法律声明

- 本课件包括：演示文稿，示例，代码，题库，视频和声音等，讲师及小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。

- 课程详情请咨询
  - 微信公众号：小象
  - 新浪微博：ChinaHadoop

Kubernetes安全

目录

1. ServiceAccount vs UserAccount（认证）
2. RBAC（授权）
3. 多租系统的构建
4. 安全漫谈

1. UserAccount vs ServiceAccount

```
ubuntu@VM-24-197-ubuntu:~$ ls
aa  env  kubectl.deb  kubernetes.tar.gz  minikube  minikube-l
ubuntu@VM-24-197-ubuntu:~$ cd .kube/
ubuntu@VM-24-197-ubuntu:~/.kube$ ls
cache  config  http-cache  shell
ubuntu@VM-24-197-ubuntu:~/.kube$ cat config
apiVersion: v1
clusters:
- cluster:
    certificate-authority: /home/ubuntu/.minikube/ca.crt
    server: https://127.0.0.1:8443
  name: minikube
contexts:
- context:
    cluster: minikube
    user: minikube
  name: minikube2
current-context: minikube2
kind: Config
preferences: {}
users:
- name: minikube
  user:
    as-user-extra: {}
    client-certificate: /home/ubuntu/.minikube/client.crt
    client-key: /home/ubuntu/.minikube/client.key
```

1. UserAccount vs ServiceAccount

```
ubuntu@VM-24-197-ubuntu:~/.kube$ kubectl get pod db -o yaml | tail -40
  terminationGracePeriodSeconds: 30
  volumes:
  - name: secrets
    secret:
      defaultMode: 420
      secretName: mysecret
  - name: default-token-htsr7
    secret:
      defaultMode: 420
      secretName: default-token-htsr7
status:
  conditions:
  - lastProbeTime: null
```

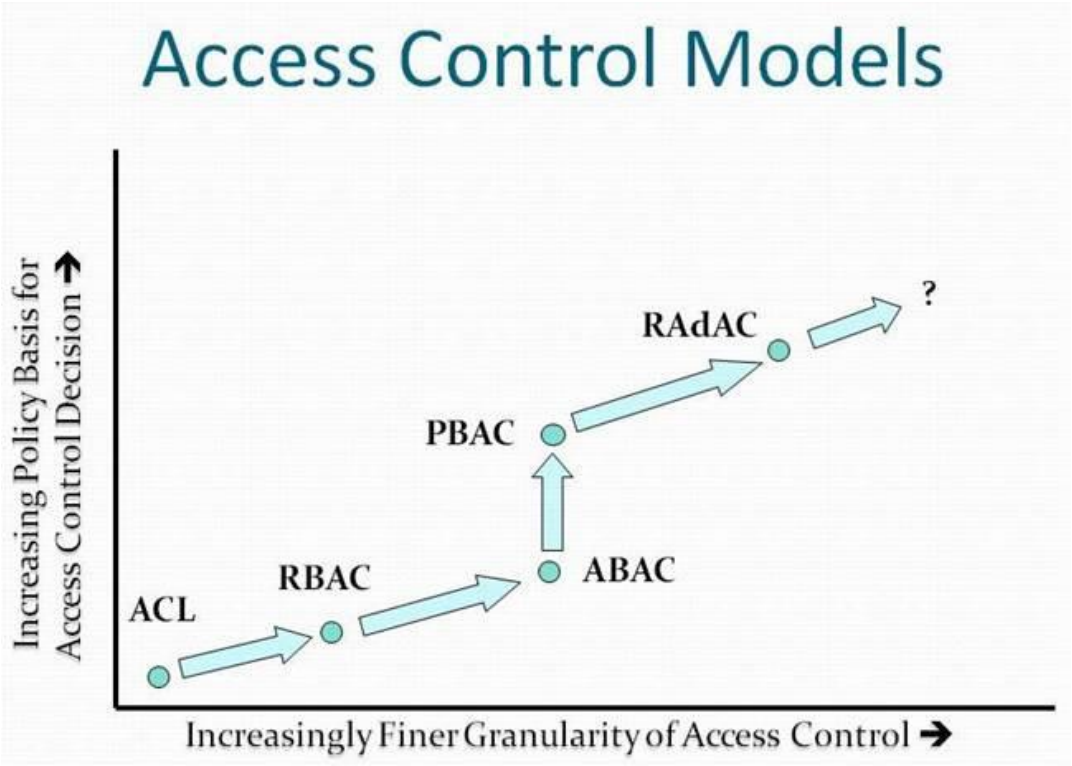
```
ubuntu@VM-24-197-ubuntu:~/.kube$ kubectl get serviceaccount default -o yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  creationTimestamp: 2018-01-23T03:09:08Z
  name: default
  namespace: default
  resourceVersion: "41"
  selfLink: /api/v1/namespaces/default/serviceaccounts/default
  uid: c74da6ac-ffea-11e7-abcd-5254003dc2cc
secrets:
- name: default-token-htsr7
```

```
ubuntu@VM-24-197-ubuntu:~/.kube$ kubectl get secret --all-namespaces
NAMESPACE     NAME                  TYPE                                  DATA      AGE
default       default-token-htsr7   kubernetes.io/service-account-token   3         42d
default       mysecret              Opaque                                2         10d
kube-public   default-token-l59jb   kubernetes.io/service-account-token   3         42d
kube-system   default-token-t9gvb   kubernetes.io/service-account-token   3         42d
ns-a          default-token-kk22z   kubernetes.io/service-account-token   3         27d
ubuntu@VM-24-197-ubuntu:~/.kube$ kubectl get secret kube-public
Error from server (NotFound): secrets "kube-public" not found
ubuntu@VM-24-197-ubuntu:~/.kube$ kubectl get secret default-token-l59jb --namespace=kube
NAME                  TYPE                                  DATA      AGE
default-token-l59jb   kubernetes.io/service-account-token   3         42d
ubuntu@VM-24-197-ubuntu:~/.kube$ kubectl get secret default-token-l59jb --namespace=kube
apiVersion: v1
```

1. UserAccount vs ServiceAccount

User Account	Service Account
为人设计，为客户端设计（kubectl/kubelet/controller/scheduler)	为Pod中的进程调用k8s API设计
跨namespace	仅当前namespace
.kube/config	Kubectl get servcieaccount Kubectl get secret
《Kubernetes集群安全配置案例》 《Kubernetes 认证》	《名词解释：Service Account》

2. 安全模型：一段历史



- 《权限系统设计模型分析》

2. Role/ClusterRole/RoleBinding/ClusterRoleBinding

Role: 角色/namespace内	ClusterRole： 角色/不区分namespace
RoleBinding 把role和[user/serviceAccount/group]关联起来	ClusterRoleBinding 把ClusterRole和[user/serviceAccount/group]关联起来

《RBAC——基于角色的访问控制》

2. namespace/非namespace

namespace	跨namespace
• configmaps (aka 'cm') • daemonsets (aka 'ds') • deployments (aka 'deploy') • endpoints (aka 'ep') • events (aka 'ev') • jobs • pods (aka 'po') • replicasets (aka 'rs') • ….	• 集群范围资源（例如节点，即node） • 非资源类型endpoint（例如”/healthz”） • 跨所有命名空间的命名空间范围资源（例如pod，需要运行命令kubectl get pods --all-namespaces来查询集群中所有的pod）

2. Role的例子之一

```
kind: Role
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  namespace: default
  name: configmap-updater
rules:
- apiGroups: [""]
  resources: ["configmap"]
  resourceNames: ["my-configmap"]
  verbs: ["update", "get"]
```

3. 多租是什么？——以水电煤为例

云计算中的多租户

“多租户”也是“云计算”的基本属性之一。云计算的三种服务层次——SaaS、PaaS和IaaS均体现了对“多租户”不同的支持。

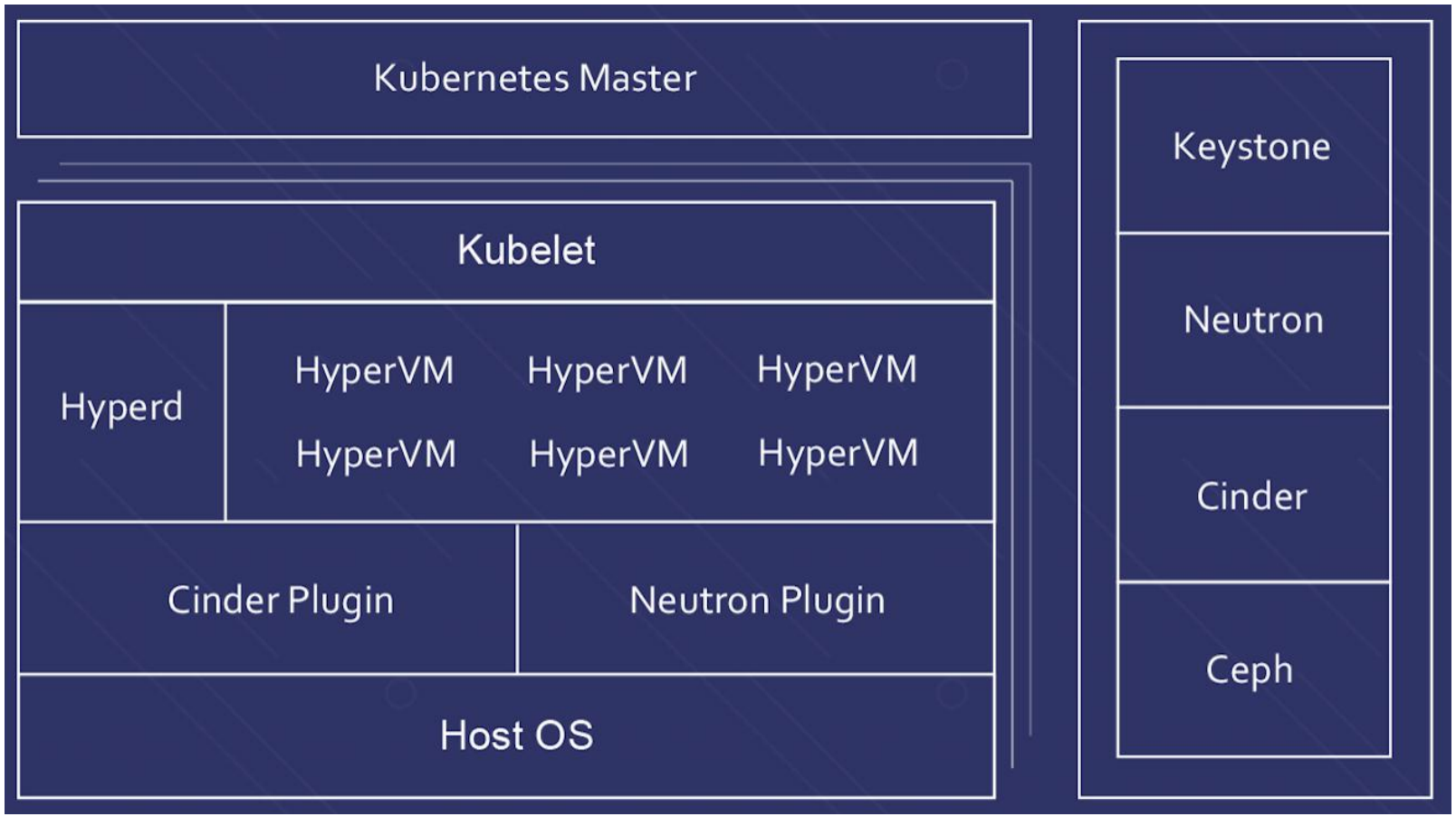
	出租的资源	举例说明
SaaS	软件的使用权。	典型如：电子邮件系统。用户（租户）拥有使用账号。租户登录使用系统。
PaaS	软件开发平台资源（如开发支撑系列工具，应用存储空间，运行容器，平台服务等等）。	如：某租户拥有1G应用存储空间，应用容器（数量不限，总内存上限4G），2个缓存服务。
IaaS	硬件基础设施（如CPU、内存，存储，IP，网络设备等等）。	如：某租户拥有2颗CPU，8G内存，80G硬盘，10IP，2负载均衡器，创建主机数量不限

用户vs租户　　公有云vs私有云

3. k8s技术栈的多租户

- API请求多租户
- 网络/存储隔离
- 容器运行时隔离

3. k8s技术栈的多租户



《Hypernetes简介——真正多租户的Kubernetes Distro》

4.什么是计算机系统的安全？

- 想多了……老师没那么牛
- 可以看看这个项目，用list check方式对比安全

https://github.com/qazbnm456/awesome-web-security

- 把握安全的基本原则，就是少犯低级错误，至于高级错误，那个成本就大的去了～

联系我们

小象学院：互联网新技术在线教育领航者

- 微信公众号：大数据分析挖掘
- 新浪微博：ChinaHadoop