



扫码添加小助手，发送“CKA”加群



# CloudNativeLives

Kubernetes管理员实训

# Kubernetes网络实训

华为云容器团队核心架构师 & CNCF社区主要贡献者倾力打造

# 大 纲

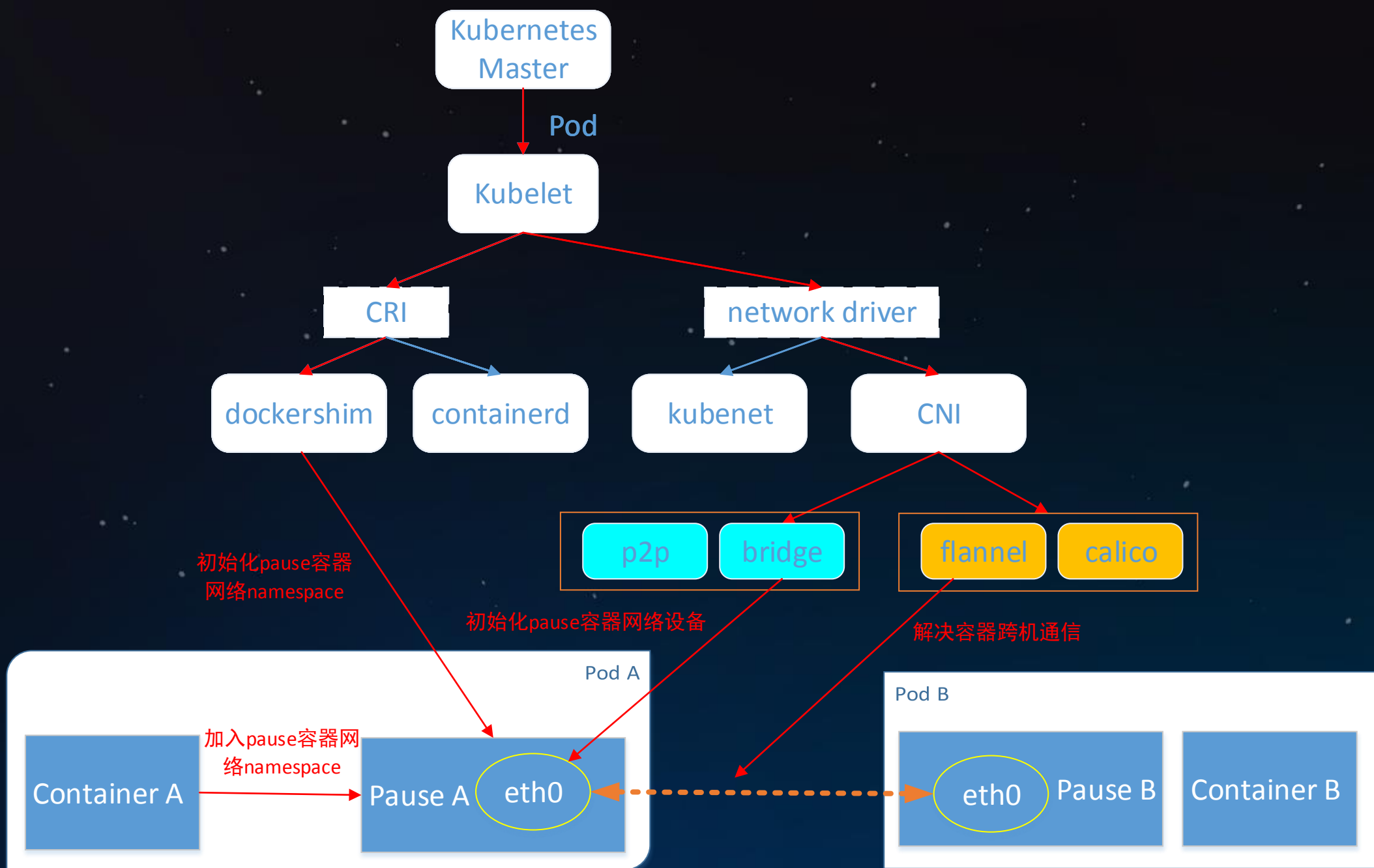
- Pod网络
- CNI
- Service概念
- 部署和配置网络load balancer
- Ingress概念
- 配置和使用集群DNS

# Pod网络

# Pod网络

- 一个Pod一个IP
  - 每个Pod独立IP，Pod内所有容器共享网络namespace（同一个IP）
  - 容器之间直接通信，不需要NAT
  - Node和容器直接通信，不需要NAT
  - 其他容器和容器自身看到的IP是一样的
- 集群内访问走Service，集群外访问走Ingress
- CNI（container network interface）用于配置Pod网络
  - 不支持docker网络

扁平网络：性能、可追溯、排错



# CNI

# CNI: Container Network Interface

- 容器网络的标准化
- 使用JSON来描述网络配置
- 两类接口：
  - 配置网络 -- 创建容器时调用  
`AddNetwork(net NetworkConfig, rt RuntimeConf) (types.Result, error)`
  - 清理网络 -- 删除容器时调用  
`DelNetwork(net NetworkConfig, rt RuntimeConf) error`

Container Runtime (e.g. k8s)

Container Networking Interface (CNI)

veth

macvlan

ipvlan

OVS

- Bridge
- PTP
- IPVLAN
- MACVLAN
- VLAN
- PORTMAP





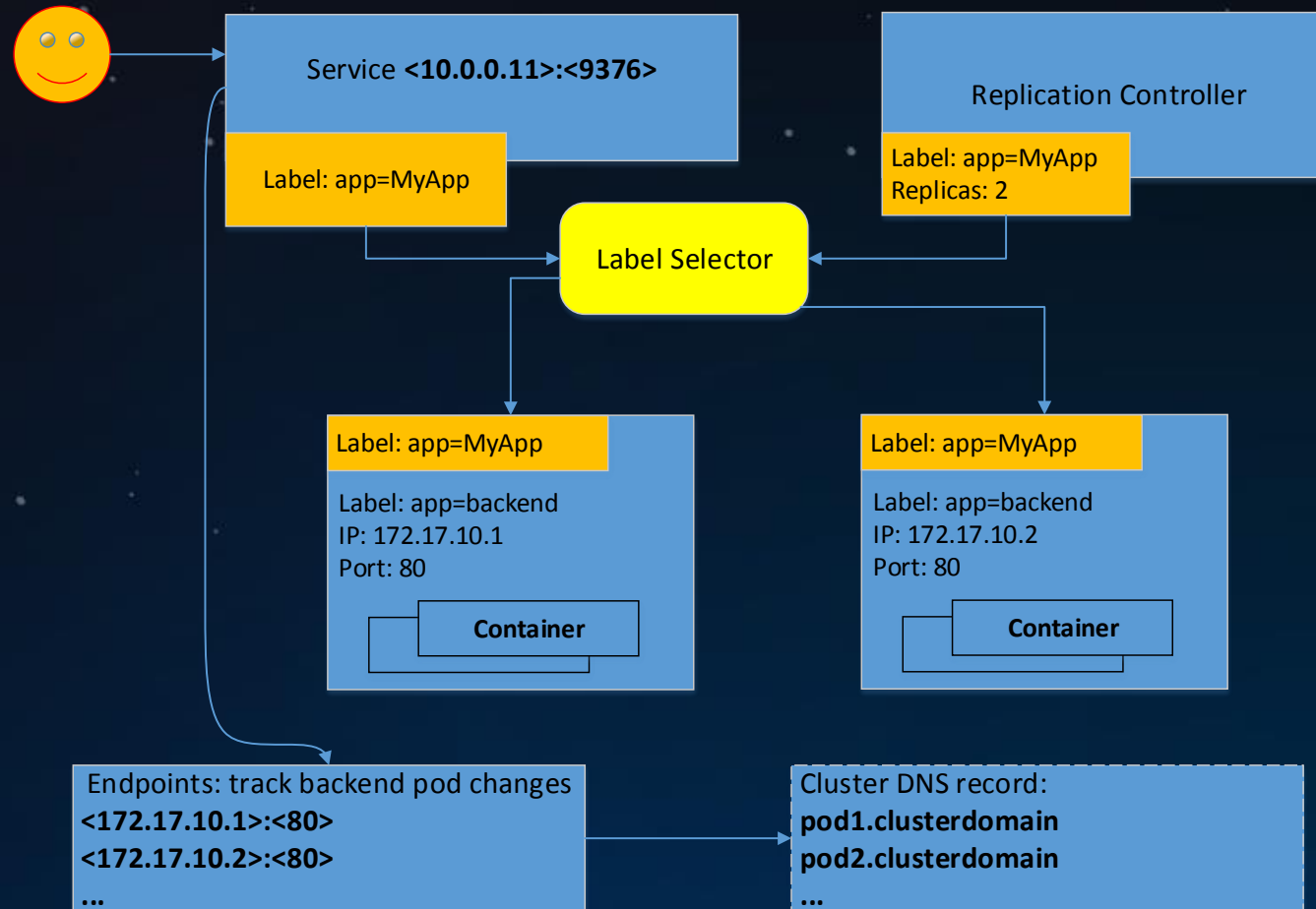
# CNI插件 : host-local + bridge

```
$ cat /etc/cni/net.d/10-mynet.conf
{
  "name": "mynet",
  "type": "bridge",
  "ipam": {
    "type": "host-local",
    "subnet": "10.10.0.0/16"
  }
}
```

CNI plugin二进制文件: /opt/cni/bin/{host-local, bridge...}

# Service

# Kubernetes Service



# Service和Endpoints定义

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: nginx-service
5    namespace: default
6  spec:
7    clusterIP: 10.101.28.148
8    ports:
9      - name: http
10        port: 80
11        protocol: TCP
12        targetPort: 8080
13    selector:
14      app: nginx
```

```
1  apiVersion: v1
2  kind: Endpoints
3  metadata:
4    name: nginx-service
5    namespace: default
6  subsets:
7    - addresses:
8      - ip: 172.17.0.2
9        nodeName: 100-106-179-237.node
10        targetRef:
11          kind: Pod
12          name: nginx-rc-c8tw2
13          namespace: default
14      - ip: 172.17.0.3
15        nodeName: 100-106-179-238.node
16        targetRef:
17          kind: Pod
18          name: nginx-rc-x14tv
19          namespace: default
20    ports:
21      - name: http
22        port: 8080
23        protocol: TCP
```

# 部署和配置网络load balancer

# LoadBalancer类型Service

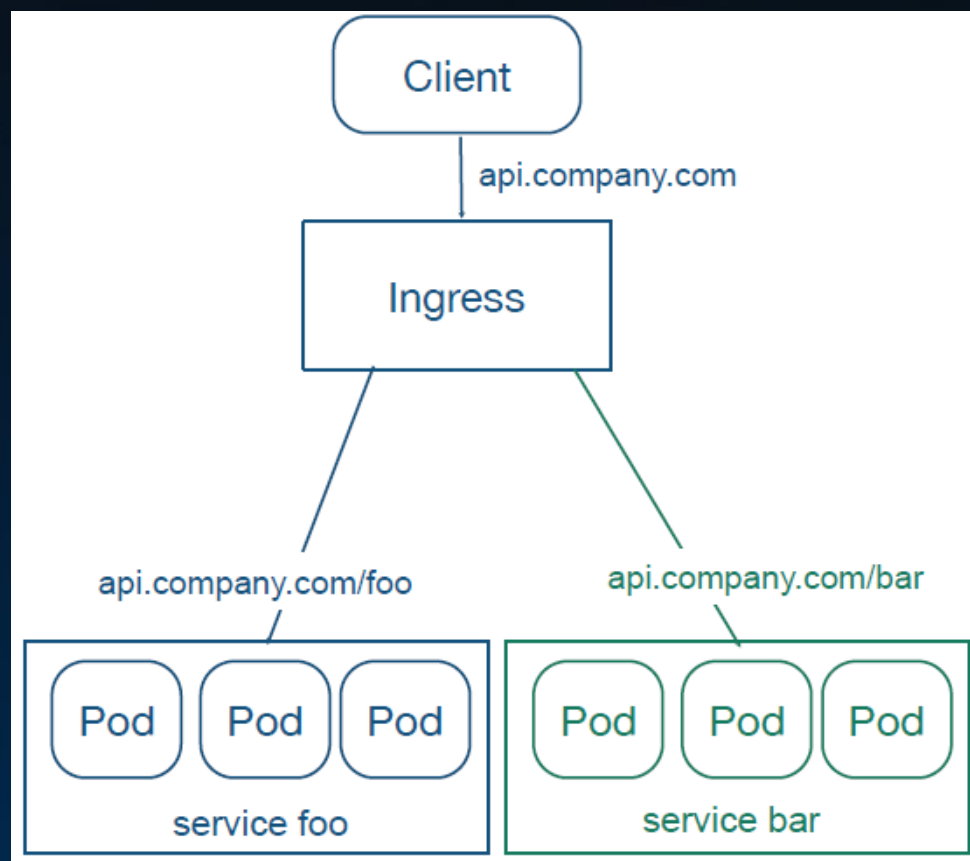
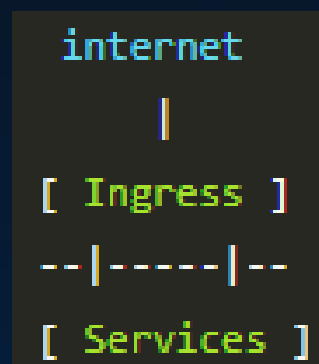
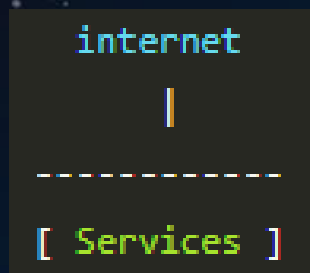
- 同时是Cluster IP类型
- 需要跑在特定的cloud provider上
  - Service Controller自动创建一个外部LB并配置安全组
  - 对集群内访问， kube-proxy用iptables或ipvs实现了云服务提供商LB的部分功能： L4转发， 安全组规则等。

```
kind: Service
apiVersion: v1
metadata:
  name: my-service
spec:
  selector:
    app: MyApp
  ports:
    - protocol: TCP
      port: 80
      targetPort: 9376
  clusterIP: 10.0.171.239
  loadBalancerIP: 78.11.24.19 #外部LB IP
  type: LoadBalancer
```

# Ingress

# Ingress

- Ingress是授权入站连接到达集群服务的规则集合
  - 支持通过URL方式将Service暴露到K8S集群外，Service之上的L7访问入口
  - 支持自定义Service的访问策略
  - 提供按域名访问的虚拟主机功能
  - 支持TLS





```
apiVersion:
extensions/v1beta1
kind: Ingress
metadata:
  name: test-ingress
spec:
  tls:
  - secretName: testsecret
  backend:
    serviceName: testsvc
    servicePort: 80
```

```
$ kubectl get ing
NAME          RULE    BACKEND    ADDRESS
test-ingress  -       testsvc:80  107.178.254.228
```

ADDRESS: Ingress的访问入口地址, 由Ingress Controller分配  
BACKEND: K8S Service + Port  
RULE: 自定义的访问策略。  
若规则为空, 则访问ADDRESS的所有流量都转发给BACKEND

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: test
spec:
  rules:
  - host: foo.bar.com
    http:
      paths:
      - path: /foo
        backend:
          serviceName: s1
          servicePort: 80
      - path: /bar
        backend:
          serviceName: s2
          servicePort: 80
```

当LB准备就绪时, Ingress Controller填充ADDRESS字段

```
$ kubectl get ing
NAME          RULE    BACKEND    ADDRESS
test          -       foo.bar.com
              /foo    s1:80
              /bar    s2:80
```

# DNS

# Kubernetes DNS

- 解析Pod和Service的域名的，K8S集群内Pod使用
- Kube-dns和CoreDNS
- 对Service

kubelet配置--cluster-dns把DNS的静态IP传递给每个容器

Kubelet传入--cluster-domain配置伪域名

- A记录

- 普通Service : my-svc.my-namespace.svc.**cluster.local** → **Cluster IP**

- headless Service : my-svc.my-namespace.svc.**cluster.local** → **后端Pod IP列表**

- SRV记录 :

- \_my-port-name.\_my-port-protocol.my-svc.my-namespace.svc.cluster.local →

## Service Port

- 对Pod

- A记录

- **pod-ip**.my-namespace.pod.cluster.local → Pod IP

1-2-3-4

# 课后作业

1. 创建1个Service和1个Pod作为其后端。通过kubectl describe获得该Service和对应Endpoints信息。
    - Service的名称为<hwcka-004-1-svc-你的华为云id>
    - Pod的名称为<hwcka-004-1-pod-你的华为云id>
    - 将所用命令、Service和Endpoints完整yaml截图上传
  2. 创建1个Service和1个Pod作为其后端。通过nslookup查询该Service的Pod的域名信息。
    - Service的名称为<hwcka-004-2-你的华为云id>
    - 将所用命令、Service和Pod域名信息截图上传
- 作业完成后，提交到论坛，包括完整的浏览器截图、华为云账号，作业中所创建的集群、应用名称要带hwcka前缀
  - **提交作业且答对的前50名，可获得满100减50的优惠券一张**



# Thank You

直播 每周四 晚20:00