Caicloud

cloud. acceleration. intelligence.

才云科技

Kubernetes 应用开发及实践

何小龙

2017年8月12日

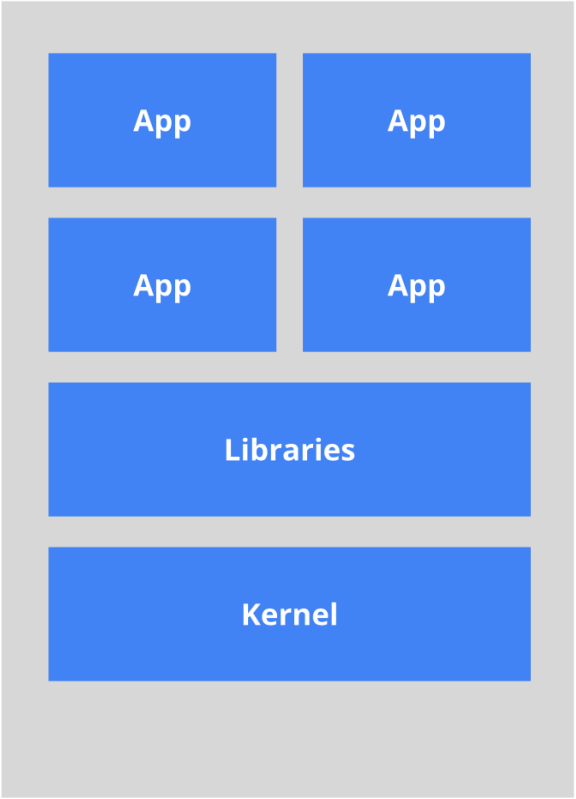
目录

Speech content

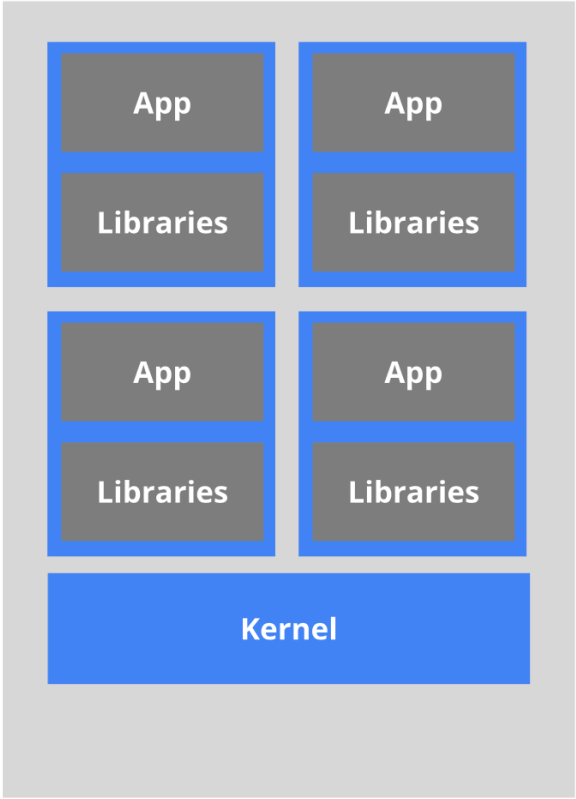
1. 为什么要学习Kubernetes
2. 搭建本地测试环境 MiniKube
3. Kubernetes 涉及的主要概念
4. Kubernetes 实践案例剖析

为什么要学习 Kubernetes

为什么要学习 Kubernetes

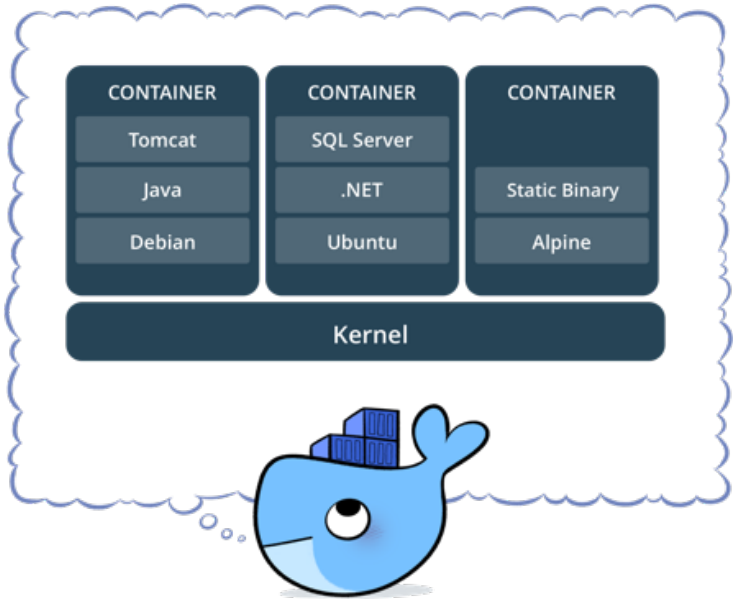


Heavyweight, non-portable
Relies on OS package manager



Small and fast, portable
Uses OS-level virtualization

容器化 + 微服务 = DevOps



Kubernetes

搭建本地测试环境 MiniKube

MiniKube是一套本地单节点Kubernetes体验环境..

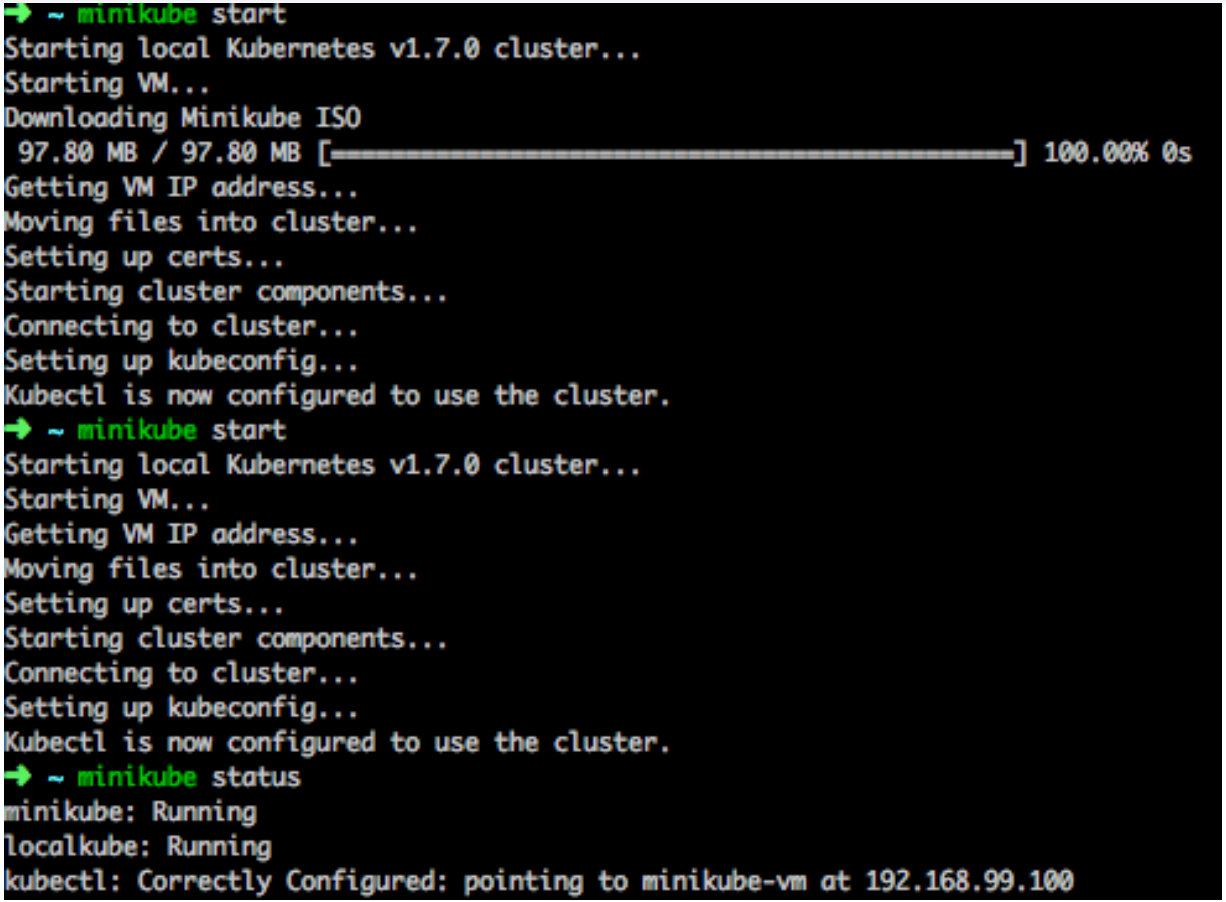
```
# 安装虚拟化软件
$ brew install docker-machine-driver-xhyve
# docker-machine-driver-xhyve need root owner and uid
$ sudo chown root:wheel
$(brew --prefix)/opt/docker-machine-driver-xhyve/bin/docker-machine-driver-xhyve
$ sudo chmod u+s $(brew --prefix)/opt/docker-machine-driver-xhyve/bin/docker-machine-driver-xhyve

# 下载 Mac 版本的minkube
$curl -Lo minikube https://storage.googleapis.com/minikube/releases/v0.21.0/minikube-darwin-amd64 &&
chmod +x minikube && sudo mv minikube /usr/local/bin/
# 启动 Minikube
$ minikube start

# 下载 kubectl 客户端工具
$ curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s
https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/darwin/amd64/kubectl
```

https://github.com/kubernetes/minikube
https://github.com/kubernetes/minikube/blob/master/docs/drivers.md#xhyve-driver

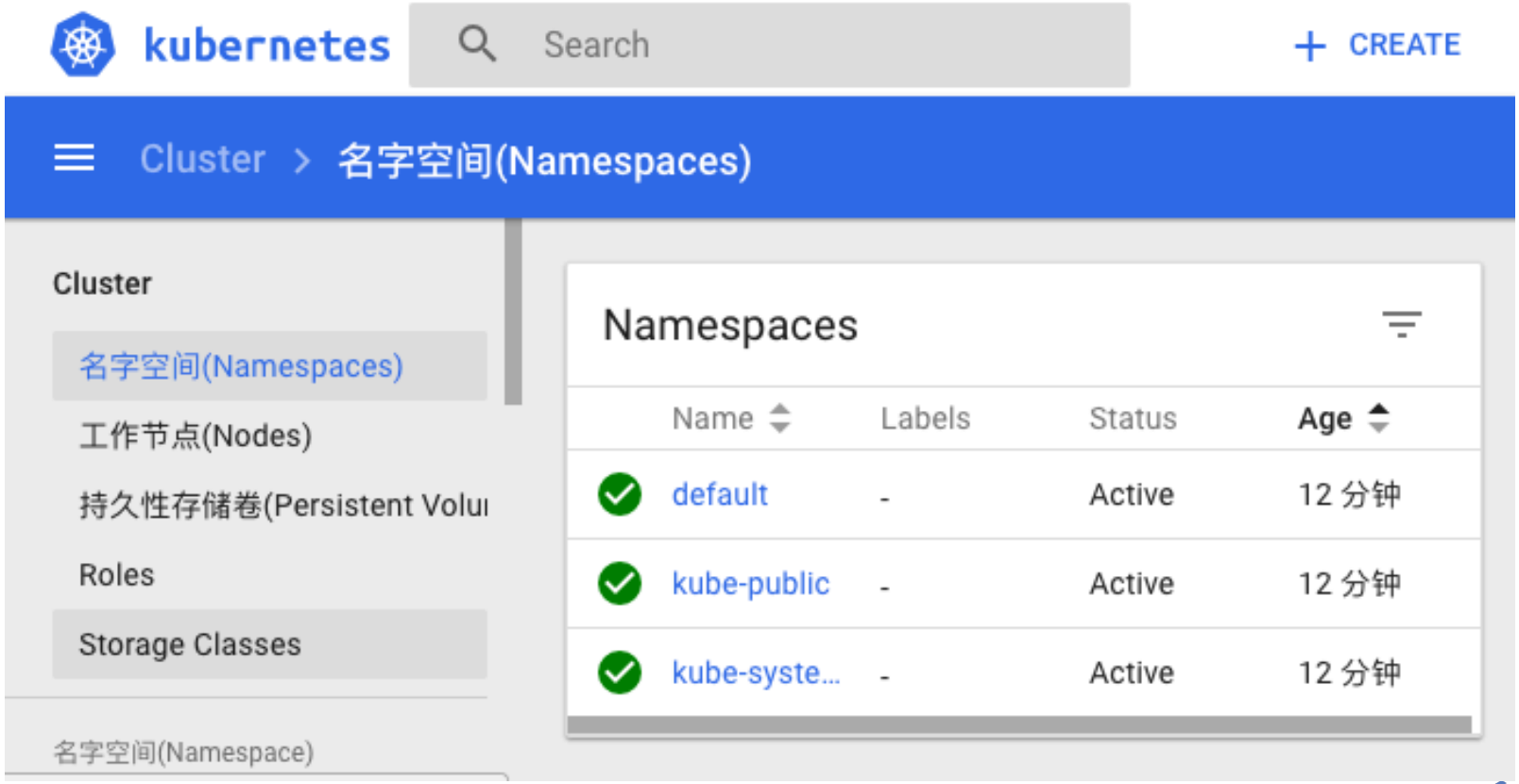
搭建本地测试环境 MiniKube



搭建本地测试环境 MiniKube

```
➜ ~ minikube service list
|-------------|----------------------|-----------------------------|
|  NAMESPACE  |         NAME         |             URL             |
|-------------|----------------------|-----------------------------|
| default     | kubernetes           | No node port                |
| kube-system | kube-dns             | No node port                |
| kube-system | kubernetes-dashboard | http://192.168.99.100:30000 |
|-------------|----------------------|-----------------------------|
➜ ~ docker images
REPOSITORY                                      TAG                 IMAGE ID            CREATED             SIZE
weblogic                                        10.3.6-domain       c439b9e70449        5 days ago          3.45GB
weblogic                                        10.3.6              8bb6c6935a83        5 days ago          3.45GB
oraclelinux                                     jdk6                9bc02d5382b9        5 days ago          505MB
oraclelinux                                     7.3                 44fb4d3b5ead        5 days ago          225MB
gcr.io/google_containers/hyperkube-amd64        v1.7.3              55d97676fd42        8 days ago          615MB
cargo.caicloudprivatetest.com/caicloud/cabin    v2.3.0              c36113476e85        5 weeks ago         738MB
cabin                                           v2.3.0              c36113476e85        5 weeks ago         738MB
➜ ~ eval $(minikube docker-env)

➜ ~ docker images
REPOSITORY                                      TAG                 IMAGE ID            CREATED             SIZE
gcr.io/google_containers/k8s-dns-sidecar-amd64  1.14.4              38bac66034a6        6 weeks ago         41.8MB
gcr.io/google_containers/k8s-dns-kube-dns-amd64 1.14.4              a8e00546bcf3        6 weeks ago         49.4MB
gcr.io/google_containers/k8s-dns-dnsmasq-nanny-amd64 1.14.4         f7f45b9cb733        6 weeks ago         41.4MB
gcr.io/google-containers/kube-addon-manager     v6.4-beta.2         0a951668696f        8 weeks ago         79.2MB
gcr.io/google_containers/kubernetes-dashboard-amd64 v1.6.1          71dfe833ce74        2 months ago        134MB
gcr.io/google_containers/pause-amd64            3.0                 99e59f495ffa        15 months ago       747kB
➜ ~
```


kubernetes
Search
+ CREATE
Cluster > 名字空间(Namespaces)
Cluster
名字空间(Namespaces)
工作节点(Nodes)
持久性存储卷(Persistent Volu
Roles
Storage Classes
名字空间(Namespace)
Namespaces
Name
Labels
Status
Age
default - Active 12 分钟
kube-public - Active 12 分钟
kube-syste... - Active 12 分钟

```
# 配置 minikube 环境
eval $(minikube docker-env)
# 移除 minikube 环境
eval $(minikube docker-env -u)

# 启动简单的容器
docker run -d -p 8081:80 nginx
# 启动两个nginx 副本
kubectl run nginx-test --image=nginx --replicas=2 --port=80

# 服务导出
kubectl expose deploy nginx-test  --port=80 --type=LoadBalancer
```

搭建本地测试环境 MiniKube

```
➜ ~ kubectl get node
NAME       STATUS    AGE       VERSION
minikube   Ready     18m       v1.7.0
➜ ~ kubectl get ns
NAME          STATUS    AGE
default       Active    19m
kube-public   Active    19m
kube-system   Active    19m
➜ ~ kubectl -n kube-system get po
NAME                                 READY     STATUS    RESTARTS   AGE
kube-addon-manager-minikube          1/1       Running   1          19m
kube-dns-910330662-qsjpt             3/3       Running   1          19m
kubernetes-dashboard-7t7jx           1/1       Running   1          19m
➜ ~ kubectl get po
No resources found.
➜ ~ kubectl run nginx-test --image=nginx --replicas=2 --port=80

deployment "nginx-test" created
➜ ~ kubectl get po
NAME                          READY     STATUS              RESTARTS   AGE
nginx-test-861205578-h69b1    0/1       ContainerCreating   0          6s
nginx-test-861205578-k3tzv    0/1       ContainerCreating   0          6s
```

```
➜ ~ kubectl get deploy
NAME         DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
nginx-test   2         2         2            0           10s
➜ ~ kubectl get po
NAME                          READY     STATUS    RESTARTS   AGE
nginx-test-861205578-h69b1    1/1       Running   0          2m
nginx-test-861205578-k3tzv    1/1       Running   0          2m
➜ ~ minikube service list
|-------------|----------------------|-----------------------------|
|  NAMESPACE  |         NAME         |             URL             |
|-------------|----------------------|-----------------------------|
| default     | kubernetes           | No node port                |
| kube-system | kube-dns             | No node port                |
| kube-system | kubernetes-dashboard | http://192.168.99.100:30000 |
|-------------|----------------------|-----------------------------|
```

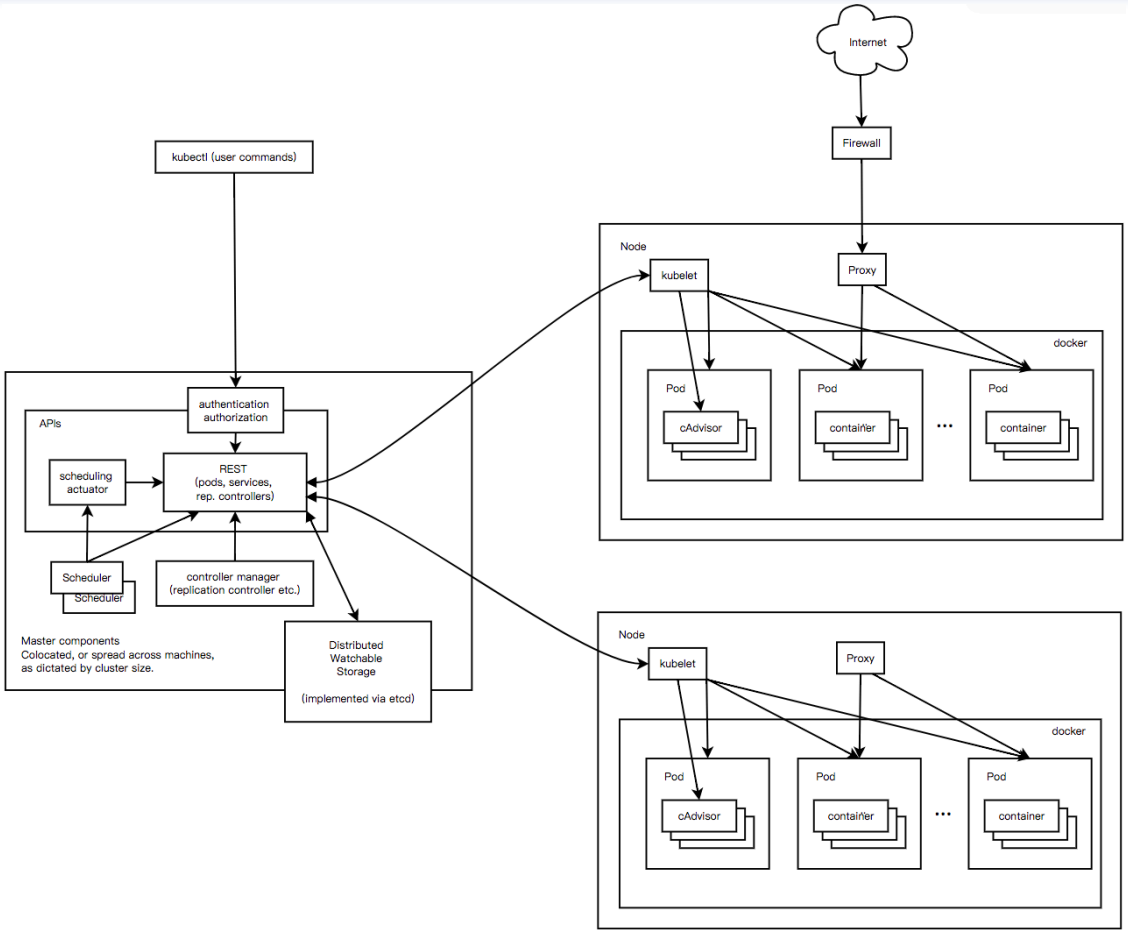
```
➜ ~ kubectl expose deployment nginx-test --port=80 --type=LoadBalancer
service "nginx-test" exposed
➜ ~ minikube service list
|-------------|----------------------|-----------------------------|
|  NAMESPACE  |         NAME         |             URL             |
|-------------|----------------------|-----------------------------|
| default     | kubernetes           | No node port                |
| default     | nginx-test           | http://192.168.99.100:30007 |
| kube-system | kube-dns             | No node port                |
| kube-system | kubernetes-dashboard | http://192.168.99.100:30000 |
|-------------|----------------------|-----------------------------|
```

Kubernetes 涉及的主要概念

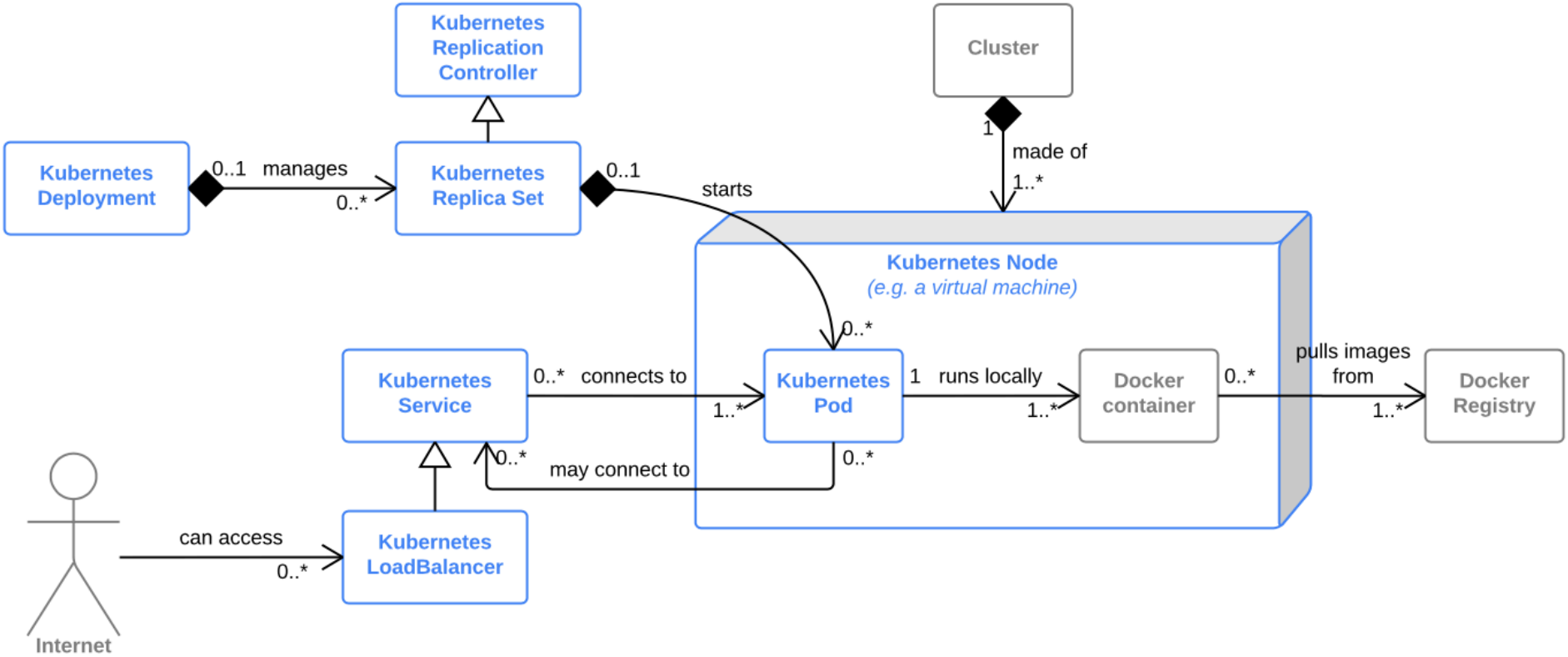
DB GEEK IT大咖说 知识分享平台

Caicloud 才云科技 cloud. acceleration. intelligence.

Kubernetes 涉及的主要概念-架构图



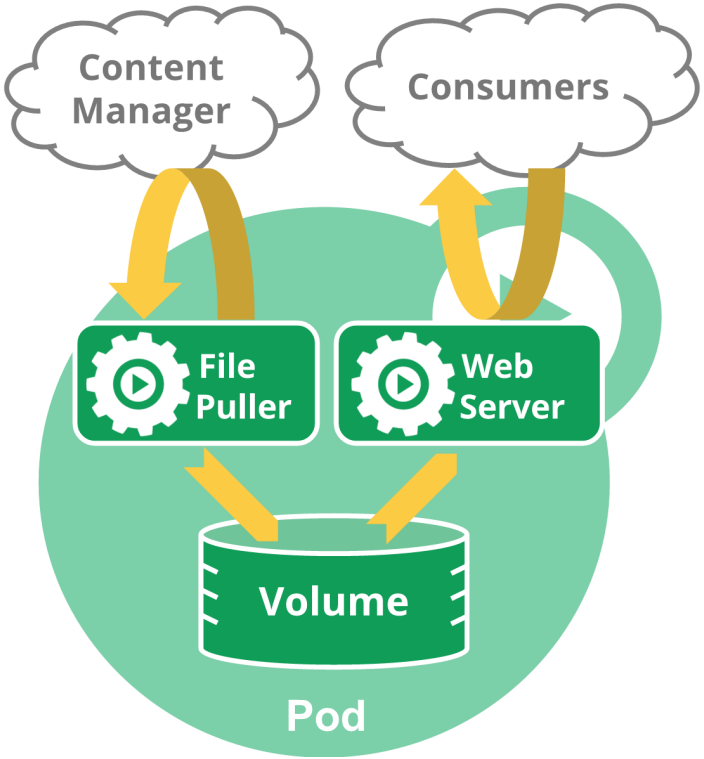
Kubernetes 涉及的主要概念-UML



- 物理机/虚拟机
- Master（管理节点）
  - 资源管理
  - Pod 调度
  - 弹性伸缩
  - 安全控制
  - 系统监控
  - 纠错
- Node（工作节点）
  - 运行 Pod 化的服务

```
[root@c1v41 ~]# kubectl get node
NAME          STATUS                     AGE       VERSION
kube-master-1   Ready,SchedulingDisabled   202d      v1.6.2-dirty
kube-master-2   Ready,SchedulingDisabled   202d      v1.6.2-dirty
kube-master-3   Ready,SchedulingDisabled   202d      v1.6.2-dirty
kube-node-1     Ready                      202d      v1.6.2-dirty
kube-node-10    Ready                      201d      v1.6.2-dirty
kube-node-11    Ready                      201d      v1.6.2-dirty
kube-node-12    Ready                      201d      v1.6.2-dirty
kube-node-13    Ready                      201d      v1.6.2-dirty
kube-node-14    Ready                      201d      v1.6.2-dirty
kube-node-15    Ready                      201d      v1.6.2-dirty
kube-node-16    Ready                      201d      v1.6.2-dirty
kube-node-17    Ready                      201d      v1.6.2-dirty
kube-node-18    Ready                      201d      v1.6.2-dirty
kube-node-19    Ready                      201d      v1.6.2-dirty
kube-node-2     Ready                      202d      v1.6.2-dirty
kube-node-20    Ready                      201d      v1.6.2-dirty
kube-node-21    Ready                      201d      v1.6.2-dirty
kube-node-22    Ready                      201d      v1.6.2-dirty
kube-node-23    Ready                      201d      v1.6.1-dirty
kube-node-24    Ready                      201d      v1.6.1-dirty
kube-node-25    Ready                      201d      v1.6.1-dirty
kube-node-26    Ready                      201d      v1.6.1-dirty
```

- 多个容器的集合
- 最小调度单位
- 共享 namespace
  - PID 名字空间（同一 Pod 内的容器能看到其他容器的 PID，Docker 容器暂不支持）
  - 网络名字空间（共享 IP 和 localhost）
  - IPC 名字空间（能够使用 SystemV IPC 或者POSIX 消息队列进行通信）
  - UTS 名字空间（共享同一个主机名）
- 共享 Volumes
- 资源配额
  - 资源限制
  - 请求的最少资源 （min request）



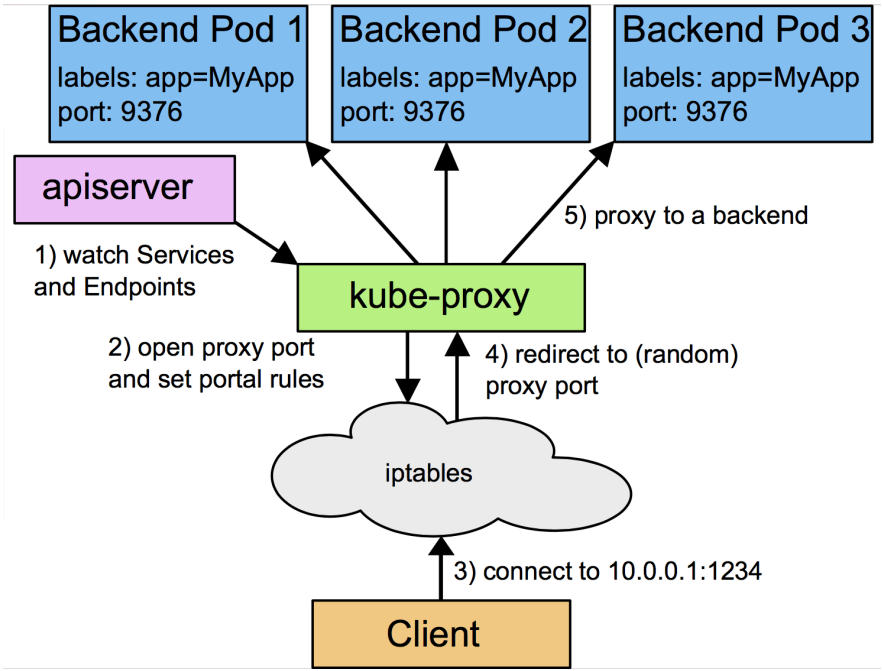
Kubernetes 涉及的主要概念-Deployment

- 提供 Pod 和 Replica Set 的声明式更新
- 把部署，滚动更新和回滚进行了自动化
- 创建 Deployment
  - kubectl create -f docs/user-guide/nginx-deployment.yaml --record
- 更新 Deployment
  - kubectl set image deployment/nginx-deployment nginx=nginx:1.9.1
- 查看 Deployment 的修订版本
  - kubectl rollout history deployment/nginx-deployment
- 回滚到上一个版本
  - kubectl rollout undo deployment/nginx-deployment
- 回滚到指定的版本
  - kubectl rollout undo deployment/nginx-deployment --to-revision=2

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.7.9
        ports:
        - containerPort: 80
```

- 一个唯一指定的名字
- 一个虚拟 IP 和端口号
- 服务类型
  - ClusterIP（default）
  - Nodeport
  - LoadBalancer
- 服务发现
  - 环境变量
  - DNS
- 负载均衡

```
{
  "kind": "Service",
  "apiVersion": "v1",
  "metadata": {
    "name": "my-service"
  },
  "spec": {
    "selector": {
      "app": "MyApp"
    },
    "ports": [
      {
        "protocol": "TCP",
        "port": 1234,
        "targetPort": 9376,
        "nodePort": 30029
      }
    ],
    "type": "NodePort"
  }
}
```



Kubernetes 实践案例剖析

```
→ tutorial kubectl create -f mysql-wordpress-pd/local-volumes.yaml
persistentvolume "local-pv-1" created
persistentvolume "local-pv-2" created
→ tutorial kubectl create secret generic mysql-pass --from-file=mysql-wordpress-pd/password.txt
kubectl create -f mysql-wordpress-pd/mysql-deployment.yaml
kubectl create -f mysql-wordpress-pd/wordpress-deployment.yaml
secret "mysql-pass" created
service "wordpress-mysql" created
persistentvolumeclaim "mysql-pv-claim" created
deployment "wordpress-mysql" created
service "wordpress" created
persistentvolumeclaim "wp-pv-claim" created
deployment "wordpress" created
```

```
→ tutorial kubectl get po
NAME                               READY     STATUS              RESTARTS   AGE
nginx-test-861205578-h69b1         1/1       Running             0          17m
nginx-test-861205578-k3tzv         1/1       Running             0          17m
wordpress-744530099-hz28d          0/1       ContainerCreating   0          2m
wordpress-mysql-2222028001-hk1g4   0/1       ContainerCreating   0          2m
→ tutorial kubectl get po
NAME                               READY     STATUS    RESTARTS   AGE
nginx-test-861205578-h69b1         1/1       Running   0          27m
nginx-test-861205578-k3tzv         1/1       Running   0          27m
wordpress-744530099-hz28d          1/1       Running   0          11m
wordpress-mysql-2222028001-hk1g4   1/1       Running   0          11m
→ tutorial kubectl get svc
NAME              CLUSTER-IP    EXTERNAL-IP   PORT(S)        AGE
kubernetes        10.0.0.1      <none>        443/TCP        53m
nginx-test        10.0.0.135    <pending>     80:30007/TCP   20m
wordpress         10.0.0.75     <pending>     80:32589/TCP   11m
wordpress-mysql   None          <none>        3306/TCP       11m
```

Kubernetes 实践案例剖析-Wordpress

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mysql-pv-claim
  labels:
    app: wordpress
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 20Gi
```

```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: local-pv-1
  labels:
    type: local
spec:
  capacity:
    storage: 20Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: /tmp/data/pv-1
```

```yaml
apiVersion: v1
kind: Service
metadata:
  name: wordpress-mysql
  labels:
    app: wordpress
spec:
  ports:
    - port: 3306
  selector:
    app: wordpress
    tier: mysql
  clusterIP: None
```

```yaml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: wordpress-mysql
  labels:
    app: wordpress
spec:
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: wordpress
        tier: mysql
    spec:
      containers:
      - image: mysql:5.6
        name: mysql
        env:
          # $ kubectl create secret generic mysql-pass --from-file=password.txt
          # make sure password.txt does not have a trailing newline
        - name: MYSQL_ROOT_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mysql-pass
              key: password.txt
        ports:
        - containerPort: 3306
          name: mysql
        volumeMounts:
        - name: mysql-persistent-storage
          mountPath: /var/lib/mysql
      volumes:
      - name: mysql-persistent-storage
        persistentVolumeClaim:
          claimName: mysql-pv-claim
```

Kubernetes 实践案例剖析-Wordpress

```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: local-pv-2
  labels:
    type: local
spec:
  capacity:
    storage: 20Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: /tmp/data/pv-2
```

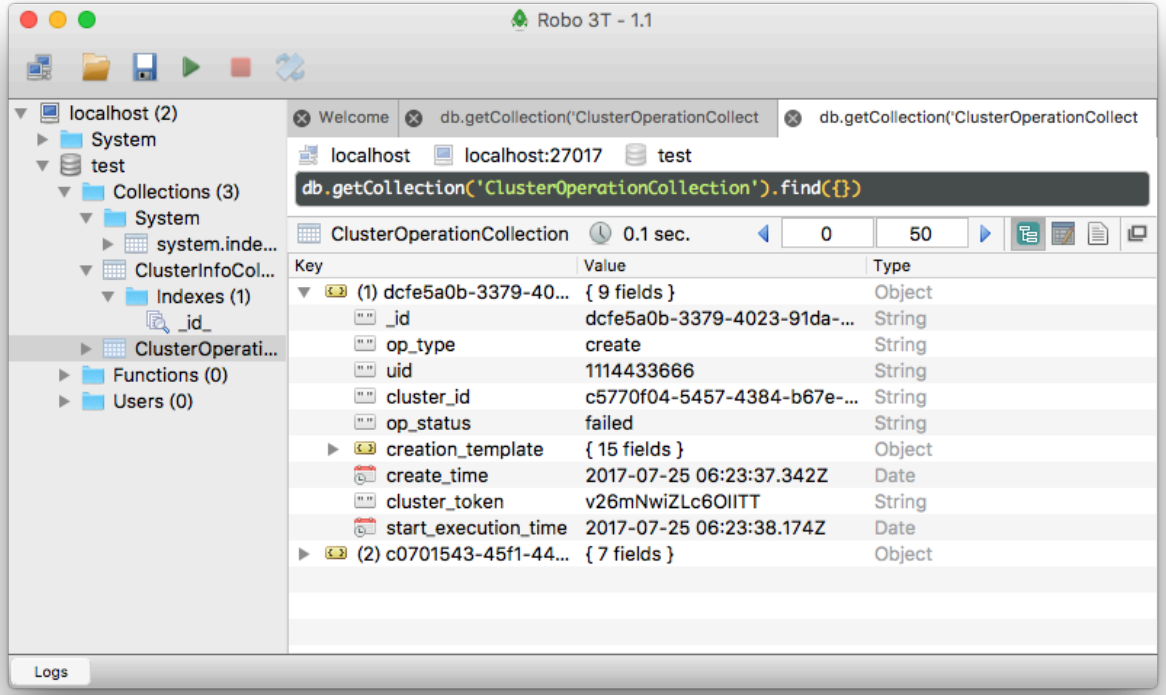
```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: wp-pv-claim
  labels:
    app: wordpress
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 20Gi
```

```yaml
apiVersion: v1
kind: Service
metadata:
  name: wordpress
  labels:
    app: wordpress
spec:
  ports:
    - port: 80
  selector:
    app: wordpress
    tier: frontend
  type: LoadBalancer
```

```yaml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: wordpress
  labels:
    app: wordpress
spec:
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: wordpress
        tier: frontend
    spec:
      containers:
      - image: wordpress:4.7.3-apache
        name: wordpress
        env:
        - name: WORDPRESS_DB_HOST
          value: wordpress-mysql
        - name: WORDPRESS_DB_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mysql-pass
              key: password.txt
        ports:
        - containerPort: 80
          name: wordpress
        volumeMounts:
        - name: wordpress-persistent-storage
          mountPath: /var/www/html
      volumes:
      - name: wordpress-persistent-storage
        persistentVolumeClaim:
          claimName: wp-pv-claim
```

Kubernetes 实践案例剖析-Wordpress

```
➜ tutorial kubectl --kubeconfig kubectl.cmkubeconfig port-forward cds-mongo-v3.0.5-fix-permission-hhf8n 28017 27017
Forwarding from 127.0.0.1:28017 -> 28017
Forwarding from [::1]:28017 -> 28017
Forwarding from 127.0.0.1:27017 -> 27017
Forwarding from [::1]:27017 -> 27017
```



Caicloud

cloud. acceleration. intelligence.

才云科技

谢谢大家！