Kubernetes安全和多租支持

@kevin-wangzefeng

多租安全的基本要素

- 基础安全
  - 阻止匿名/未授权访问，区别对待一些用户（如管理员）
- 可见性隔离
  - 知道你能知道的，看不到他人的应用和数据
- 避免干扰
  - 分工明确，各司其职
  - 做你能做的，用你能用的，防止越权

Kubernetes多租常见形态

形态1:小公司内部共享平台

1. 简单的租户定义
2. 角色划分较少，权限粒度粗放
3. 所有用户来自企业内部，数据敏感性要求单一
4. 企业内部应用，业务多样性及复杂度低

形态2:大公司内部共享平台

1. 组织结构复杂，租户粒度多样，存在嵌套关系

2. 有用户权限控制，租户资源限额等要求

3. 企业内部应用，业务多样性及复杂度低

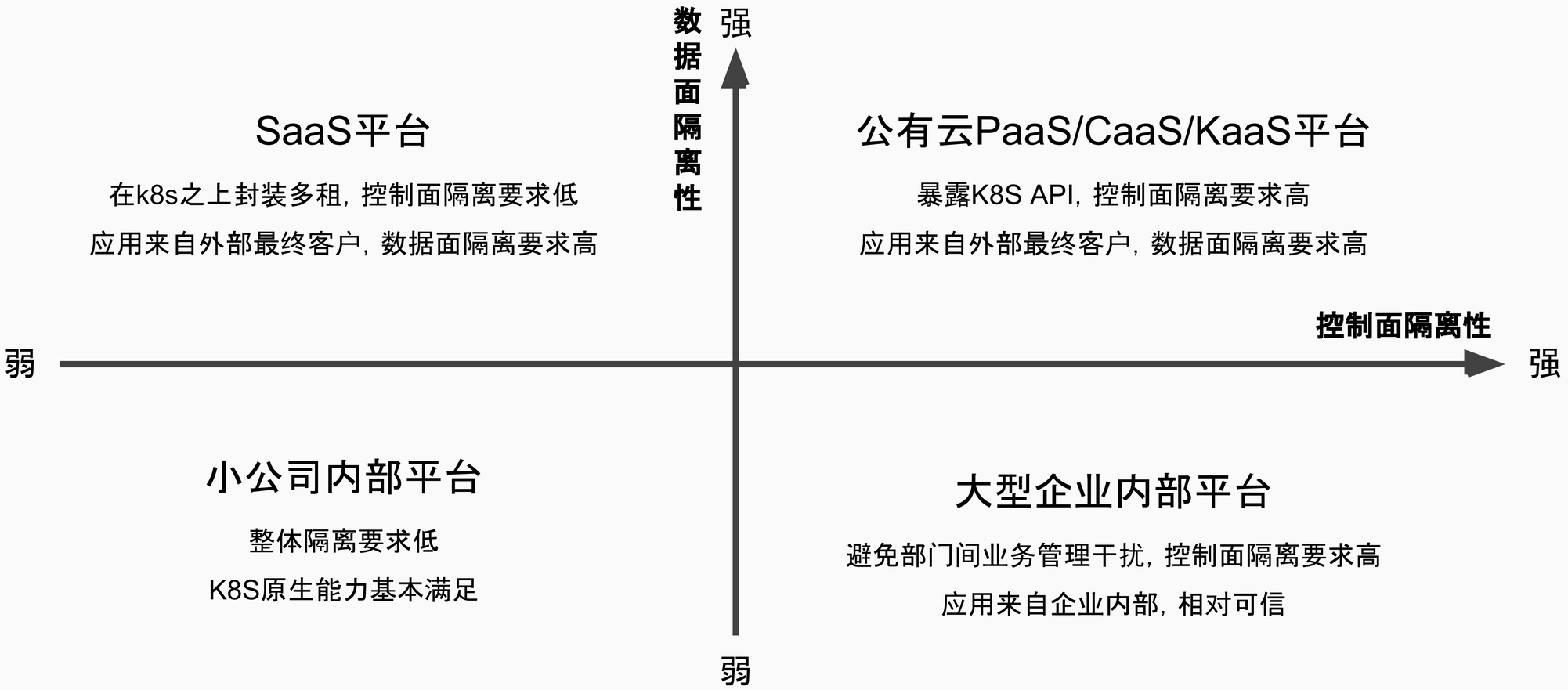
形态3：公有云SaaS平台

1. 基于kubernetes二次开发，不暴露k8s API
2. 在K8S之上封装多租模型，K8S不感知租户概念
3. 应用来自最终用户，可能包含恶意代码

形态4：公有云PaaS/CaaS/KaaS平台

1. 提供K8S原生API，或在其基础上扩展
2. K8S感知租户
3. 应用来自最终用户，可能包含恶意代码
4. 需要防范攻击者、异常应用等

多租场景的隔离性要求


数据面隔离性
强
弱
控制面隔离性
弱
强
SaaS平台
在k8s之上封装多租，控制面隔离要求低
应用来自外部最终客户，数据面隔离要求高
公有云PaaS/CaaS/KaaS平台
暴露K8S API，控制面隔离要求高
应用来自外部最终客户，数据面隔离要求高
小公司内部平台
整体隔离要求低
K8S原生能力基本满足
大型企业内部平台
避免部门间业务管理干扰，控制面隔离要求高
应用来自企业内部，相对可信

如何构建k8s多租集群

- 主要问题
  - 租户定义及API访问控制
  - 节点隔离和Runtime安全
  - 网络隔离

租户定义及API访问控制

- 自定义Tenant API v.s. 套用Namespace
- 隔离的颗粒度
- Namespace，租户安全的边界基础
    - 独立的对象命名空间（Namespaced APIs）
    - RBAC, ResourceQuota, LimitRange...
- Root scope API的处理
    - 对普通用户屏蔽 —— 简单，普遍做法
    - 自定义tenant api实现分组 —— 功能强大，实现上容易失控
- RBAC, RBAC and RBAC...

节点隔离和Runtime安全

- 控制面 v.s. 数据面
  - 为控制面组件预留足够资源
- 租户节点隔离
  - Resource Pools align with different plans
  - untrusted code with docker/containerd/runC
- 或者，安全容器
  - KataContainer

网络隔离

- 租户间网络隔离
  - 配置NetworkPolicy（隔离性弱，实现成本低）
  - 多网络平面 & Network API（隔离性强，实现成本高）
- DNS及服务发现
  - 每个租户/namespace一个dns

快速配置一个多租集群

- 基础安全
  - 禁用: web UI、legacy auth、basic auth、client cert generation
  - 启用审计日志，
  - 启用Admission Controller
    - NamespaceLifecycle
    - NodeRestriction
    - DenyEscalatingExec
    - PodTolerationRestriction
  - 开启PodSecurityPolicy，参考配置
- 可见性隔离
  - 配置RBAC，普通用户只能访问自己namespace下的对象，详见指导文档
- 避免干扰
  - 控制面请求限流
  - 限制集群级别API权限，普通用户只能访问自己的namespace
  - 禁用AutomountServiceAccountToken
  - 启用Admission Controller
    - AlwaysPullImages
    - LimitPodHardAntiaffinity
    - NamespaceExists
    - PodTolerationRestriction
  - 每个Namespace相关默认配置（resourceQuota, NetworkPolicy，role & role binding）

多租在华为PaaS的演进

Past - Sole tenancy with dedicated nodes

Some early releases for internal and on-perm

Now - Secure container with multi-networks

Huawei Cloud - Cloud Container Instance (https://www.huaweicloud.com/product/cci.html)

Future - Hard tenancy with Tenant API

社区后续关键特性

- Secure Container Isolation, Sandbox API
- Tenant API
- 多租网络隔离

Thanks

References:

- https://groups.google.com/forum/#!forum/kubernetes-wg-multitenancy
- https://docs.google.com/document/d/1PjIsBmZw6Jb3XZeVyZ0781m6PV7-nSUvQrwObkvz7jg
- https://docs.google.com/document/d/15w1_fesSUZHv-vwjiYa9vN_uyc--PySRoLKTuDhimjc
- https://docs.google.com/document/d/1jAcsC4sLgEV9__TdgJrMvPa3G73G62tFtMcKQgeIlHM