/metrics-server.git
```

```
[root@vlnx251101 ~ (⎈ |kubernetes:kube-system)]# vim
/etc/kubernetes/controller-manager
```

```
KUBE_CONTROLLER_MANAGER_ARGS="--address=127.0.0.1 --service-cluster-ip-range=10.254.0.0/16 --cluster-name=kubernetes --cluster-signing-cert-file=/etc/kubernetes/ssl/ca.pem --cluster-signing-key-file=/etc/kubernetes/ssl/ca-key.pem  --service-account-private-key-file=/etc/kubernetes/ssl/ca-key.pem --root-ca-file=/etc/kubernetes/ssl/ca.pem --leader-elect=true --horizontal-pod-autoscaler-use-rest-clients=true"

[root@vlnx251101 ~ (⎈ |kubernetes:kube-system)]# systemctl restart kube-controller-manager.service ; systemctl status kube-controller-manager.service


[root@vlnx251101 ~ (⎈ |kubernetes:kube-system)]# vim /etc/kubernetes/apiserver


KUBE_API_ARGS="--authorization-mode=Node,RBAC --enable-bootstrap-token-auth --runtime-config=rbac.authorization.k8s.io/v1beta1 --kubelet-https=true --token-auth-file=/etc/kubernetes/token.csv --service-node-port-range=30000-32767 --tls-cert-file=/etc/kubernetes/ssl/kubernetes.pem --tls-private-key-file=/etc/kubernetes/ssl/kubernetes-key.pem --client-ca-file=/etc/kubernetes/ssl/ca.pem --service-account-key-file=/etc/kubernetes/ssl/ca-key.pem --etcd-cafile=/etc/kubernetes/ssl/ca.pem --etcd-certfile=/etc/kubernetes/ssl/kubernetes.pem --etcd-keyfile=/etc/kubernetes/ssl/kubernetes-key.pem --enable-
```

```
swagger-ui=true --apiserver-count=3 --audit-log-maxage=30
--audit-log-maxbackup=3 --audit-log-maxsize=100 --audit-
log-path=/var/lib/audit.log --event-ttl=1h --
requestheader-client-ca-file=/etc/kubernetes/ssl/ca.pem --
requestheader-allowed-names=aggregator --requestheader-
extra-headers-prefix=X-Remote-Extra- --requestheader-
group-headers=X-Remote-Group --requestheader-username-
headers=X-Remote-User --proxy-client-cert-
file=/etc/kubernetes/ssl/kubernetes.pem --proxy-client-
key-file=/etc/kubernetes/ssl/kubernetes-key.pem --enable-
aggregator-routing=true"
```

```
[root@vlnx251101 ~ (⎈ |kubernetes:kube-system)]# systemctl
restart kube-apiserver.service ; systemctl status kube-
apiserver.service
```

```
[root@vlnx251101 ~ (⎈ |kubernetes:kube-system)]# cd
metrics-server/deploy/1.8+/
```

```
[root@vlnx251101 1.8+ (⎈ |kubernetes:default)]# vim metrics-server-
deployment.yaml
```

```
    containers:
    - name: metrics-server
      image: zhaoyonggang/metrics-server-amd64:v0.3.1
      imagePullPolicy: Always
      volumeMounts:
      - name: tmp-dir
        mountPath: /tmp
      command:
```

```
      - /metrics-server
      - --metric-resolution=30s
      - --kubelet-insecure-tls
      - --kubelet-preferred-address-
types=InternalIP,Hostname,InternalDNS,ExternalDNS,ExternalIP
```

```
[root@vlnx251101 1.8+ (⎈ |kubernetes:kube-system)]# vim
auth-RBAC.yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: view-metrics
rules:
- apiGroups:
    - metrics.k8s.io
  resources:
    - pods
    - nodes
  verbs:
    - get
    - list
    - watch
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: view-metrics
roleRef:
  apiGroup: rbac.authorization.k8s.io
```

```
  kind: ClusterRole
  name: view-metrics
subjects:
  - apiGroup: rbac.authorization.k8s.io
    kind: User
    name: kubernetes
```

```
[root@vlnx251101 1.8+ (⎈ |kubernetes:kube-system)]# kubectl create -f .
```

```
[root@vlnx251101 1.8+ (⎈ |kubernetes:kube-system)]# vim hpa.yaml
```

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: busybox
  namespace: default
  labels:
    app: busybox
spec:
  replicas: 1
  selector:
    matchExpressions:
    - key: app
      operator: In
      values:
      - busybox
```

```yaml
  template:
    metadata:
      labels:
        app: busybox
    spec:
      containers:
      - name: busybox
        image: busybox
        resources:
          requests:
            cpu: "200m"
            memory: "1000Mi"
          limits:
            cpu: "500m"
            memory: "2000Mi"
        command:
        - "sleep"
        args:
        - "3600"
---
apiVersion: autoscaling/v2beta1
kind: HorizontalPodAutoscaler
metadata:
  name: metric-hpa
  namespace: default
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: busybox
```

```yaml
minReplicas: 1
maxReplicas: 3
metrics:
- type: Resource
  resource:
    name: cpu
    targetAverageUtilization: 50
```