

Openstack与Kubernetes融合下的容器管理





01

方案背景

02

融合方案

03

Cetus 详析

04

案例分享

方案背景



K8S-on-openstack



openstack-on-K8S

IaaS事实标准

全球流行的IaaS管理平台，占据了开源IaaS江湖老大的地位

容器编排事实标准

K8S在容器编排引擎的竞争中脱颖而出，成为事实标准

K8S-on-opensack

- ✓ 提高部署效率
- ✓ 提高集群管理能力
- ✓ 多资源融合管理
- ✓ 统一认证体系

openstack-on-K8S

- ✓ 提高部署效率
- ✓ 环境隔离
- ✓ 提高服务可靠性

社区方案

社区中与容器相关的项目有magnum、zun、Kolla、Kuryr。

- **Magnum**，是OpenStack中一个提供容器集群部署的服务，通过Heat部署虚拟机和物理机，组成集群，然后调用COE接口完成容器的部署。
- **Kolla**，利用容器来简化 OpenStack 部署
- **Zun**，将容器作为Openstack管理的资源，为用户提供了创建和管理这些容器的接口，集成neutron、glance、cinder等，zun尚不成熟，功能简陋，不准备实现COE提供的很多先进的功能（例如容器保活、负载均衡等），而是提供基本的容器操作（CRUD），无法满足融合需求。
- **Kuryr**， Docker 的网络插件，可以让 Docker 使用 Neutron 的网络

在浪潮云产品推广过程中遇到客户需求，使用IaaS平台资源自动化创建出Kubernetes集群，并基于该集群实现容器应用、容器镜像管理，社区方案均无法满足，所以提出当前方案



01

方案背景

02

融合方案

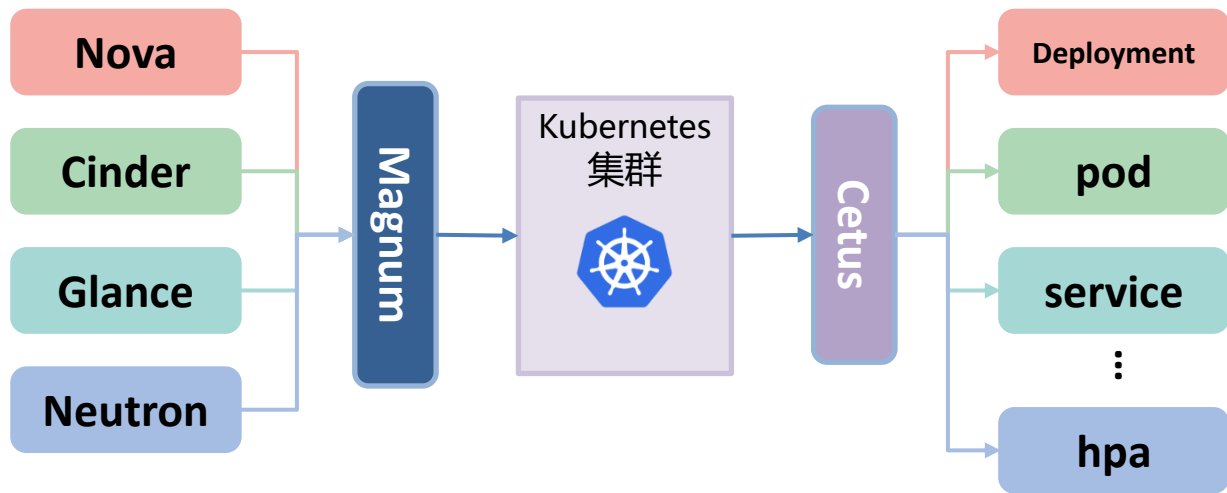
03

Cetus 详析

04

案例分享

融合方案

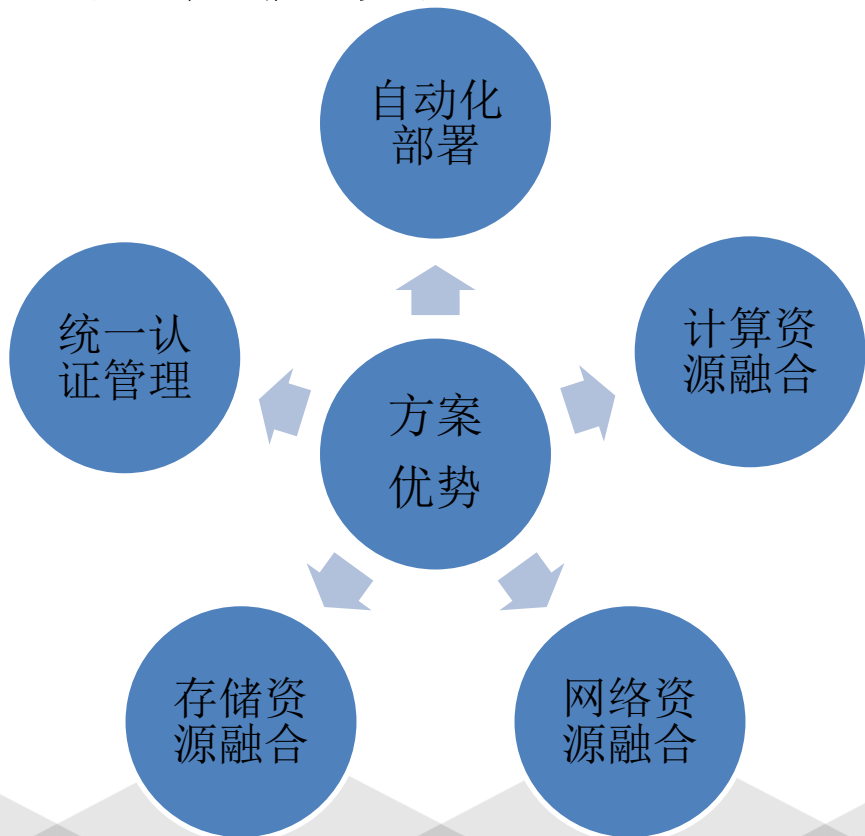


使用社区组件magnum实现Kubernetes集群管理，其中包括部署、规模调整等。

使用浪潮自研组件cetus实现对Kubernetes中资源的管理，将Kubernetes中

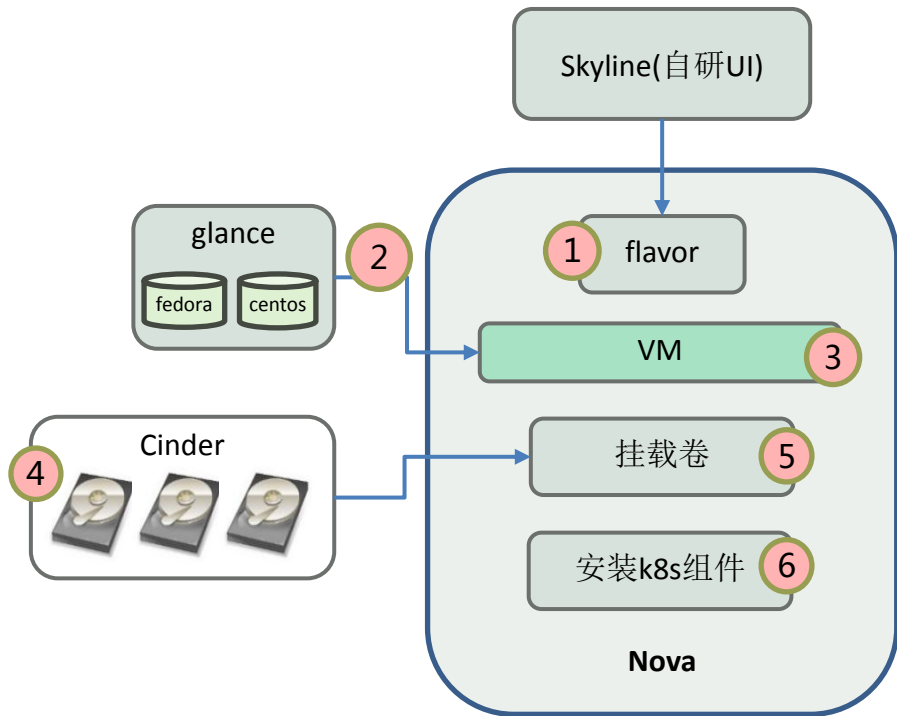
的各种特性包括容器应用保活、服务发现、弹性伸缩等体现在openstack页面中，用户能够简单易用地在magnum创建出来的容器集群上创建应用，进而实现在openstack中对虚拟机资源和容器资源统一管理，达到融合的目的。

方案优势



- 自动化部署，提高部署效率，降低产品使用门槛
- 计算资源融合，实现容器和虚拟机多粒度计算资源统一管理
- 网络资源融合，轻松解决在复杂业务网络中虚拟机与容器通信问题
- 存储资源融合，实现容器资源持久化，与虚拟机采用相同存储资源池，统一管理
- 统一认证管理，减少多套用户体系之间相互认证的烦扰，提高资源管理便捷性

计算存储资源融合



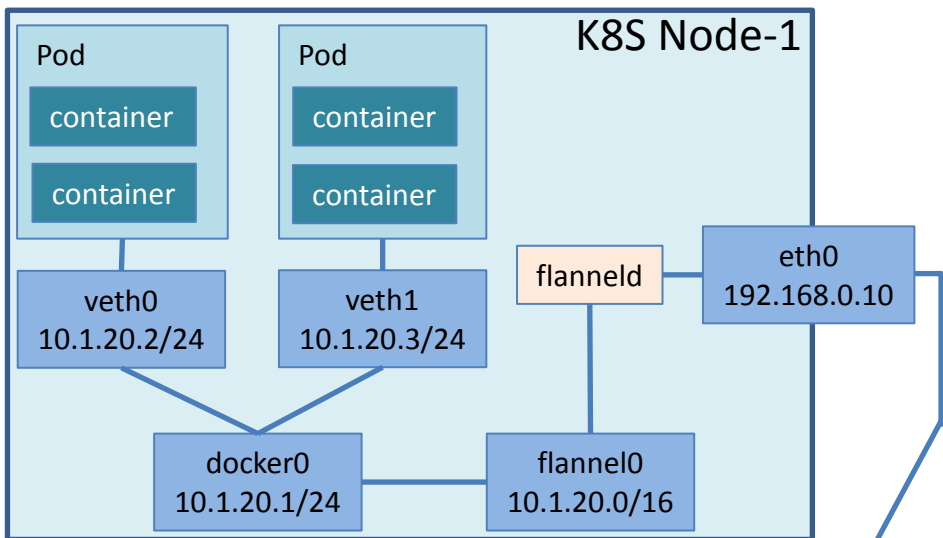
Kubernetes集群创建步骤:

1. 创建所需Flavor
2. 在glance中拉取所需镜像
3. 创建集群虚拟机
4. 创建数据卷
5. 挂载数据卷
6. 安装k8s组件及服务

在创建过程中虚拟机为Kubernetes集群的宿主机，承载各组件和服务的运行，完成服务提供，并将Kubernetes系统数据落地到Cinder的卷上，实现了计算存储资源的融合。

网络资源融合

部署出Kubernetes集群，在业务网中，同时为其绑定浮动IP暴露到管理网，Kubernetes可以将容器服务以NodeIP+Port的形式将服务同时暴露到业务网和管理网，实现业务网络上虚拟机与容器网络连通，同时在管理网对其管理，实现网路融合



Openstack业务网络

Openstack管理网络





01

方案背景

02

融合方案

03

Cetus 详析

04

案例分享

CETUS详析

功能地图

容器实例
管理



应用管理

容器保活

网络管理

无状态服务

容器卷

日志管理

容器弹性伸缩 (HPA)

容器监控

负载均衡

健康检查

容器镜像
管理



镜像版本管理

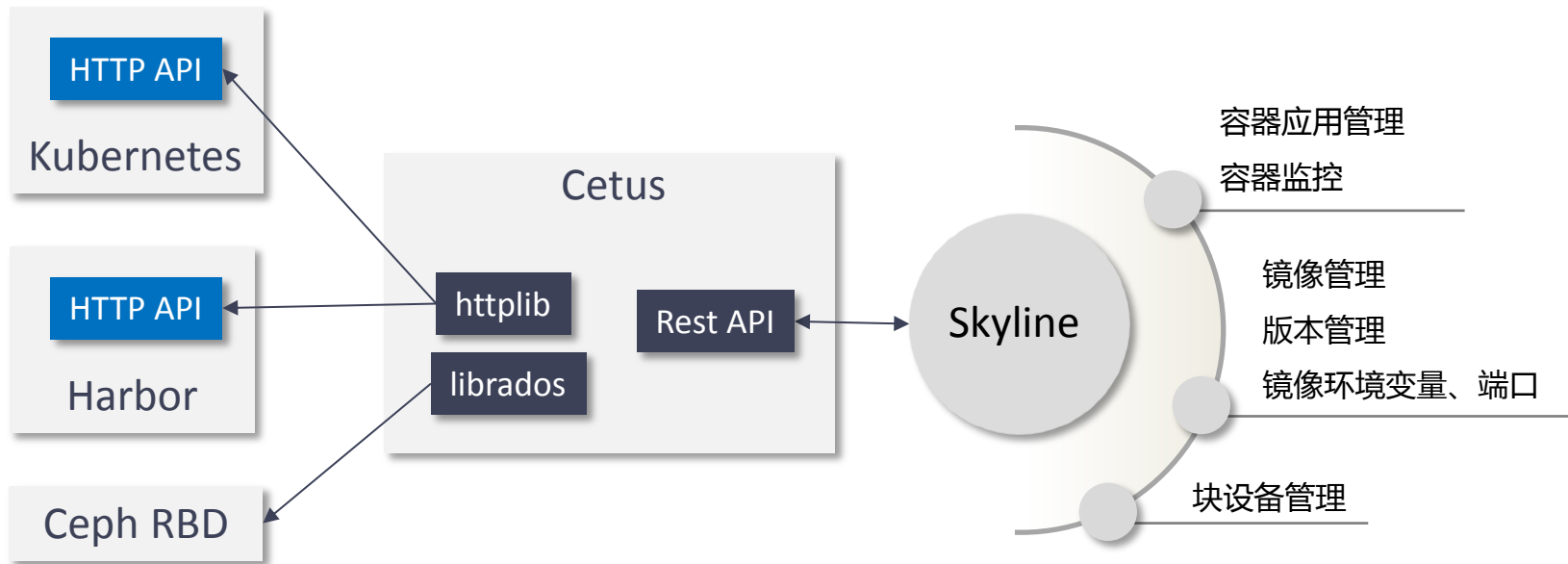
镜像管理

镜像分组管理

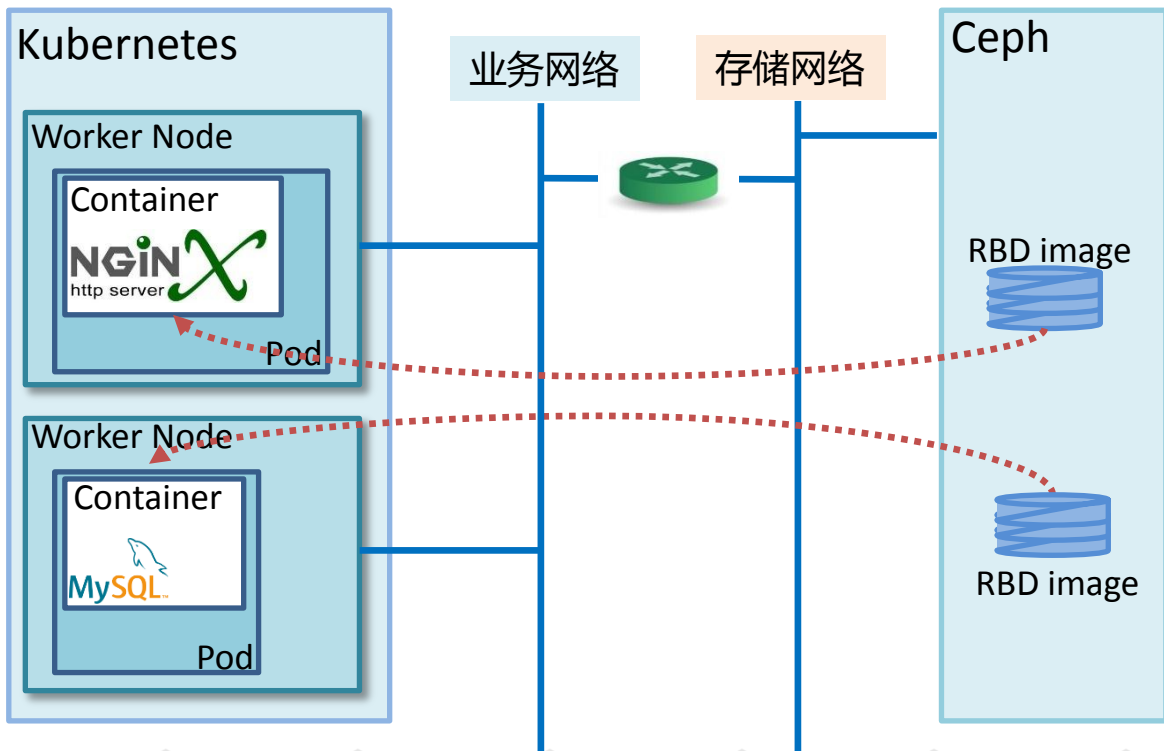
Cetus组件提供以下功能服务

- 容器应用管理
- 容器块设备管理
- 容器监控
- 容器镜像管理

CETUS架构

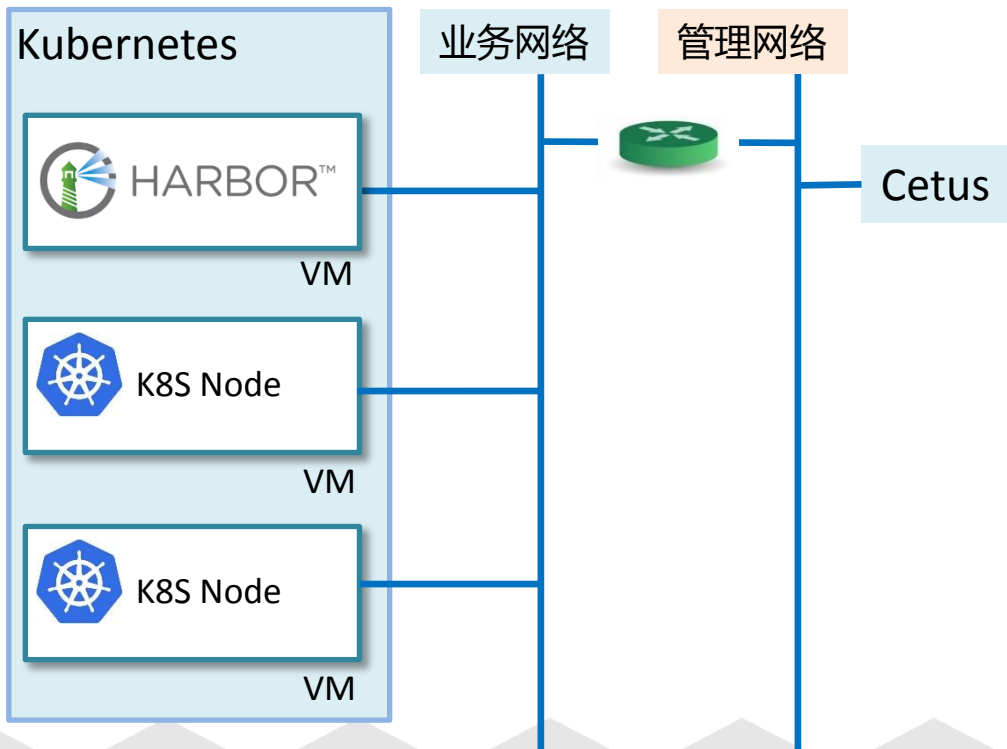


容器卷



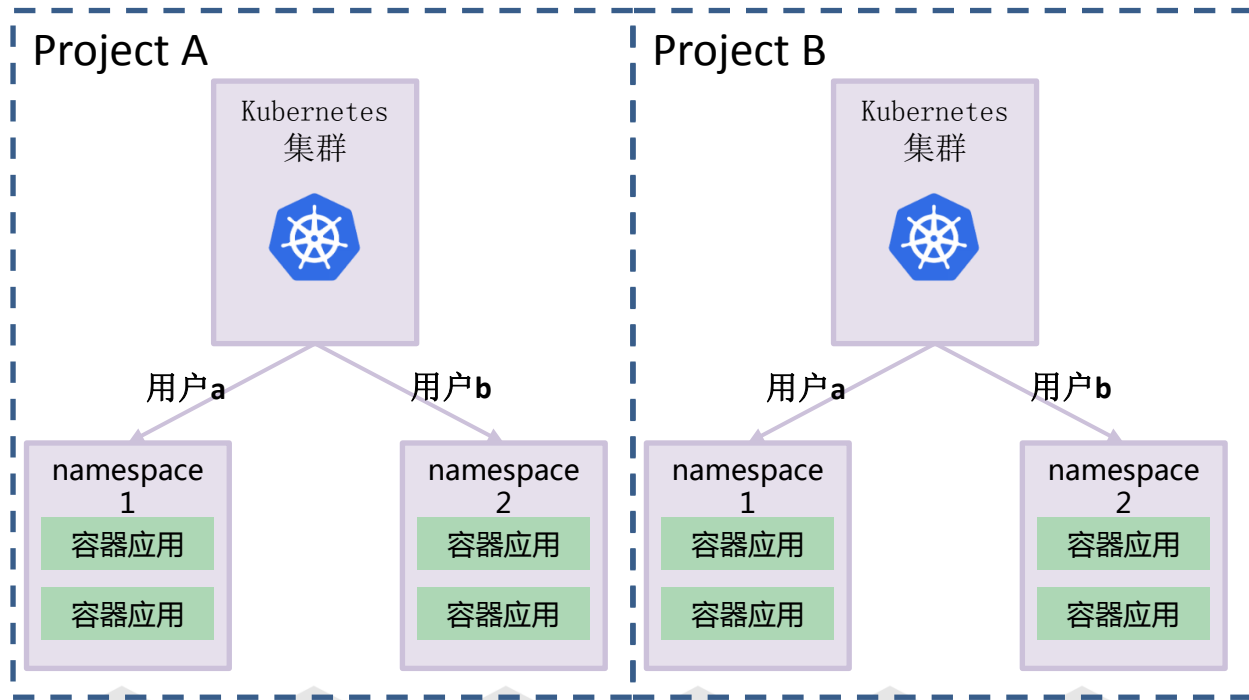
- 使用Ceph RBD作为容器的持久化卷
- Cetus调用Ceph实现对RBD设备的创建、删除操作，并调用Kubernetes API实现对容器卷的挂载、卸载操作
- 页面直观展示Kubernetes应用与容器卷的挂载关系以及挂载状态

容器镜像管理实现



- 使用提前准备好的qcow2虚拟机镜像创建虚拟机，完成相关配置，将其作为Harbor服务器
- Harbor服务器与Kubernetes集群使用相同的业务网络，即节省了网络资源，又确保了Harbor服务器的数据安全
- Cetus维护Harbor镜像库的基本信息，包括IP、端口、用户名、密码等
- Cetus调用Harbor API实现对镜像组、镜像、镜像版本的管理，并能够获取镜像的环境变量、端口等元数据信息

租户资源隔离



- 基于magnum将Kubernetes集群部署到不同的project中，实现project间的资源隔离，同时将使用的资源限制在project内
- 借助Kubernetes中namespace特性，将project中不同用户创建的容器应用归属到不同namespace实现用户级资源隔离

容器监控



Grafana

Grafana数据图形化展示



InfluxDB数据存储



Heapster数据汇聚



cAdvisor

cAdvisor数据采集



01

方案背景

02

融合方案

03

Cetus 详析

04

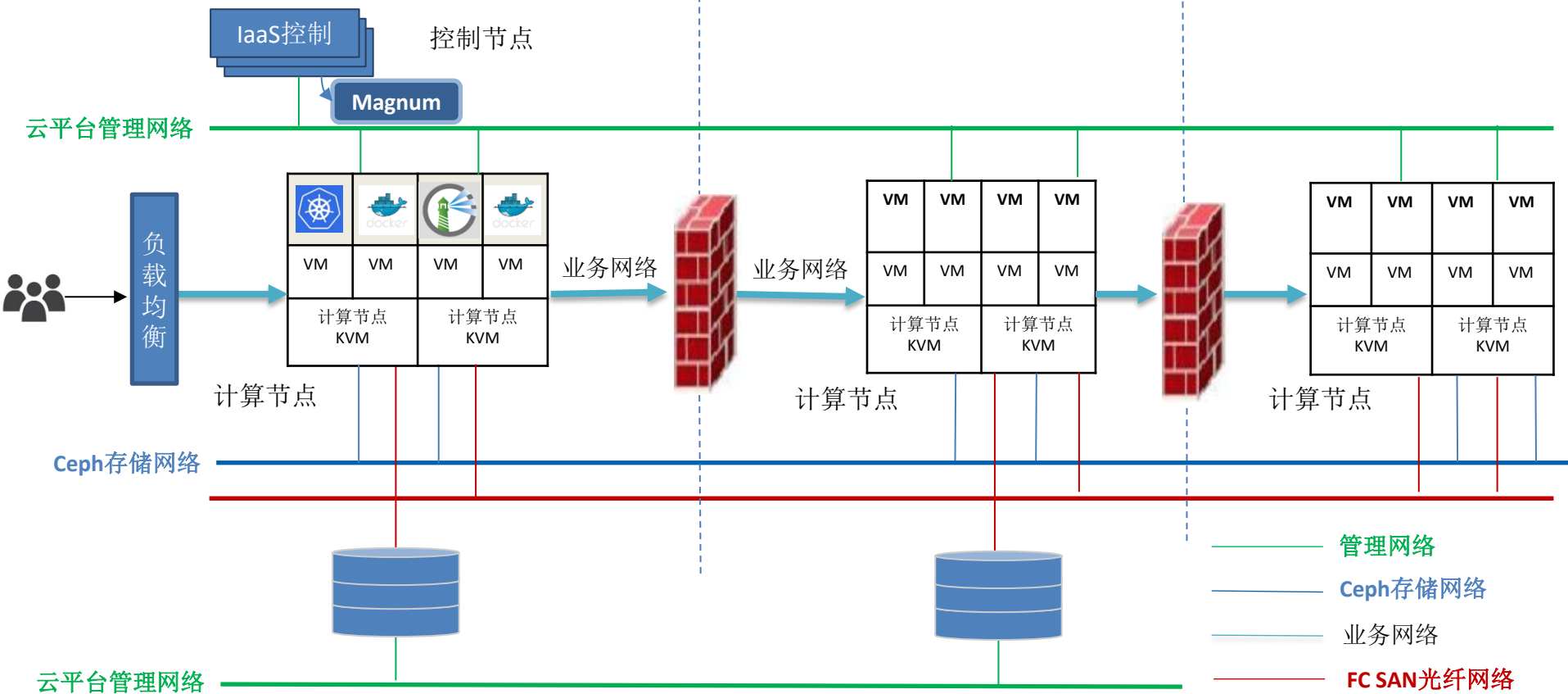
案例分享

案例网络模型

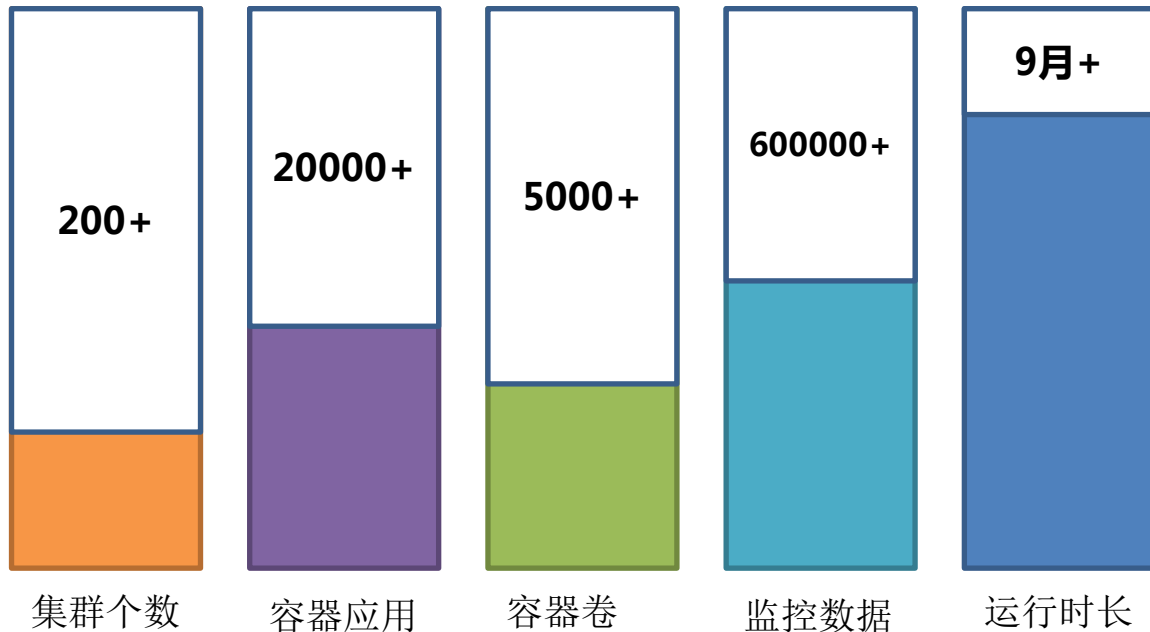
WEB区

APP区

堡垒区



应用案例



应用案例

- 集群个数200+
- 容器应用20000+
- 容器卷5000+
- 监控数据600000+
- 正常运行9月+

Thank You

