KUBERNETES COMMUNITY DAYS BEIJING 2025

机密计算在无服务器（Serverless）架构中的应用：机遇和挑战

中国移动云能力中心 容器服务团队

李磊　　刘艳松

Content

目录

Part 01

Serverless和机密计算

Serverless架构

- Serverless强调的是一种架构理念和服务模型，所谓的“无服务器”是对用户而言的，本质并不是不需要服务器。
- Serverless架构允许用户将主要精力集中在产品代码的开发上，而将基础设施建设、系统管理、应用构建以及部署等任务全部委托给第三方，也就是云供应商来负责管理。

给用户带来的核心价值：

- **弹性伸缩**：根据流量自动扩展或缩减资源
- **按需付费**：用多少资源就花多少钱，不用为闲置资源来买单
- **简化运维**：省去资源管理的烦恼，快速迭代和部署应用

安全方面的挑战：

- 业务数据的**机密性**
- 业务逻辑执行的**完整性**

依赖云供应商提供的无服务器运行环境的安全性，同时需要信任云供应商

机密计算（Confidential Computing）

- Confidential Computing Consortium 中的定义：

  Confidential Computing is the protection of data in use by performing computation in a hardware-based, attested Trusted Execution Environment.

- 主要成员：

Trusted Execution Environment

基本思想

- 在CUP和内存中单独划分一块隔离区域，进行敏感数据相关的计算。
- 通过基于硬件的加密保护，使得CPU的其它部分无法访问这块区域。
- TEE中的数据不能被TEE之外的任何代码读取和篡改。
- 只有经过适当授权的代码，才能够在TEE内操作数据。
- 对外仅提供经过授权的接口

关键特征

- Data confidentiality：未经授权的实体无法查看TEE中正在使用的数据
- Data integrity：未经授权的实体无法篡改TEE中正在使用的数据
- Code integrity：未经授权的实体无法篡改运行在TEE中的代码

未经授权的实体可能包括：

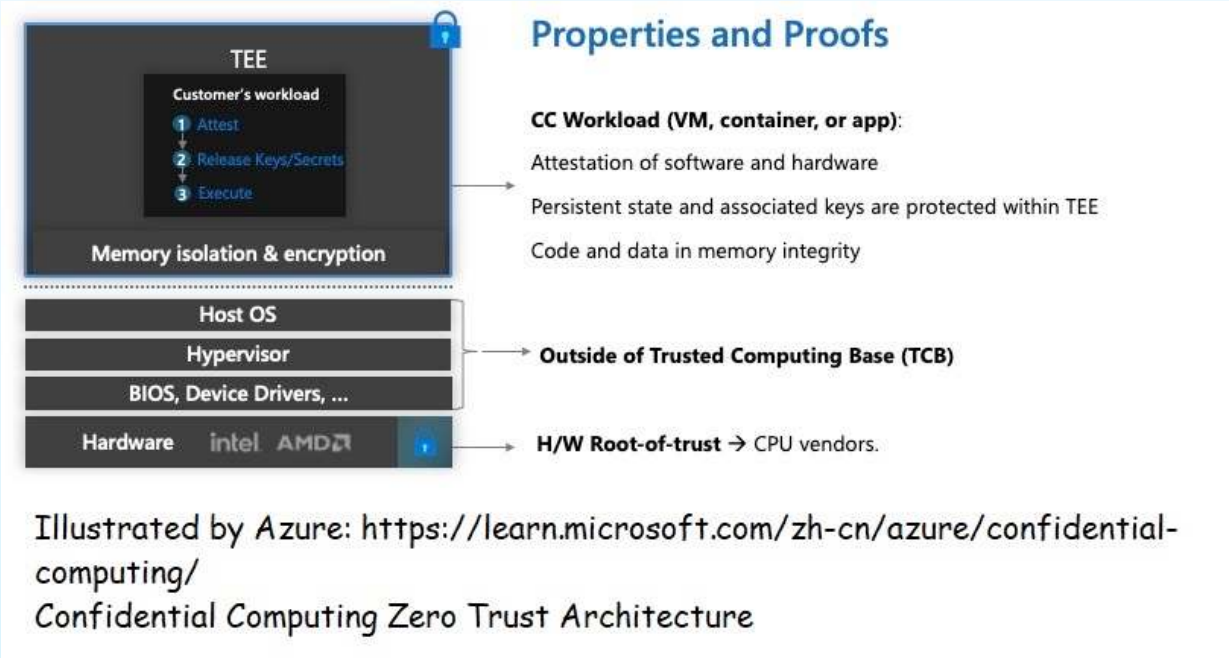
- 主机操作系统
- hypervisor
- 系统管理员
- 服务提供商
- 基础设施所有者
- 或者任何可以物理访问硬件的人员

可信计算基（TCB，Trusted Computing Base）

可信计算基 (TCB) 是指构成提供安全环境的系统的所有硬件、固件和软件组件

- 如果 TCB 内部有一个组件存在风险，那么可能会危及整个系统的安全
- 较低的TCB意味着其包含的组件更少， 也就意味着更高的安全性

采用机密计算技术，将云供应商及其提供的基础设施排除在TCB之外，提升了云上应用的安全性



Illustrated by Azure: https://learn.microsoft.com/zh-cn/azure/confidential-computing/
Confidential Computing Zero Trust Architecture

机密容器（Confidential Containers）

- CNCF 下的创新沙箱（Sandbox）项目，简称COCO
- **宗旨**：将TEE和云原生结合起来，提供云原生的机密计算。
- **设计目的**：
  - Remove cloud and infrastructure providers from the guest application Trusted Computing Base (TCB).

    从客户应用TCB中移除云和基础架构提供商。
  - Integrate natively with the Kubernetes control plane.

    与 Kubernetes 控制平面进行原生集成。
  - Provide an unmodified Kubernetes user and developer experience.

    Kubernetes的使用和开发体验不变。
  - Deploy unmodified workloads.

    部署的工作负载无需修改。

CONFIDENTIAL CONTAINERS

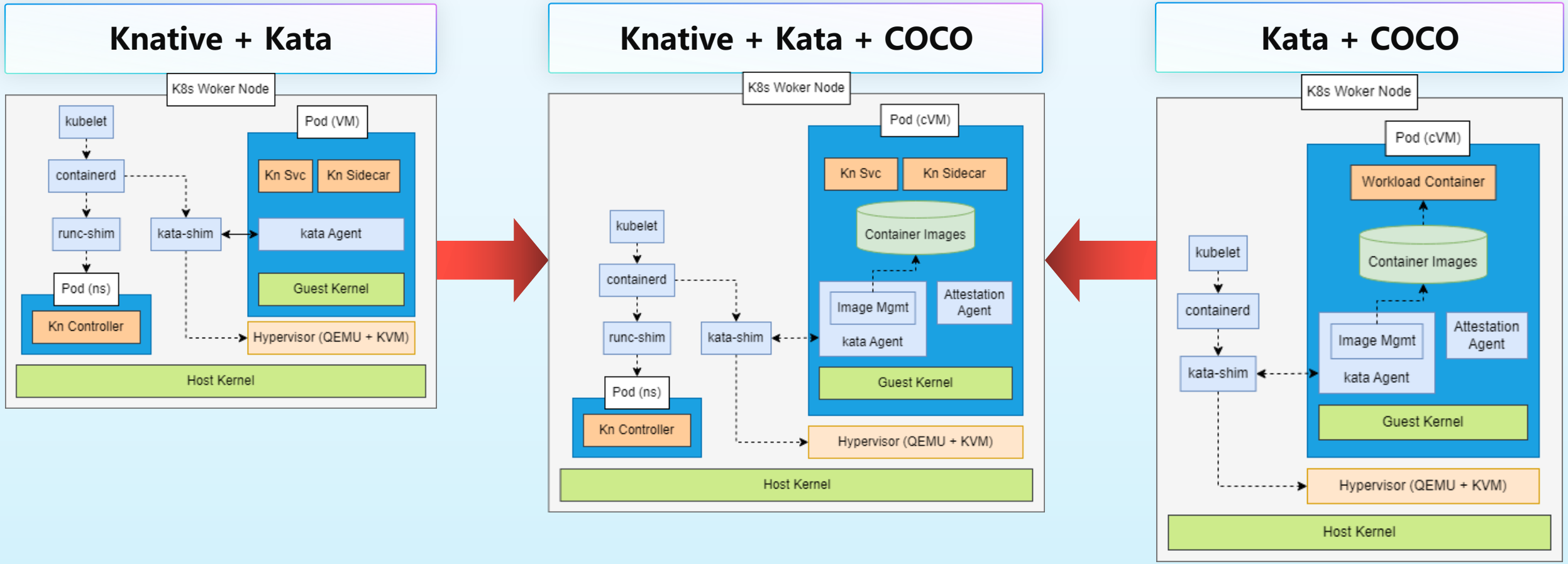
View Project Website

KCD
GROWING CLOUD NATIVE TOGETHER
BEIJING

Part 02

Knative + Confidential Containers

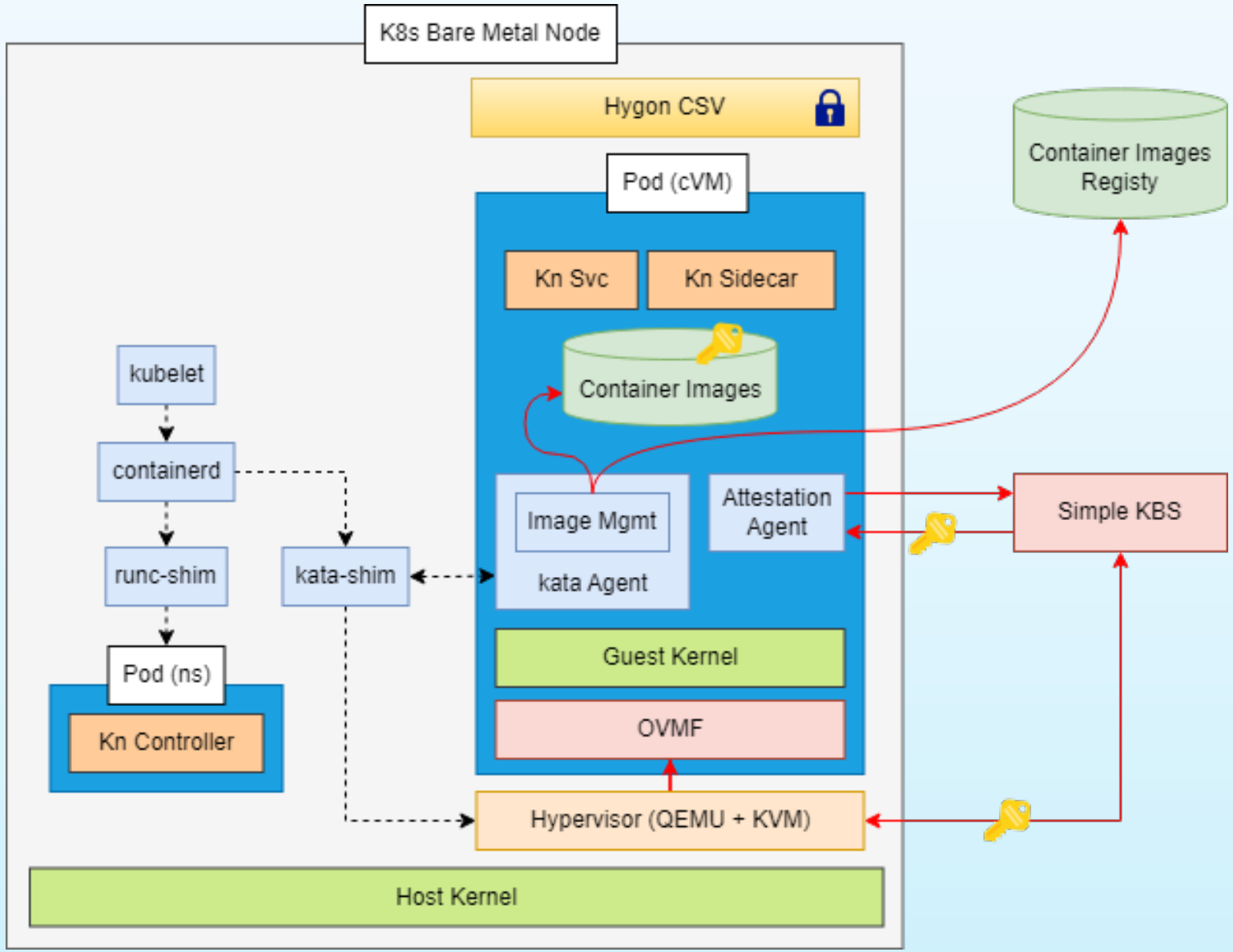
将机密容器应用到Serverless架构



环境搭建

- TEE选择：
  - ✓ **海光 China Secure Virtualization (CSV)** ← AMD Secure Encrypted Virtualization (SEV)
  - Intel Software Guard Extensions (SGX) / Trusted Domain Extensions (TDX)
  - 鲲鹏 TrustZone
- 基础设施：海光C86 7380 CPU的裸金属物理机
- 操作系统：BC-Linux v8.8 (Anolis 8.8) with kernel 5.10.134-csv
- Kubernetes Version: v1.28。单节点集群。
- Attestation：simple-kbs (Simple Key Broker Server)

顶层架构


K8s Bare Metal Node
Hygon CSV
Pod (cVM)
Kn Svc
Kn Sidecar
Container Images
Image Mgmt
kata Agent
Attestation Agent
Guest Kernel
OVMF
Hypervisor (QEMU + KVM)
Host Kernel
kubelet
containerd
runc-shim
kata-shim
Pod (ns)
Kn Controller
Container Images Registy
Simple KBS

Part 03

性能评估

方案设计和测试结果

基准（Baselines）

- runc container：在主机上运行container，用namespace隔离。
- Kata VM：在普通虚拟机中运行container。
- COCO cVM：在使能了CSV的机密虚拟机中运行container。

指标（Metrics）

- 冷启动（Cold Start）：指在创建 Knative Service 之后，首次发起请求并得到返回结果这一过程所花费的时间。
- 热启动（Warm Start）：指在 Knative Service 缩容至零实例（Scale to Zero）后，再次发起请求并得到返回结果这一过程所花费的时间。

方案设计和测试结果

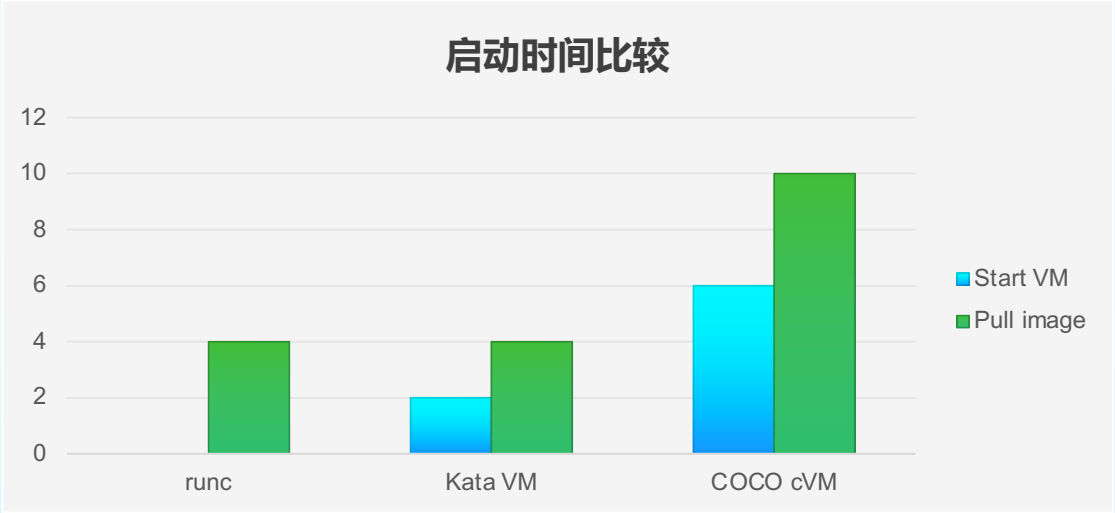
- https://github.com/knative/docs/tree/main/code-samples/serving/hello-world/helloworld-go
- https://github.com/knative/serving/releases/download/knative-v1.15.0
- 修改containerd的默认运行时：runc -> kata-qemu-csv

Baseline	Cold Start	Warm Start
runc container	5 s	1 s
Kata VM	7 s	2 s
COCO cVM	18 s	18 s

性能瓶颈分析

冷启动时间分析

- 在启用 CSV 的条件下，通过 OVMF 启动机密虚拟机比启动普通虚拟机多花费 **4 秒**。
- 在机密虚拟机内拉取容器镜像（含签名验证或镜像解密）的耗时是主机上拉取镜像的 **2.5 倍**。



热启动分析

- 热启动时间 ≈ 冷启动时间
- 原因：每次都需要创建新的机密虚拟机，然后重新拉取镜像。

Part 04
结论和挑战

结论分析

- ACM数字图书馆一篇研究论文结论基本一致
  - https://dl.acm.org/doi/10.1145/3642977.3652097

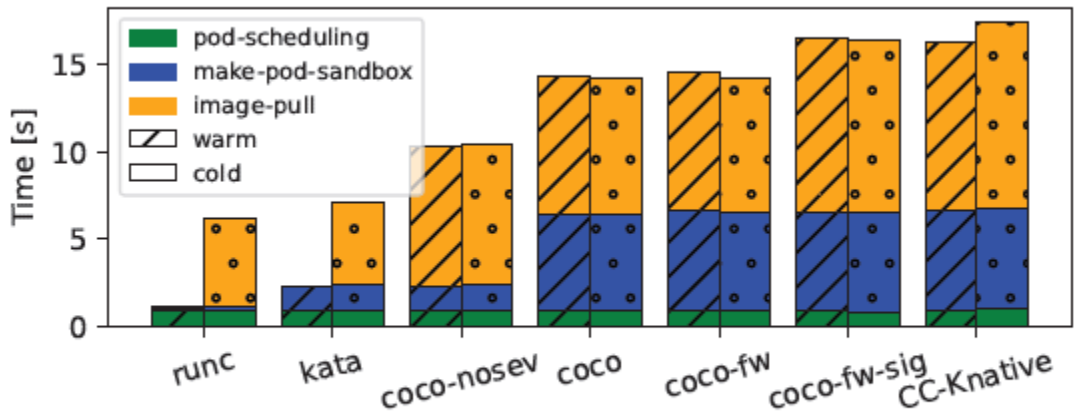
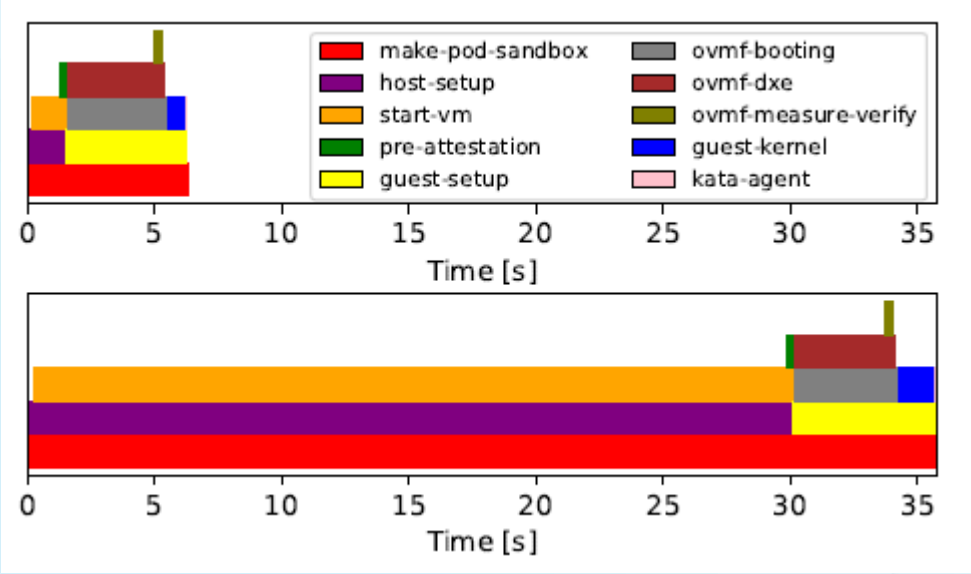


Figure 3: Start-up latency for a Knative service.

- pod-schedulling耗时很少，差异不大
- runc和kata 冷热启动差异巨大（Image-pull）
- CoCo冷热启动差异不大（都很慢）
- cVM启动耗时巨大

结论分析

内存分页带来时间花费
2GB vs 128GB



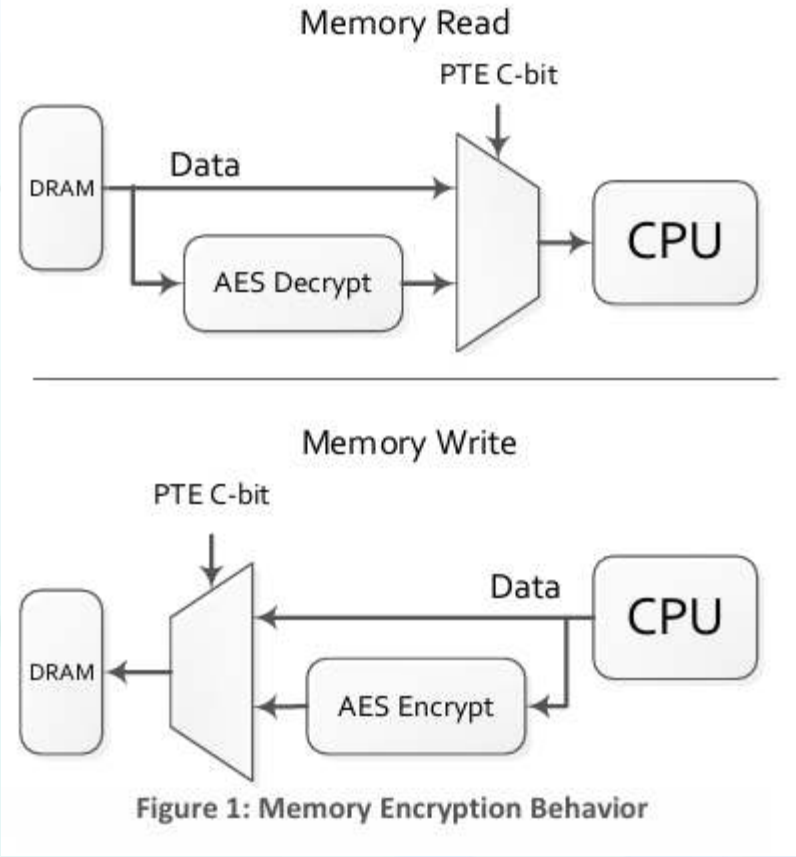
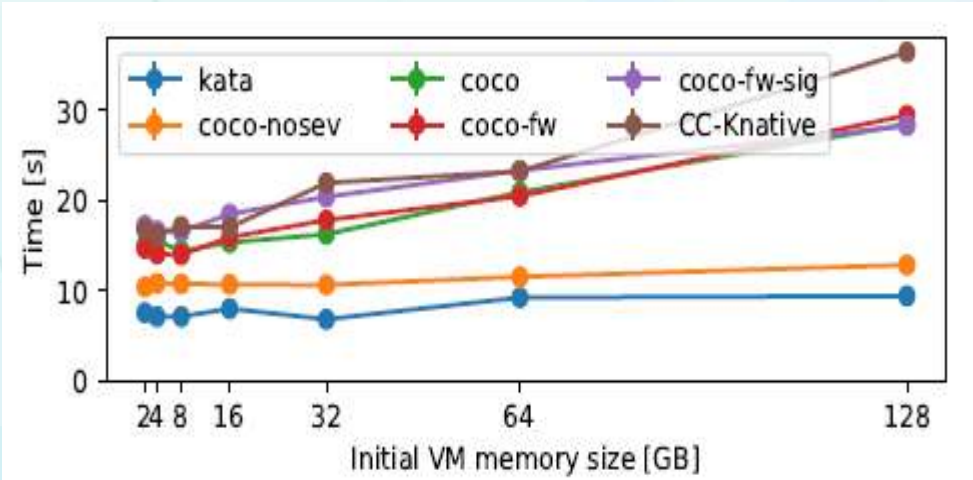


Figure 1: Memory Encryption Behavior

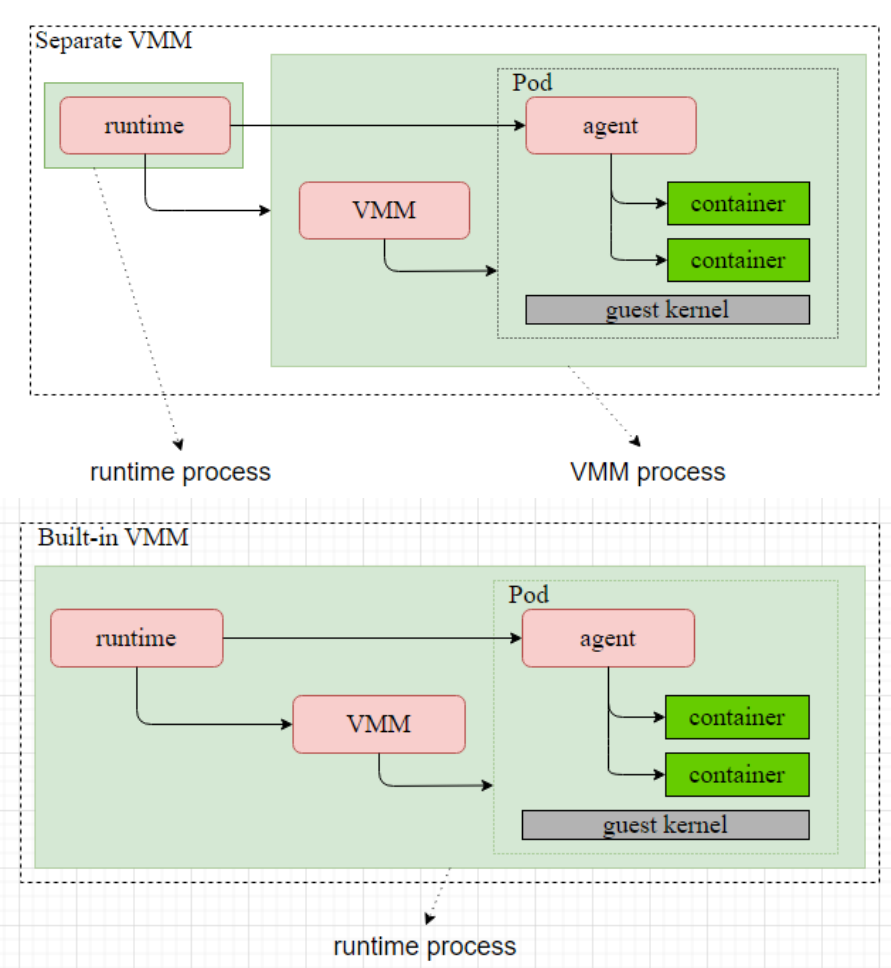


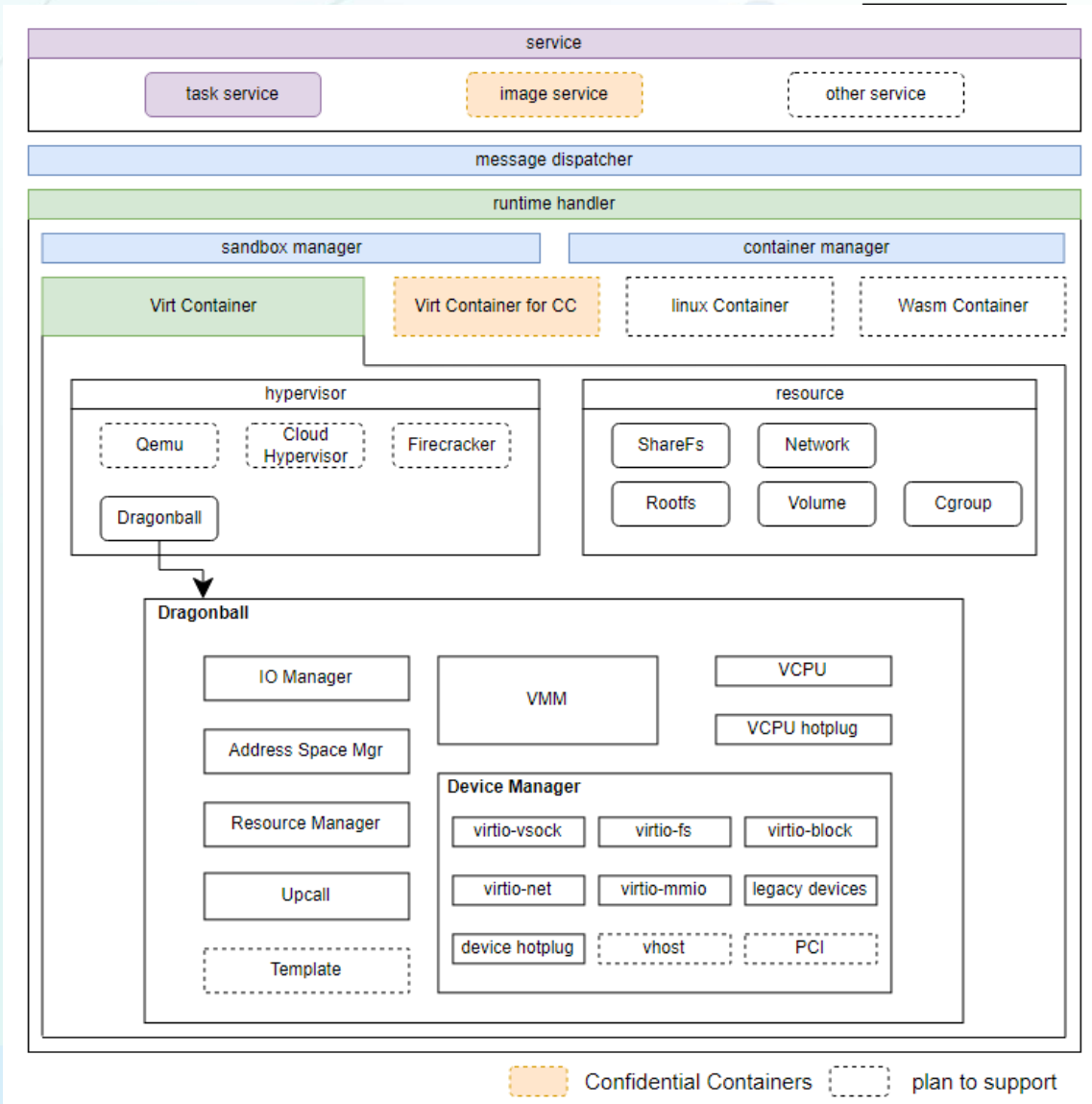
结论分析

- Pod的启动：沙盒（sandbox，pause）->拉取镜像（image pull）->容器启动（负载相关）
- kata运行时（runtime）：guest OS -> VM启动
  - OVMF(Open Virtual Machine Firmware)
- 内存分页（memory page）

沙箱启动优化

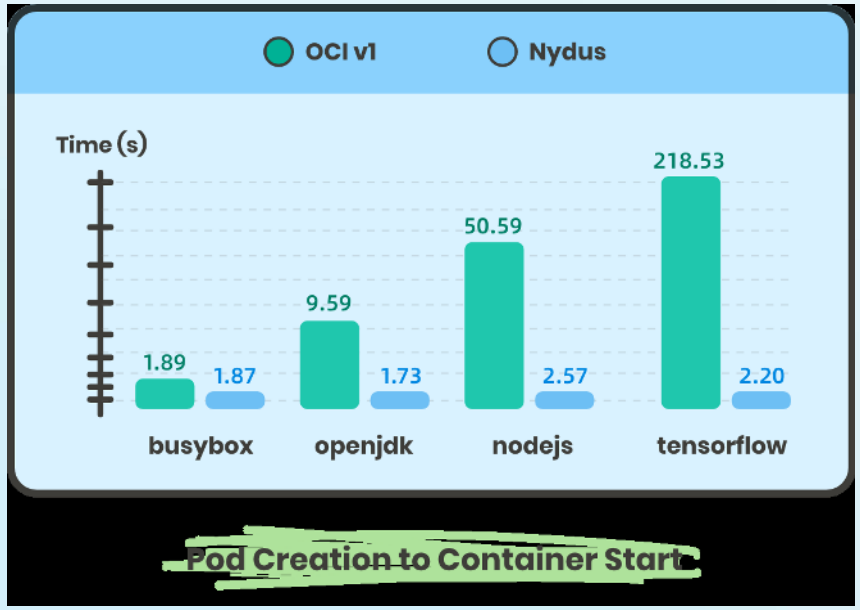
- VMM优化 - Dragonball

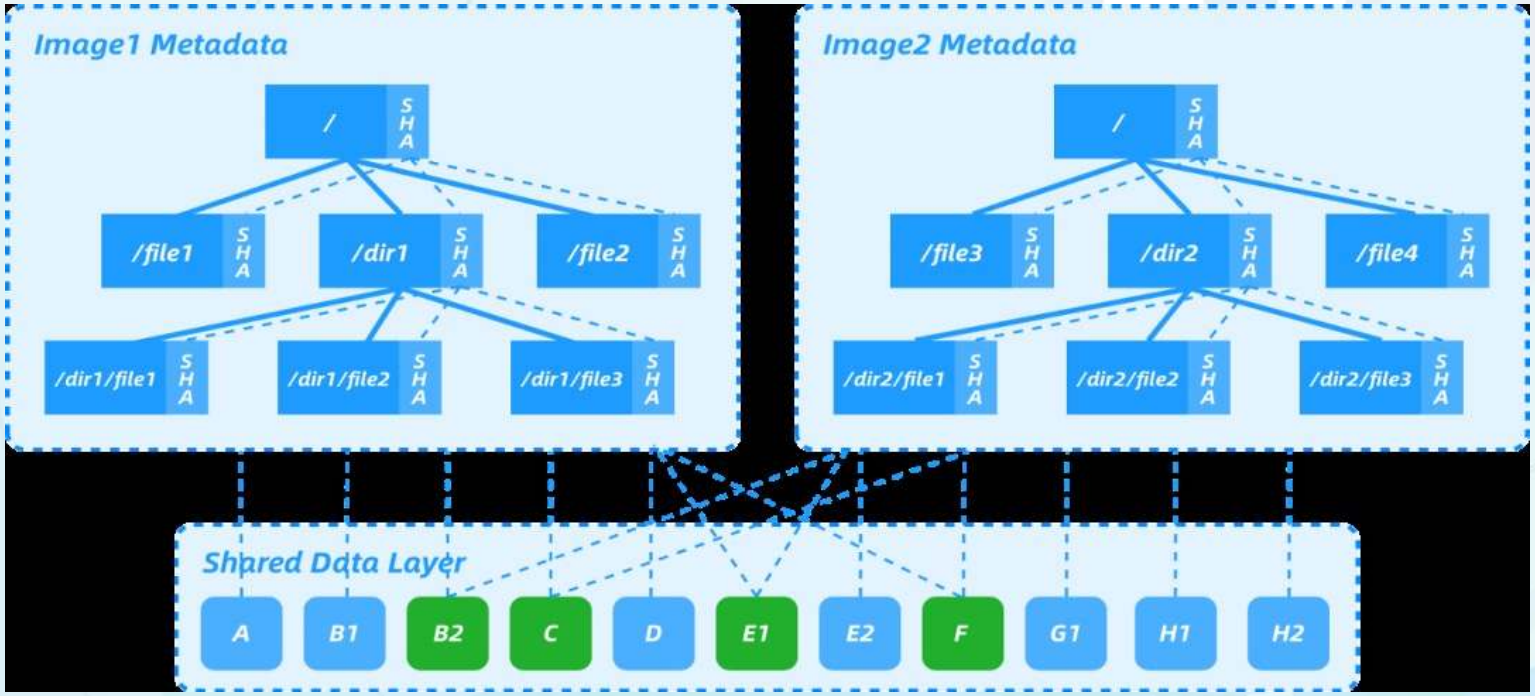




Nydus

- 容器镜像按需下载，用户不再需要下载完整镜像就能启动容器
- 块级别的镜像数据去重，最大限度为用户节省存储资源
- 镜像只有最终可用的数据，不需要保存和下载过期数据
- 端到端的数据一致性校验，为用户提供更好的数据保护
- 兼容 OCI 分发标准和 artifacts 标准，开箱即可用





移动云 容器服务 Serverless版

- https://ecloud.10086.cn/portal/product/ESK
- 容器服务 Serverless版，无服务器Kubernetes容器服务。
- 无需购买节点，无需对集群进行节点维护和容量规划，即可直接部署容器应用，并且只需要为应用配置的CPU和内存资源量进行按需付费。
- 集群中没有节点费用，Pod基于容器实例按量计费。

Thanks.