

# Kubernetes 中的容量管理和实践

张晋涛

# 个人介绍



- 张晋涛
- Apache APISIX committer
- Kubernetes ingress-nginx reviewer
- 『K8S 生态周报』发起人
- GitHub: [tao12345666333](https://github.com/tao12345666333)





- 01 Kubernetes 集群中的资源
- 02 Kubernetes 集群中资源用量管理
- 03 Kubernetes 集群规模的设计
- 04 小容量多节点 vs 大容量少节点



# 01、K8s 集群中的资源



# K8s 相关的资源类型

## 不可变类型

- 硬件（容量上限）
- 网段（CIDR）
- ...

## 可变类型

- Pod
- Deployment
- Service
- ...

# K8s 相关的资源类型



## 共享资源

- CPU
- 内存
- ...

## 独占资源

- IP
- 磁盘
- ...

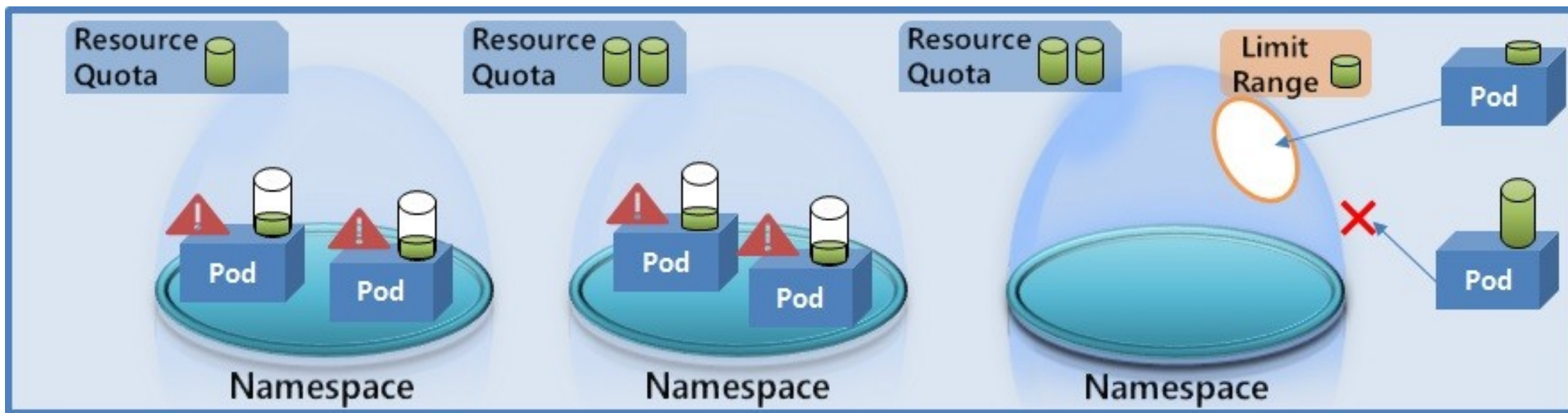


## 02、K8s 集群中资源用量管理

# K8s 中的容量管理



- Resource Quota、Limit Range 和 QoS







# K8s 中 Resource Quota

- 计算资源
- 存储资源
- 资源数量

```
1 cat <<EOF > object-counts.yaml
2 apiVersion: v1
3 kind: ResourceQuota
4 metadata:
5   name: object-counts
6 spec:
7   hard:
8     configmaps: "10"
9     persistentvolumeclaims: "4"
10    pods: "4"
11    replicationcontrollers: "20"
12    secrets: "10"
13    services: "10"
14    services.loadbalancers: "2"
15 EOF
16
17 kubectl create -f ./object-counts.yaml
```



# K8s 中 LimitRange

- Kubernetes v1.10
- 可设置默认策略

```
1 apiVersion: v1
2 kind: LimitRange
3 metadata:
4   name: mem-limit-range
5 spec:
6   limits:
7   - default:
8       memory: 512Mi
9     defaultRequest:
10       memory: 256Mi
11   type: Container
```



# K8s 中的资源用量

- Request
- Limit

```
1 apiVersion: v1
2 kind: Pod
3 metadata:
4   name: constraints-mem-demo-2
5 spec:
6   containers:
7   - name: constraints-mem-demo-2-ctr
8     image: nginx
9     resources:
10      limits:
11        memory: "1.5Gi"
12      requests:
13        memory: "800Mi"
```

# K8s 中的 QoS



## Guaranteed

- 所有 container 都且有 limits
- 所有 container 都有 requests 和 limits 且相等

## Burstable

- 只要有一个 container 的 requests 和 limits 不相等

## BestEffort

- 不设置 requests 和 limits



## 03、K8s 集群规模的设计



## K8s 集群的容量

- 每个节点不超过 110 个 Pod
- 不超过 5000 个节点
- 不超过 150000 个总 pod
- 不超过 300000 个容器

<https://kubernetes.io/docs/setup/best-practices/cluster-large/>

# K8s 单节点容量



## 节点

- Pod CIDR
- 通常使用两倍容量规避 IP 重复的问题

默认 110 个 Pod

- /24 CIDR
- 节点数  $N = 2^{(\text{节点 CIDR} - \text{Pod CIDR})}$

| 每个节点的最大 Pod 数量 | 每个节点的 CIDR 范围 | IP 地址数量 |
|----------------|---------------|---------|
| 8              | /28           | 16      |
| 9 - 16         | /27           | 32      |
| 17 - 32        | /26           | 64      |
| 33 - 64        | /25           | 128     |
| 65 - 110       | /24           | 256     |



# K8s 中的节点容量

需求：

- 每个节点 110 个 Pod
- Pod CIDR 172.16.0.0/12

解答：

- 节点 CIDR 掩码为 24
- Pod CIDR 掩码为 12
- $2^{(24-12)} = 4096$





# K8s 中的节点容量

需求：

- 每个节点 8 个 Pod
- Pod CIDR 172.16.0.0/16

解答：

- 节点 CIDR 掩码为 28
- Pod CIDR 掩码为 16
- $2^{(28-16)} = 4096$



# K8s 中 Service 的容量

- service CIDR 最小范围: /27
- service CIDR 最大范围: /16
- Service 数  $N = 2^{(32 - \text{service CIDR 掩码范围})}$



# K8s 中 Service 的容量

需求：

- 需要 4096 个 service

解答：

- $2^{(32-20)} = 4096$
- 所以 service 网段的掩码 /20



# K8s 集群容量限制

- etcd 建议的最大容量： 8G
- etcd 默认容量： 2G
- <https://etcd.io/docs/v3.5/dev-guide/limit/>

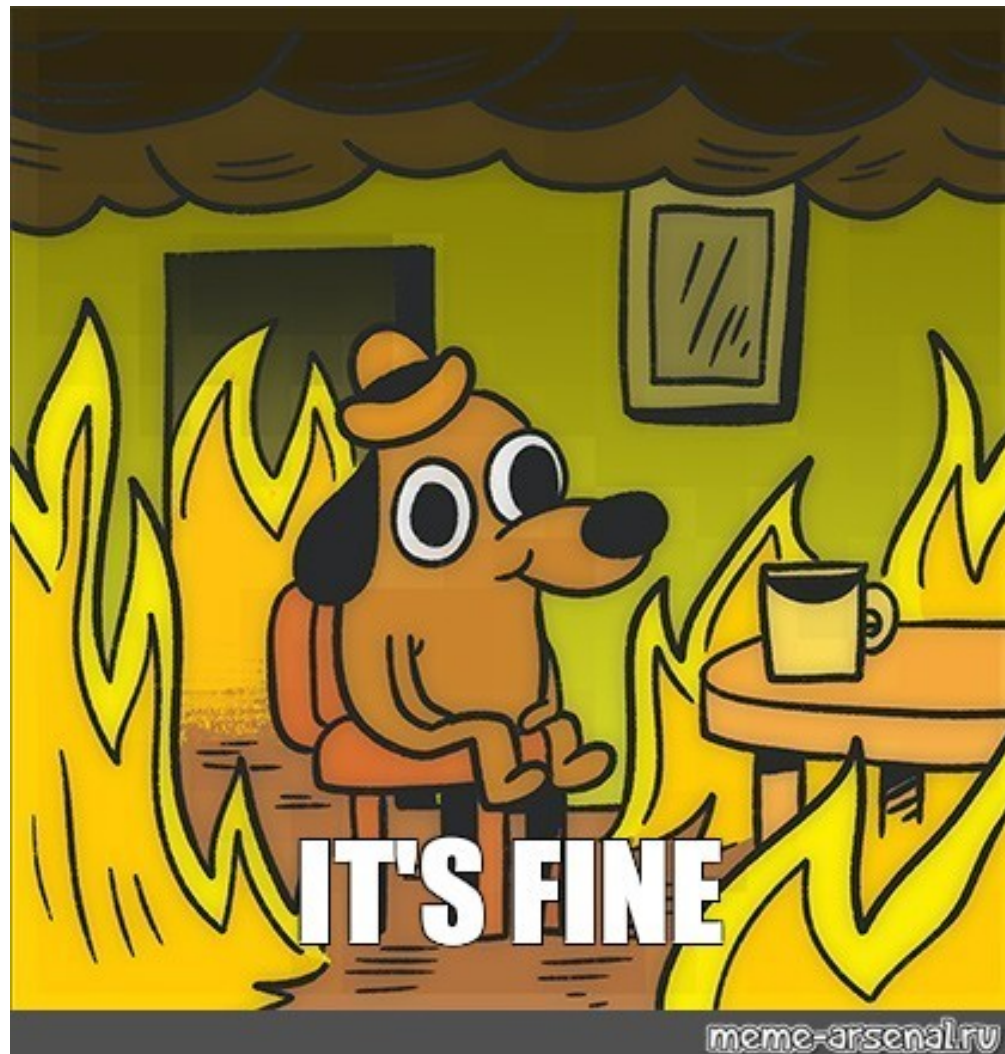


# 04、小容量多节点 vs 大容量少节点

# 优劣对比



- 管理成本
- 单节点成本
- 爆炸半径
- 资源伸缩量
- 副本数
- 资源利用率





北京开课吧科技有限公司  
[www.kaikeba.com](http://www.kaikeba.com)  
400-996-0826

**Thanks**