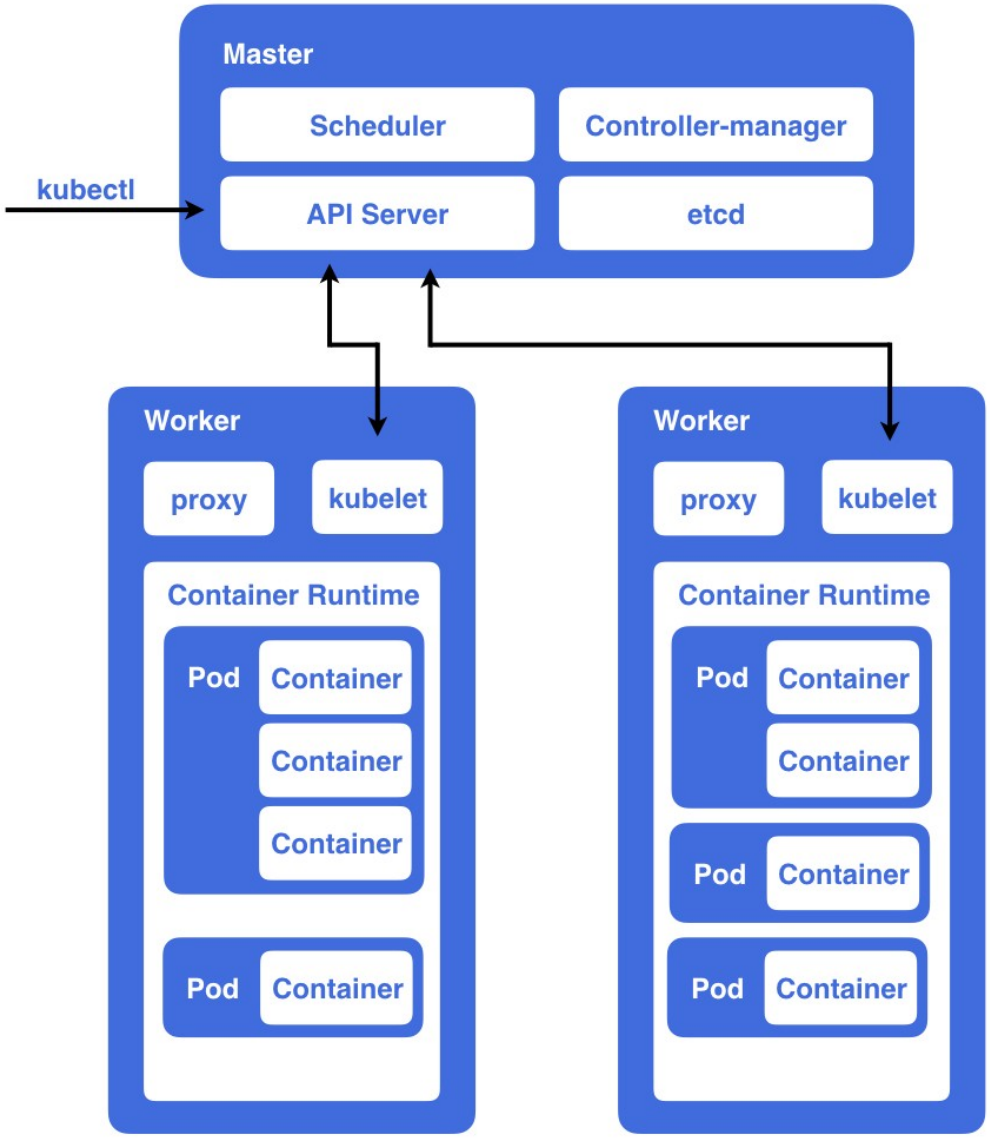
K8S 使用說明

架構

Cluster


Master
Scheduler
Controller-manager
kubectl
API Server
etcd
Worker
proxy
kubelet
Container Runtime
Pod
Container
Container
Container
Pod
Container
Worker
proxy
kubelet
Container Runtime
Pod
Container
Container
Pod
Container
Pod
Container

Deployment

設定 Pod 橫向擴展

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: jenkins
  labels:
    app: jenkins
spec:
  replicas: 1
  selector:
    matchLabels:
      app: jenkins
  template:
    metadata:
      labels:
        app: jenkins
    spec:
      containers:
        - name: jenkins
          image: jenkins:local
          ports:
            - containerPort: 8080
          volumeMounts:
            - mountPath: /var/jenkins_home
              name: jenkins
            - mountPath: /usr/java
              name: java
            - mountPath: /var/release
              name: release
          env:
            - name: JENKINS_OPTS
              value: {{.Values.jenkins.opts}}
            - name: TZ
              value: Asia/Taipei
      volumes:
        - name: jenkins
          hostPath:
            path: {{.Values.jenkins.filePath}}
            type: Directory
        - name: java
          hostPath:
            path: {{.Values.javaPath}}
```

```
          type: Directory
      - name: release
        hostPath:
          path: {{.Values.releasePath}}
          type: Directory
```

Service

```
設定 Pod 連線
port: Cluster ip 上的 port, 提供 Cluster 內部訪問
nodePort: 提供 Cluster 外部訪問
targetPort: Pod 上的 port

apiVersion: v1
kind: Service
metadata:
  name: jenkins
spec:
  type: NodePort
  selector:
    app: jenkins
  ports:
    - name: jenkins
      protocol: TCP
      port: 8080
      targetPort: 8080
      nodePort: {{.Values.jenkins.port}}
    - name: mongo
      protocol: TCP
      port: 8082
      targetPort: 8082
      nodePort: 8082
    - name: mysql
      protocol: TCP
      port: 3306
      targetPort: 3306
      nodePort: 3306
```

Ingress

```
route 服務, 轉發請求到對應 Service
```

Resource type

1. node
2. deployment(deploy)
3. pod
4. service(svc)
5. ingress
6. crd(CustomResourceDefinitions)

安裝

安裝 **kubeadm, kubelet, kubectl**

```
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add

sudo apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"

sudo apt-get update

sudo apt-get install kubeadm kubelet kubectl -y
```

忽略 **Swap** 錯誤

```
sudo vim /etc/systemd/system/kubelet.service.d/10-kubeadm.conf

新增 Environment="KUBELET_EXTRA_ARGS=--fail-swap-on=false"
```

建立 **master node**

```
sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=Swap

重新設定
sudo kubeadm reset
```

設定 **kubectl** 指令

```
mkdir -p $HOME/.kube

sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

sudo chown $(id -u):$(id -g) $HOME/.kube/config

移除
rm -rf $HOME/.kube
```

部署 **Pod** 網路

```
sudo kubectl apply -f
https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml

移除
sudo kubectl delete -f
https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
```

設定 **Pod** 可以調度到 **master** 上

kubectl taint node [NODE_NAME] node-role.kubernetes.io/master-

查詢 **node**
kubectl get nodes

安裝 **Helm**

curl https://baltocdn.com/helm/signing.asc | sudo apt-key add -

sudo apt-get install apt-transport-https --yes

echo "deb https://baltocdn.com/helm/stable/debian/ all main" | sudo tee /etc/apt/sources.list.d/helm-stable-debian.list

sudo apt-get update

sudo apt-get install helm

重命名命令

vim ~/.bashrc
新增 alias k=kubectl $*
新增 alias h=helm $*

/etc/sysctl.conf
新增 vm.max_map_count=655360

sudo sysctl -p

調整 **NodePort** 端口範圍

vim /etc/kubernetes/manifests/kube-apiserver.yaml
新增 --service-node-port-range=1-65535

指令

kubectl

```
kubectl get [TYPE] -o wide
kubectl describe ([TYPE] | [TYPE/NAME])
kubectl delete crd --all
kubectl create -f deployment.yaml
kubectl exec -it [Pod] bash
kubectl get pods --all-namespaces
k logs -f [Pod] [Container]
```

Helm

```
helm install [NAME] [CHART]
helm list
helm repo list
helm create [NAME]
helm upgrade [NAME]
```

```
netstat -tunlp | grep [PORT]
```

部署

/home/rd3admin/cid/k8s

~~Ingress~~

~~下載 **nginx-ingress**~~
~~helm repo add nginx-stable https://helm.nginx.com/stable~~
~~helm pull nginx-stable/nginx-ingress~~

~~helm install ingress ingress~~

Jenkins(Mongo Express)

~~建立 **image**~~
~~docker build -f Dockerfile_jenkins -t jenkins:local .~~

建立 **Secret** 物件
kubectl create secret docker-registry registrykey-01 --docker-server=127.0.0.1:8006 --docker-username=admin --docker-password=7ujm9ol. --docker-email=

helm install jenkins jenkins

SonarQube

helm install sonarqube sonarqube

Nexus

helm install nexus nexus

使用

~~設定 local DNS~~

~~/etc/hosts~~
~~新增 172.16.110.150 rd3server.com~~

~~Server URL~~

	~~Ingress~~	~~NodePort~~
~~Jenkins~~	~~rd3server.com~~ ~~rd3server.com/jenkins~~	~~172.16.110.150:9999~~
~~SonarQube~~	~~rd3server.com/sonarqube~~	~~172.16.110.150:30002/sonarqube~~
~~SonarQube - PostgreSQL~~		
~~SonarQube - Adminer~~		~~172.16.110.150:30003~~
~~Nexus~~		~~172.16.110.150:8083~~

備份復原

備份

/etc/crontab
bkK8S.sh

復原

/home/rd3admin
git clone https://172.16.10.90/git/cid

tar -xzvf dockerContainerData.FULL.tar.gz -C /

部署服務

helm uninstall $(helm ls -aq)

/mnt
sudo mkdir RD3

sudo mount -t cifs -o username="",password="",gid="rd3admin",uid="rd3admin" "//172.16.10.199/BonusWinner/部門/RD3" "/mnt/RD3"

/mnt/RD3
cp -R jdk1.8.0_261 /usr/java/

疑難雜症

docker image 自動被刪除

df 指令檢查硬碟空間是否不足

查看 node 狀態
k describe node rd3admin

重啟 **Dcoker** 服務
systemctl restart docker