LINUXCON containercon CHINA 中国 CLOUDOPEN

THINK OPEN

开放性思维

Build Container Instance Service

腾讯云基础PaaS团队 洪志国

第一节

多种容器产品形态对比

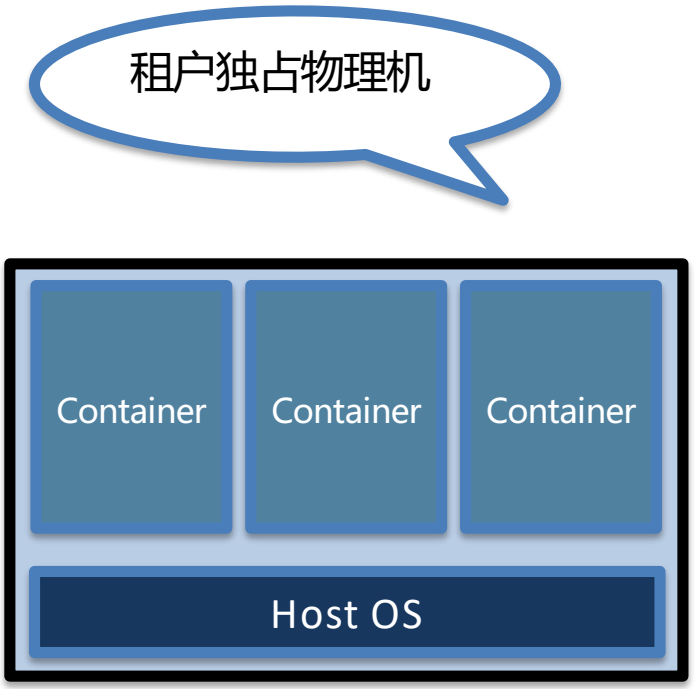
TKE@黑石

优点：

- 资源开销小
- 没有租户间共享

缺点：

- 节点管理, 扩缩容都不灵活



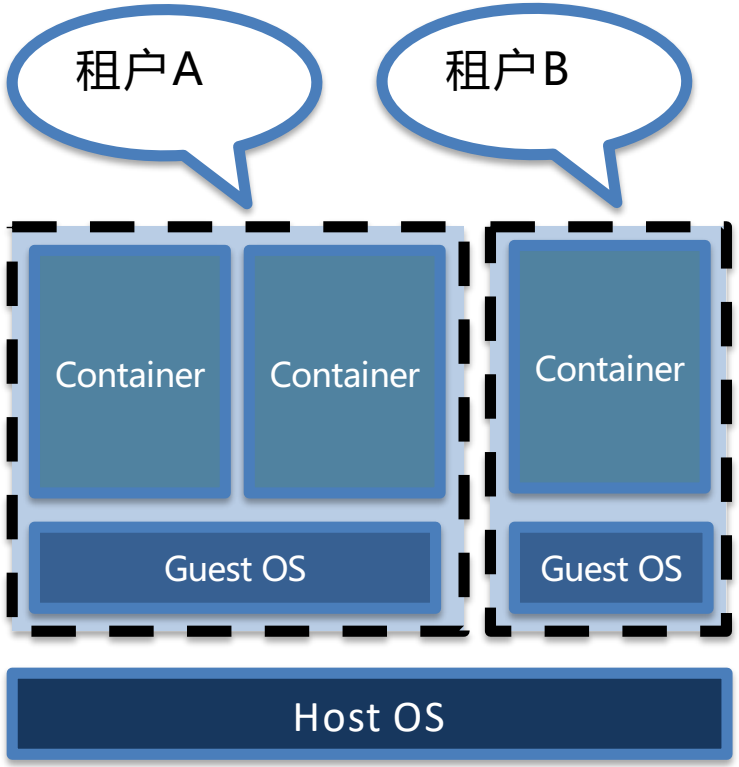
TKE@CVM

优点：

- 节点管理和扩缩容比较方便

缺点：

- VM带来额外资源开销
- VM给客户带来额外管理负担



CIS

便捷——无须购买虚拟机或物理机的docker产品

- 无须购买底层资源，通过简单的配置，就可以用docker image生成容器实例；
- 容器结束实例即会结束，无须手工释放，也可以配置重启策略使实例长久存在；

安全——虚拟化、网络双层隔离

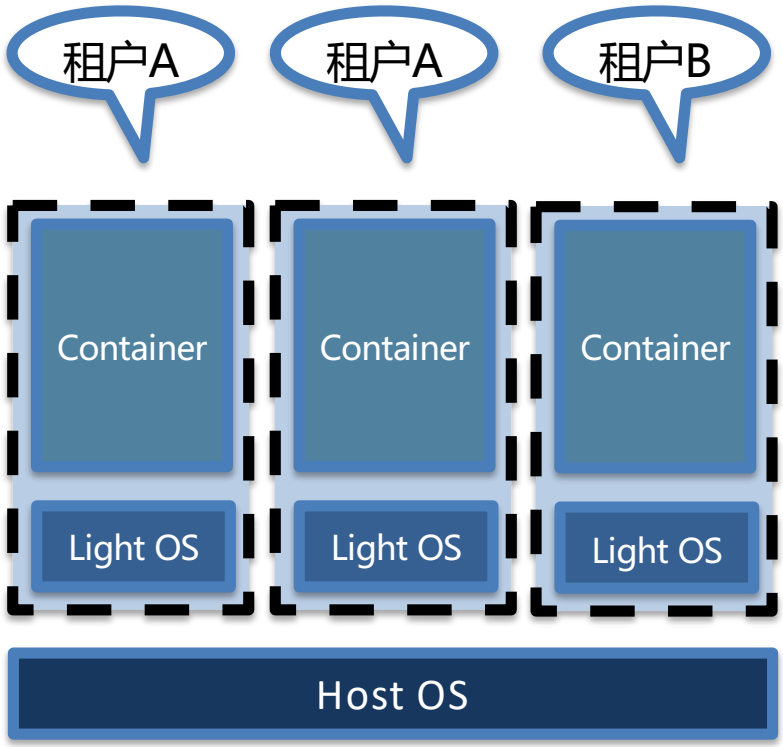
- Kata containers提供了docker级别的生产速度和vm级别的资源隔离
- 实例运行在用户的vpc网络中，支持配置安全组和ACL策略进行访问控制

便宜——根据消耗资源按秒计费

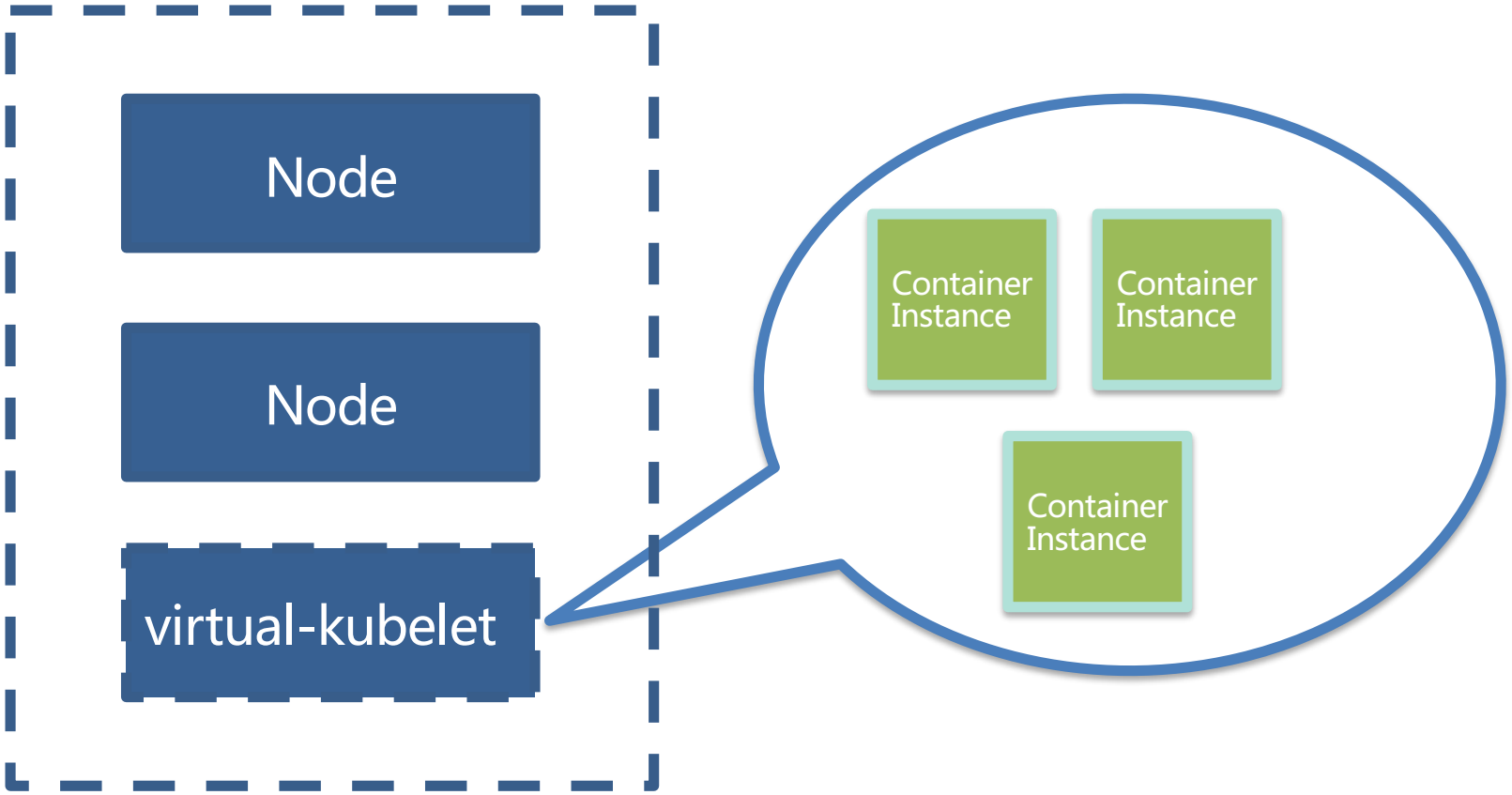
- 根据购买的cpu、内存量，按实例实际运行的时间按秒计费
- 从启动开始，实例无论出于什么原因一旦结束即停止计费

灵活——按所需自定义配置，支持购买超小资源

- 容器实例支持购买超小资源；
- 一个实例可以是一个容器，亦可以包含多个相关容器；



CIS with virtual-kubelet



user's kubernetes cluster

第二节

轻量级虚拟化方案对比

CIS轻量级虚拟化方案调研

- hyper.sh
- Clear Containers
- kata Containers

他们性能如何？
如何选择？

特性对比

差异点	CC	Hyper
实现方式	1、包含cc-runtime、cc-shim、cc-proxy、cc-agent、miniOS（kernel和rootfs） 2、只实现runtime，把原来runc拆分成cc-runtime和cc-agent，再加上VM带来的额外通信机制	1、包含hyperctl、hyperd、hyperkernel、hyperstart 2、和docker类似，实现一个前端命令hyperctl，和一个后台daemon hyperd
Hypervisor	1、只支持qemu，并且高度定制qemu 2、只支持qemu 2.7.0，后续会迁移至2.9.0	1、支持qemu、XEN、VirtualBox等 2、支持qemu 2.0以后的任何版本
硬件平台	1、支持Intel x86_64	1、支持Intel、ARM64、PPC等
其他依赖	1、go 1.8.0以上 2、docker 1.2.1以上、gcc-6.2.0、json-glib-1.2.2、glib-2.46.2、gmp-6.1.0、mpfr-3.1.4	1、go 1.7.0以上

基本参数对比

	Docker	CC	Hyper
容器启动时间	632ms	1249ms	3256ms
VM镜像大小	0M	62M	11M
容器内存开销	15M	136M	80M
ping host平均时延	0.05ms	0.14ms	0.2ms

CPU：8核Intel(R) Xeon(R) CPU X3440 @ 2.53GHz

系统：CentOS Linux release 7.2 (Final)

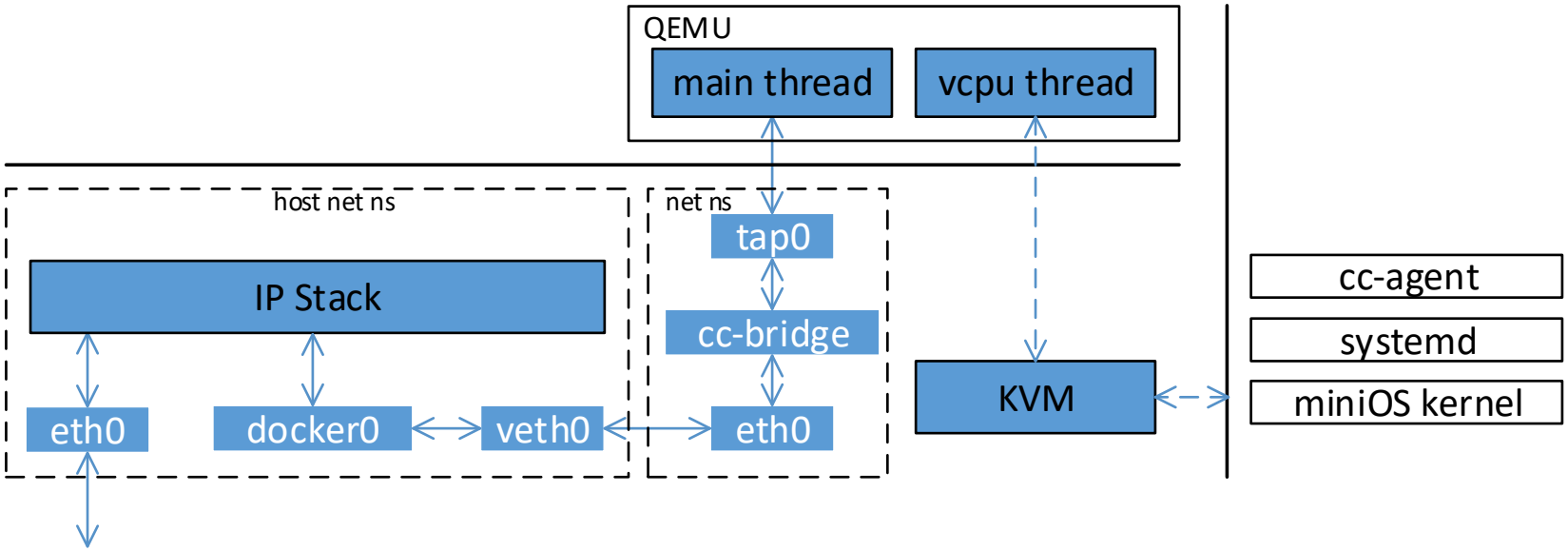
内核：3.10.0-693_12.tl2

Docker版本：1.12.6

CC版本：3.0.14

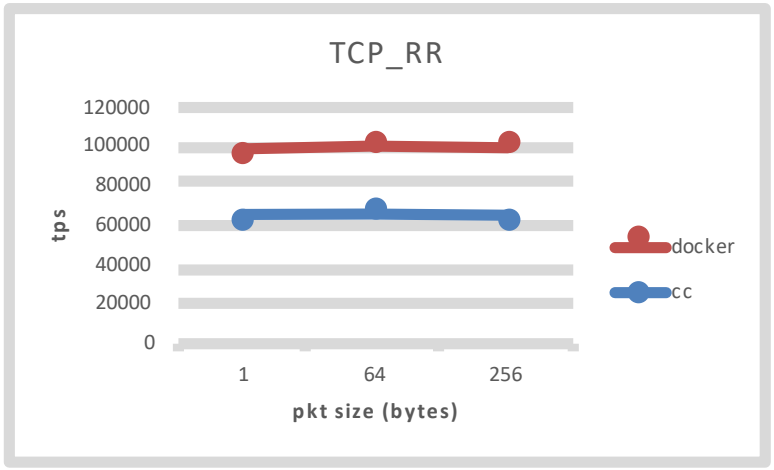
Hyper版本：1.0

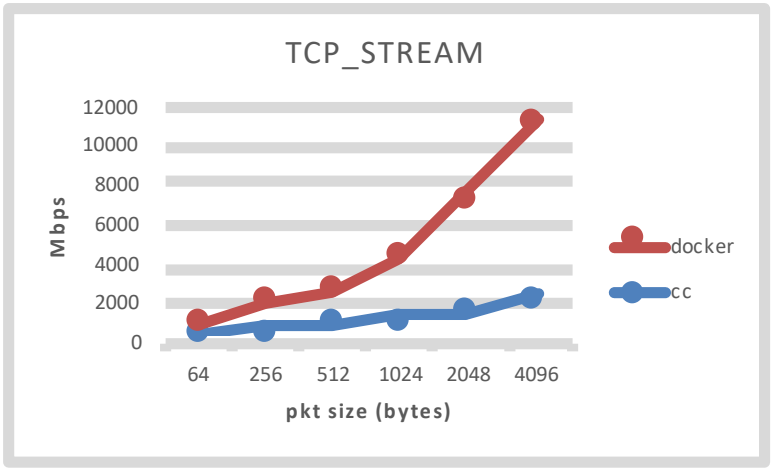
网络架构



- 虚线为控制流，实线为网络数据流
- CC为每个容器创建一个network namespace
- VM不支持veth pair，创建cc-bridge连接tap和veth pair

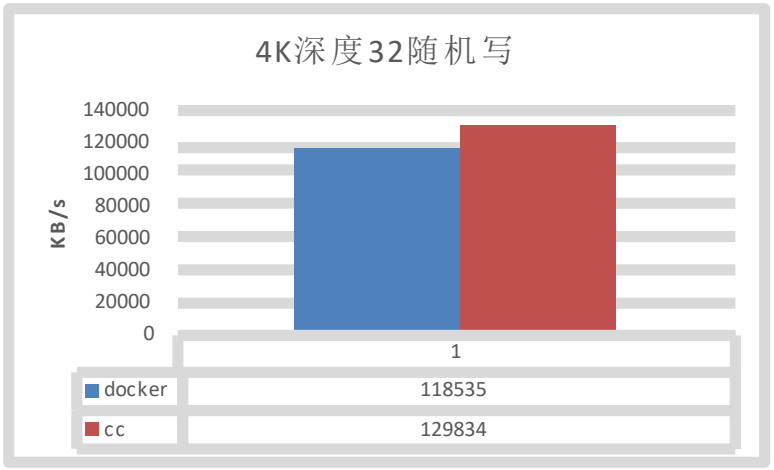
网络性能对比

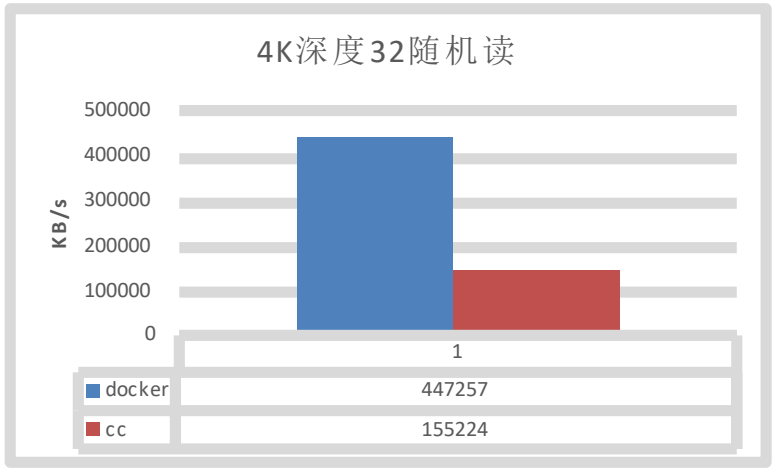


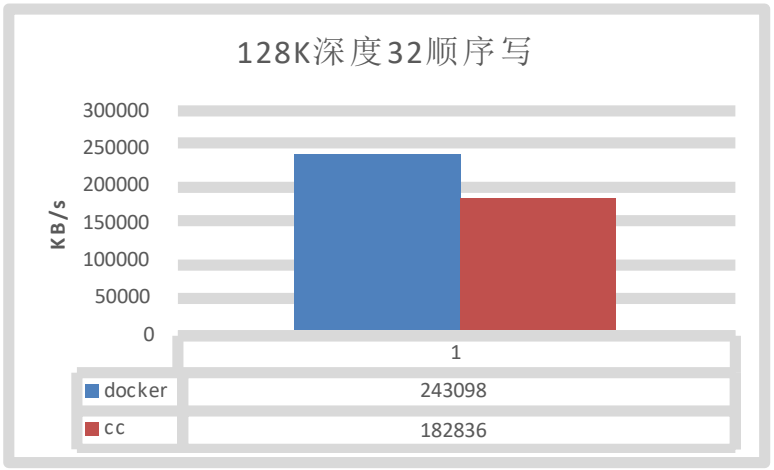


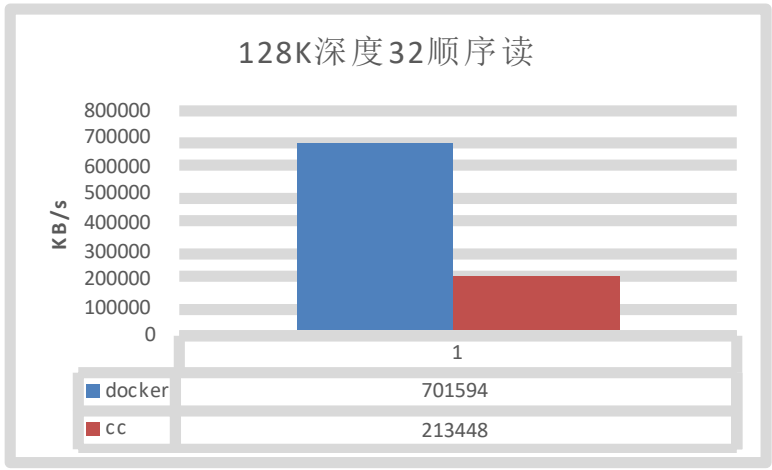
测试机型：浪潮 SA5212M5(6133*2/32G*12/SSD-480G*2 RAID/NVMe-4T*12/25GE*2)

磁盘IO性能对比







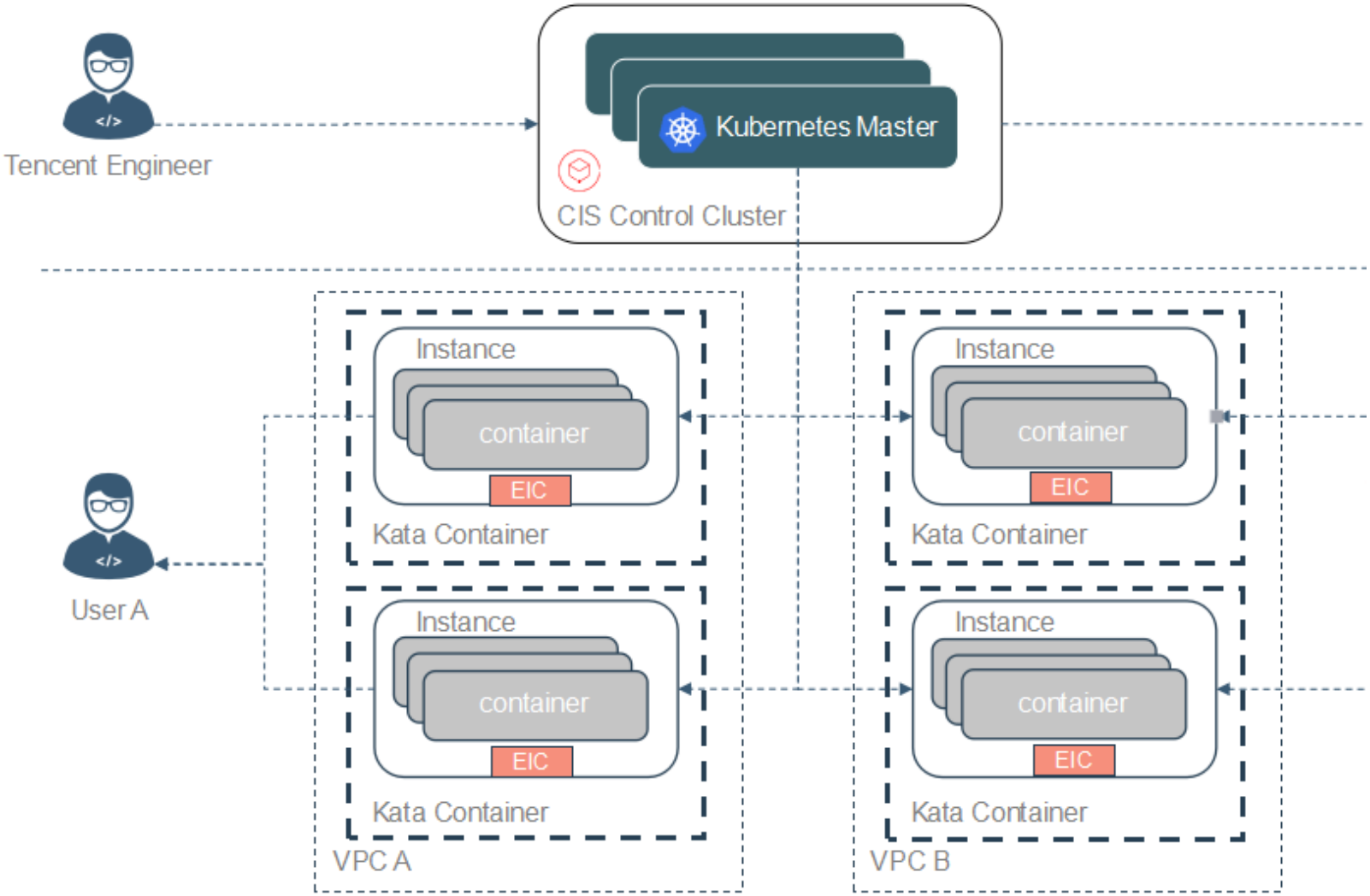


测试机型：浪潮 SA5212M5(6133*2/32G*12/SSD-480G*2 RAID/NVMe-4T*12/25GE*2)

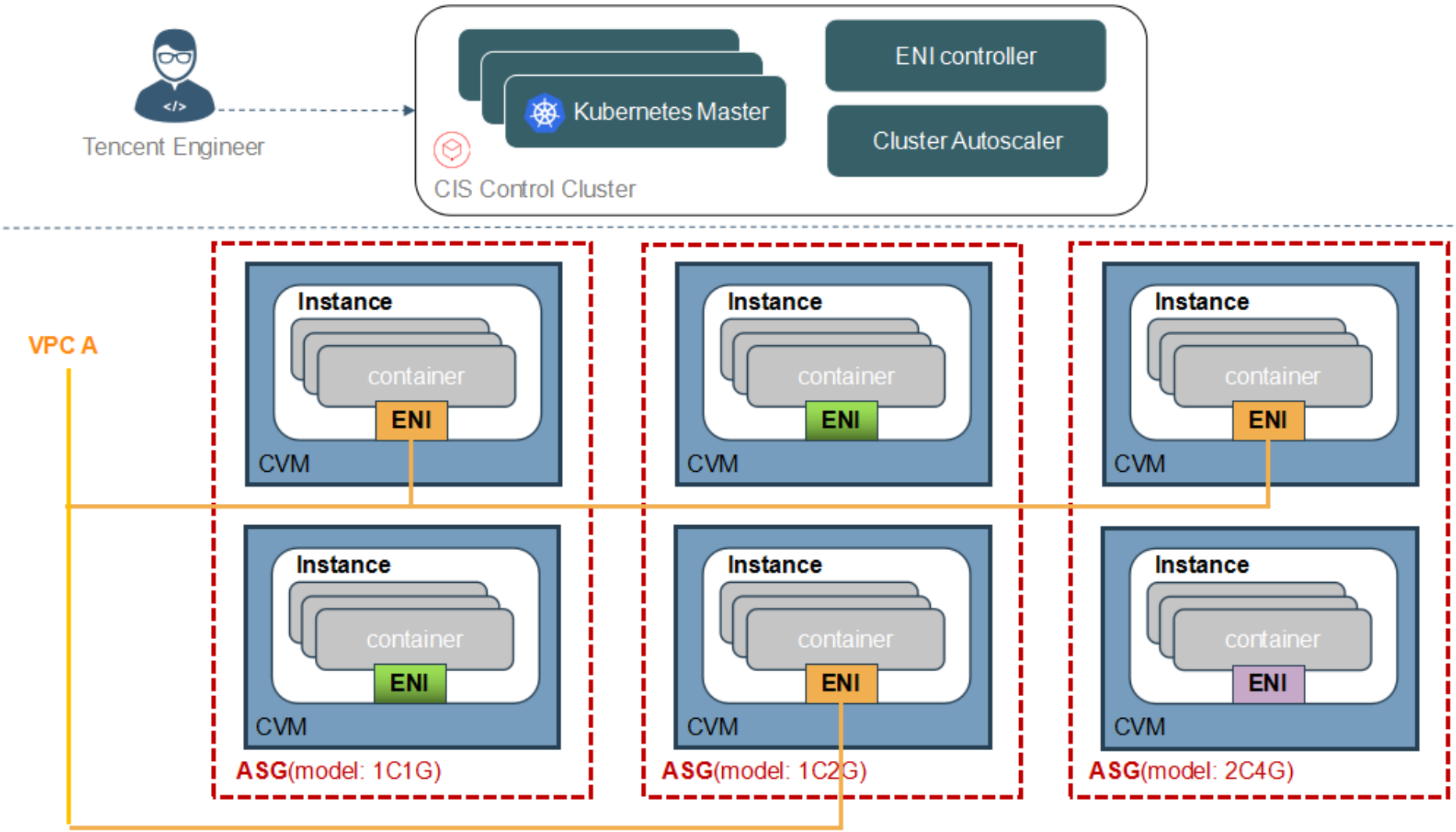
第三节

CIS v1 实现架构

CIS v1 架构概览


Tencent Engineer
Kubernetes Master
CIS Control Cluster
Instance
container
EIC
Kata Container
User A
VPC A
VPC B

CIS v1 架构概览


Tencent Engineer
Kubernetes Master
ENI controller
Cluster Autoscaler
CIS Control Cluster
VPC A
Instance
container
ENI
CVM
ASG(model: 1C1G)
ASG(model: 1C2G)
ASG(model: 2C4G)

CIS v1 架构概览和原则

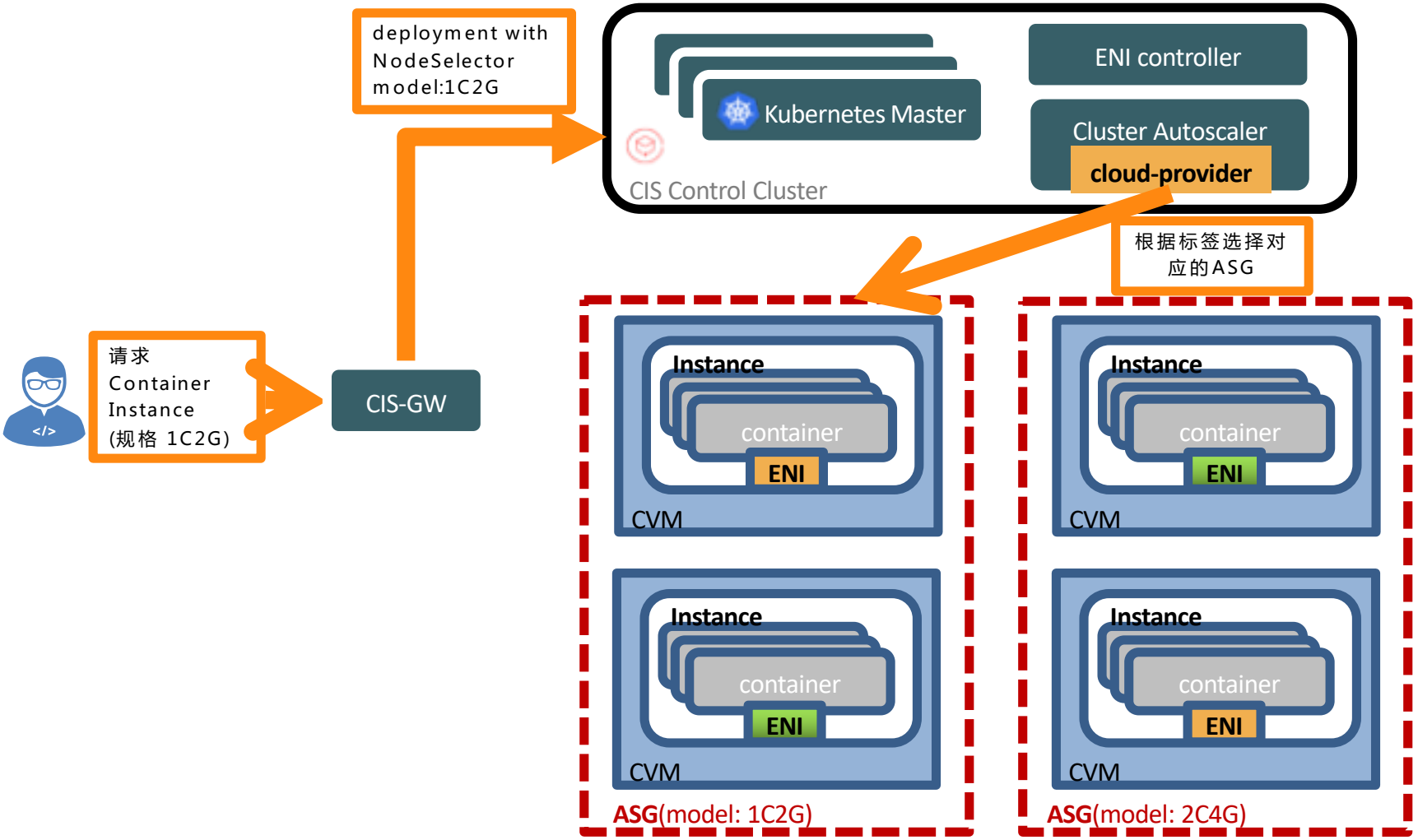
- ControlCluter本身是一个TKE托管集群
- 每个节点是特殊类型的CVM
- 节点的生命周期由vstation管理
- 每个节点比CIS实例规格稍大
- 用户购买一个CIS，后台在ControlCluster中创建一个Pod，通过NodeSelector调度到相应规格的CVM
- 通过CVM的自动伸缩组，控制节点的增加和删除

实现原则： 尽量复用腾讯云和k8s已有功能，提升CIS上线速度和稳定性。

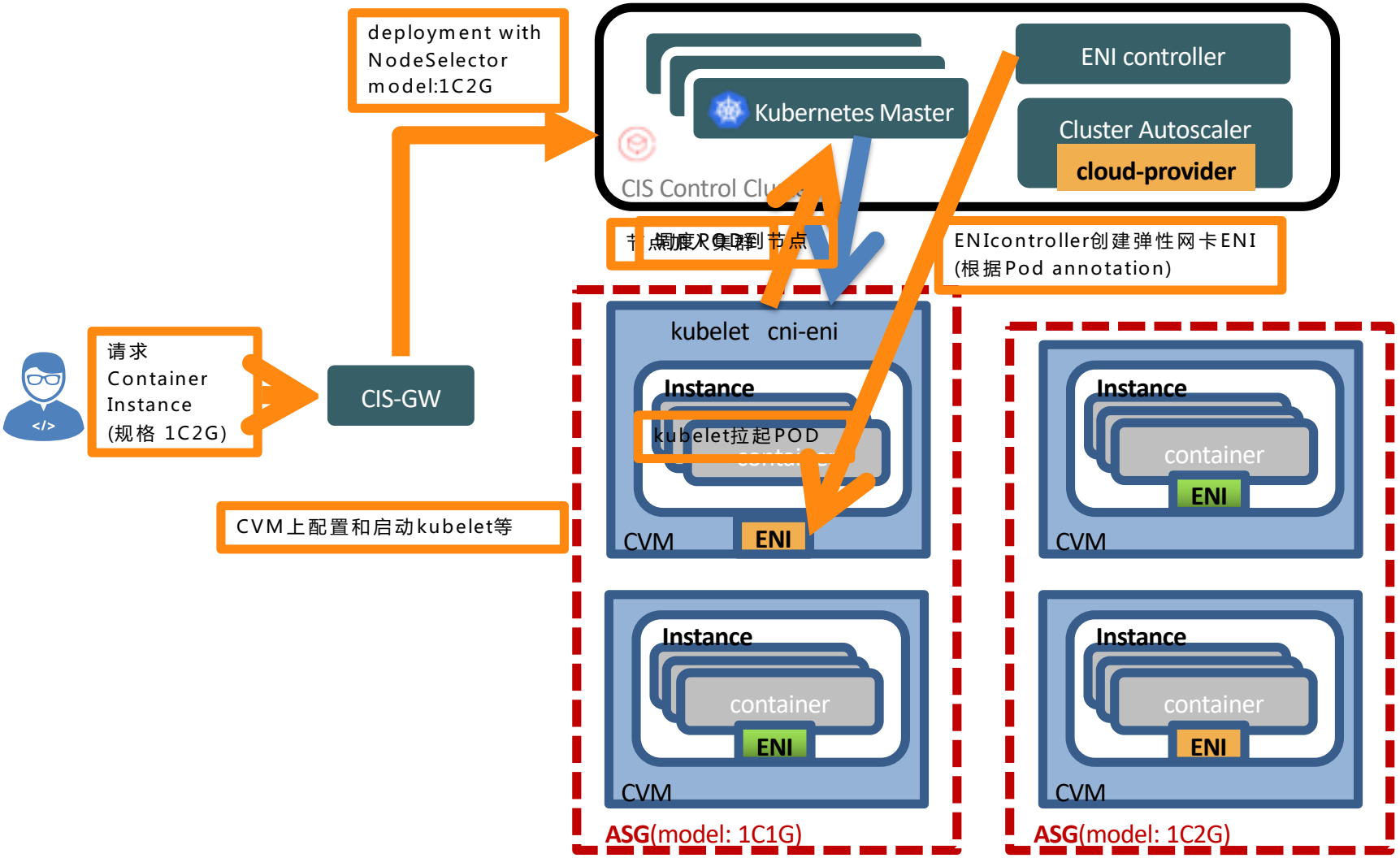
ASG自动打标签

- 给ASG关联一组标签，ASG创建出来的节点，会自动打上k8s 标签

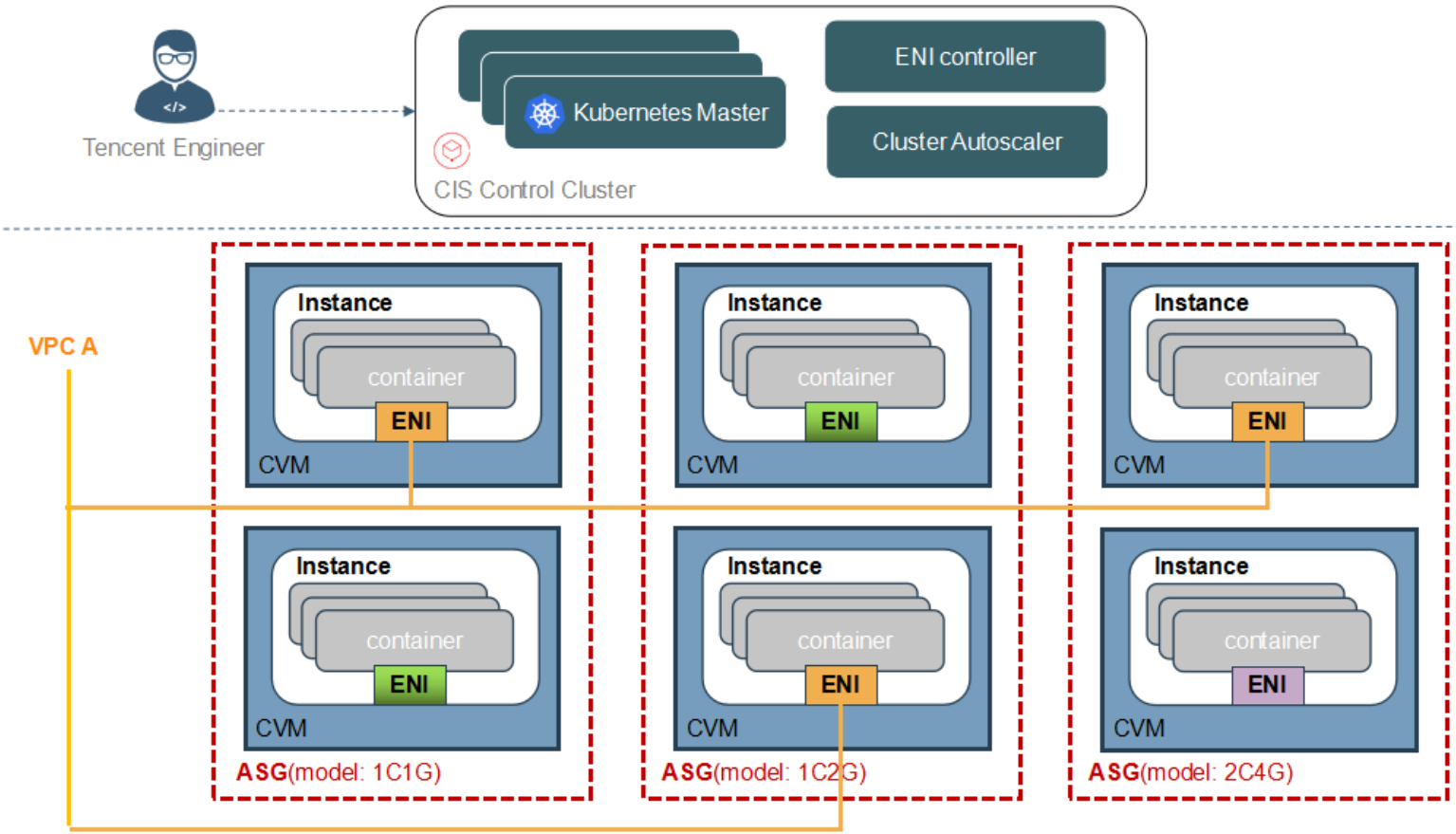
CIS实例请求 -> CVM


deployment with
NodeSelector
model:1C2G
Kubernetes Master
ENI controller
Cluster Autoscaler
cloud-provider
CIS Control Cluster
根据标签选择对
应的ASG
请求
Container
Instance
(规格 1C2G)
CIS-GW
Instance
container
ENI
CVM
ASG(model: 1C2G)
ASG(model: 2C4G)

CVM -> CIS实例


deployment with
NodeSelector
model:1C2G
Kubernetes Master
ENI controller
Cluster Autoscaler
cloud-provider
CIS Control Cluster
ENIcontroller创建弹性网卡ENI
(根据Pod annotation)
请求
Container
Instance
(规格 1C2G)
CIS-GW
CVM上配置和启动kubelet等
kubelet cni-eni
Instance
kubelet拉起POD
container
ENI
CVM
ASG(model: 1C1G)
ASG(model: 1C2G)

CIS网络架构



- 每个POD绑定一个弹性网卡，不绑定veth设备，不连bridge
- 同VPC的POD可以互通，不同VPC的POD不能通信
- POD跟节点之间不能通信
- nfs等volume插件需要修改，直接在容器中挂载

第四节

如何管理kata container

CIS v2 - 如何管理kata container

- 方案一

在底层把kata container当作一种新的CVM机型，完全交给vstation来管理。

Pro:

- 复用云主机系统的母机管理和虚拟机管理机制

Con:

- 需要自己实现相关Pod语义，以便支持virtual-kubelet

CIS v2 - 如何管理kata container

- 方案二

由kubernets直接管理母机和kata container

Pro:

- 完美支持Pod语义，方便接入virtual-kubelet

Con:

- 需要介入复杂的母机管理流程

LINUXCON containercon CHINA 中国 CLOUDOPEN

THINK OPEN

开放性思维

Thanks

zhiguohong@tencent.com