



CoreOS Contianer Linux介绍

CoreOS公司背景

- 基于云原生技术的科技创新公司
- 很多开源项目的主要代码贡献者
- 发布了超过100个开源项目
- 2018年2月被红帽以2.5亿美金收购



container linux
launched 2013
242 contributors
10,405 commits



etcd
launched 2013
382 contributors
8,099 commits



QUAY
launched 2013



flannel
launched 2014
73 contributors
605 commits



kubernetes
launched 2014
2,461 contributors
94,949 commits



rkt
launched 2014
202 contributors
5,248 commits



Tectonic
launched 2015



clair
launched 2015
46 contributors
570 commits



dex
launched 2015
73 contributors
1,157 commits

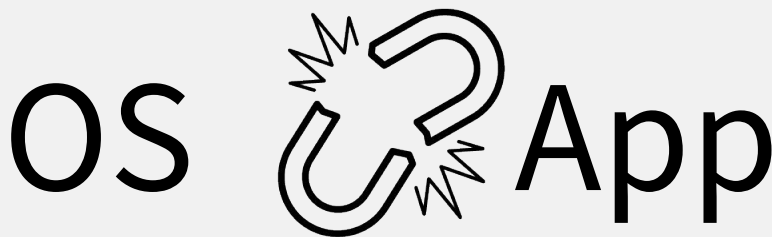
Since its inception in 2013 CoreOS has launched more than 100 open source projects. Our solar system is strong and growing thanks to all our contributors like you!

CoreOS

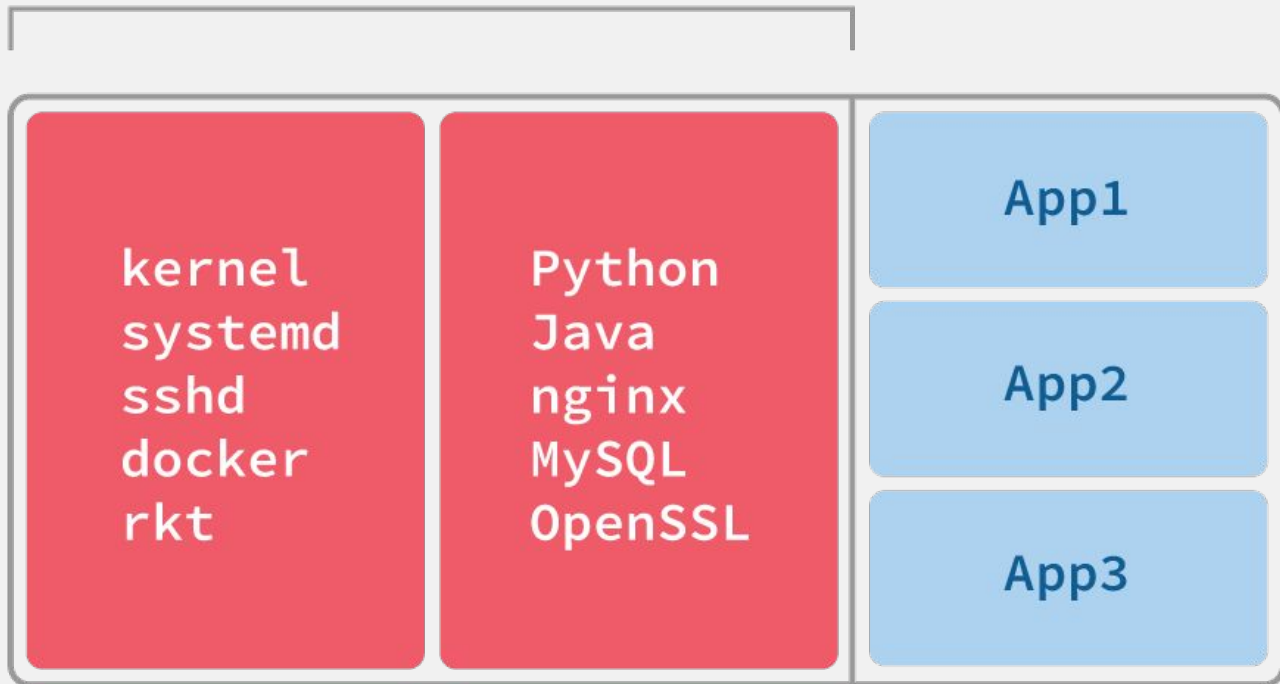


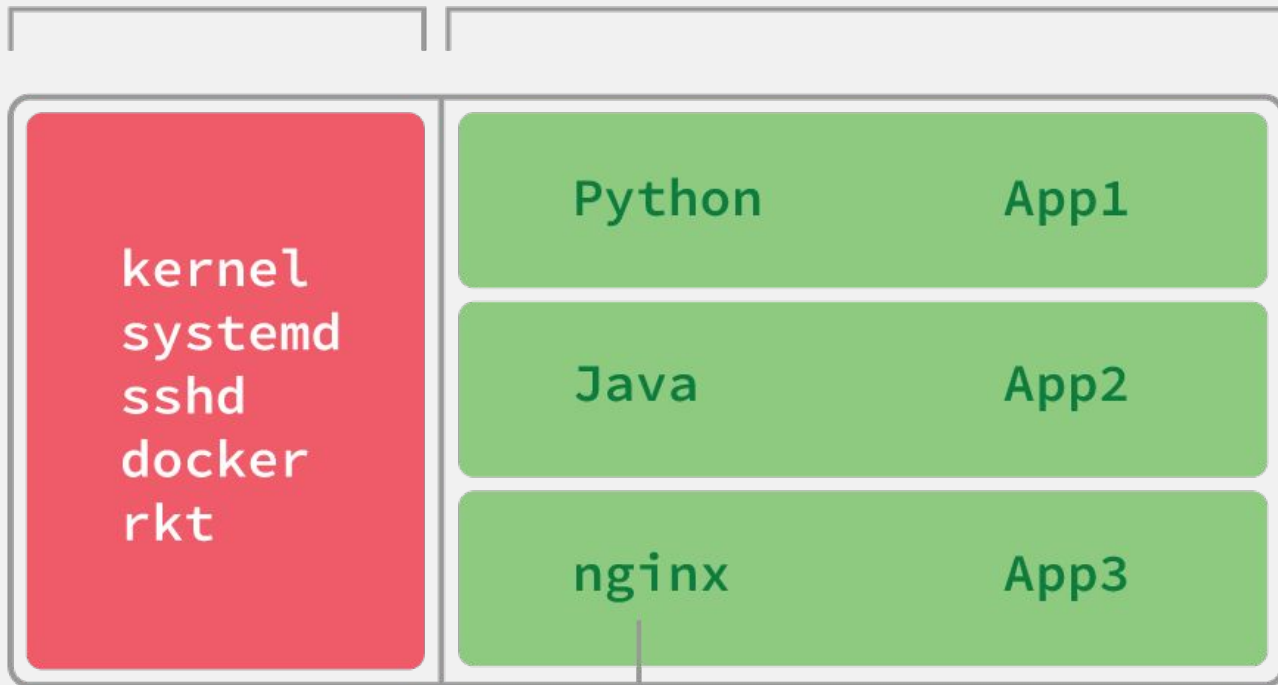
Linux设计理念——解耦操作系统和App

1. 底层操作系统应该尽量精简以加速引导速度和易于上手
2. 所有应用以容器的方式运行在OS之上
3. 服务都被init系统引导，例如：**systemd**
4. 操作系统应该有和**web**浏览器一样的自动升级策略
5. 操作系统需要多关注安全问题
6. 系统应该以集群为单位而不是独立的个体
7. 应该内建支持服务发现、容器网络和编排
8. 使用**etcd**作为分布式数据库



传统 Linux

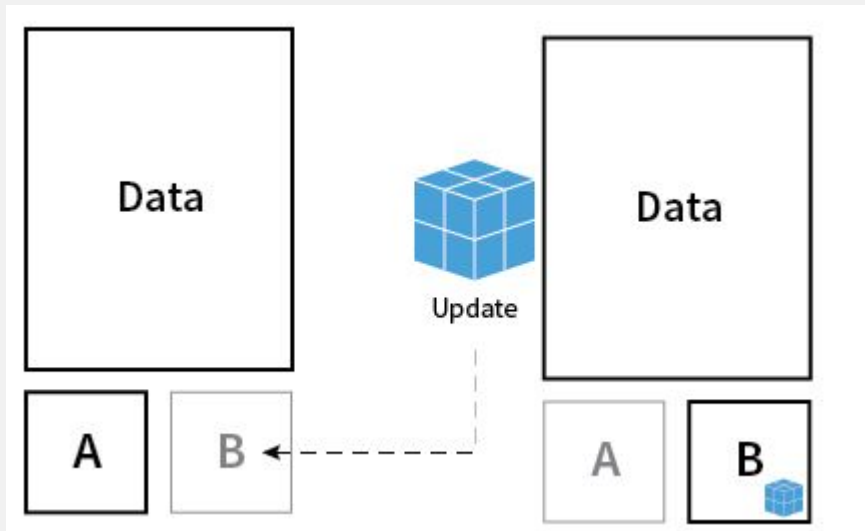




需要的软件

Container Linux 自动化更新机制

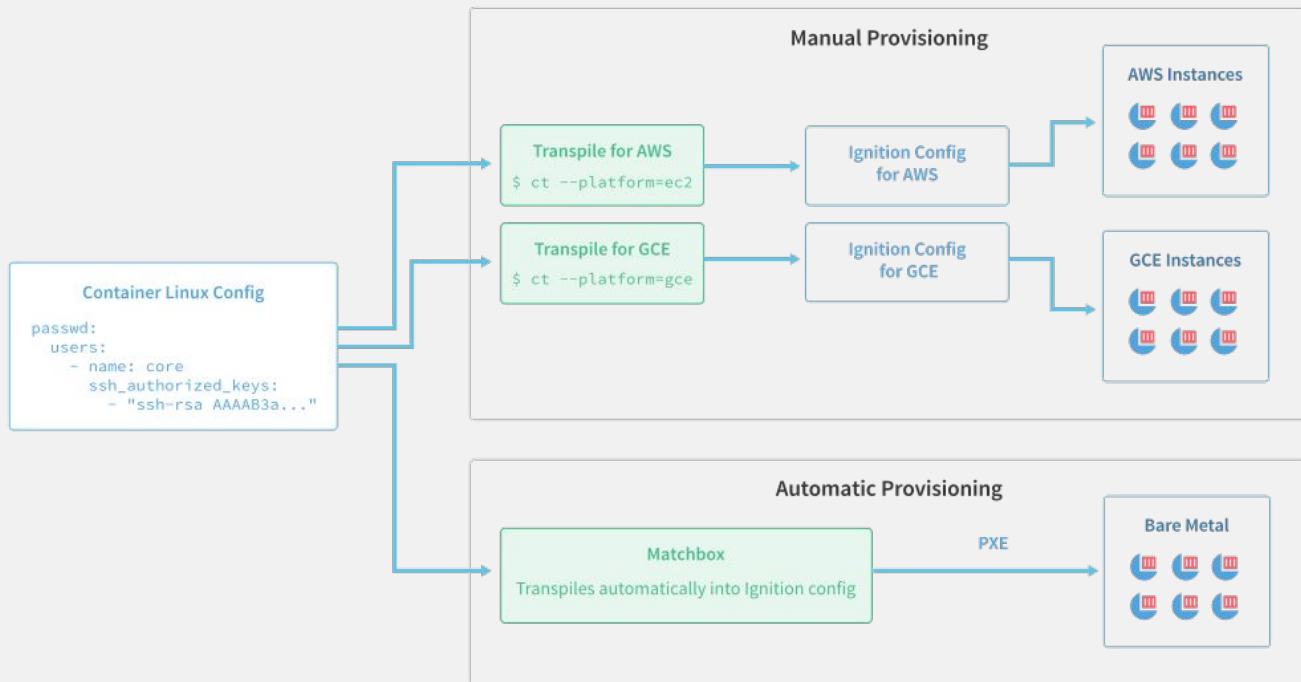
- CoreOS的Container Linux的更新机制基于Google的开源Omaha协议，该协议被用于各类谷歌产品
- 使用双分区来实现自动升级和处理在升级过程中出现的失误
- CoreOS处理升级的关键服务是update-engine 和locksmith service



CoreOS Ignition

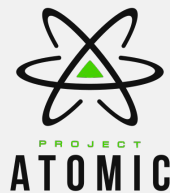
- 一键式部署工具，只运行一次
- 描述了部署后系统的状态
- 配置指定的机器
- 用户可以描述配置Container Linux的时候必须存在哪些文件系统，哪些文件需要生成、哪些用户必须存在等。以及使用systemd服务来实现进一步的配置。
- 支持的平台包括：
 - Bare Metal
 - PXE
 - Amazon EC2
 - Microsoft Azure
 - VMWare
 - Google Compute Engine
 - Packet
 - QEMU
 - DigitalOcean

Ignition配置CoreOS的机制



收购后CoreOS的Container Linux将与红帽的Atomic项目合并

- CoreOS和Fedora社区正在社区的整合工作
 - 具体计划将在年底发布
 - 会包含非Openshift的用例场景
 - 整合后更名为Red Hat CoreOS, 社区整合后更名为Fedora CoreOS



Red Hat CoreOS - 设计目标

保留两个产品的最佳特性

- Container Linux
 - 用户体验
 - 最小的构成
 - 容器化所有的工作负载
- Atomic Host
 - 红帽的生态系统

不会遗漏任何用户或者社区

Kubelet	CRI-O	docker
systemd		sshd
Kernel	SELinux	

Red Hat CoreOS - 设计目标

保留两个产品的最佳特性

- Container Linux
 - 用户体验
 - 最小的构成
 - 容器化所有的工作负载
- Atomic Host
 - 红帽的生态系统

不会遗漏任何用户或者社区

- 基于镜像部署
- 通过Ignition实现一键式部署
- 原来CoreOS的cloud-init部署功能将被删除
- 自动升级

Kubelet	CRI-O	docker
systemd		sshd
Kernel	SELinux	

Red Hat CoreOS - 设计目标

保留两个产品的最佳特性

- Container Linux
 - 用户体验
 - 最小的构成
 - 容器化所有的工作负载
- Atomic Host
 - 红帽的生态系统

不会遗漏任何用户或者社区

- 只保留最核心的操作系统功能
 - 在不牺牲核心功能的前提下尽量精简
- 集成运行容器所需的全部堆栈



Red Hat CoreOS - 设计目标

保留两个产品的最佳特性

- Container Linux
 - 用户体验
 - 最小的构成
 - 容器化所有的工作负载
 - Atomic Host
 - 红帽的生态系统
- 不会遗漏任何用户或者社区
- 主机和容器的生命周期独立
 - 更新节奏是分开的
 - 工作负载基于容器化的用户空间而不是主机



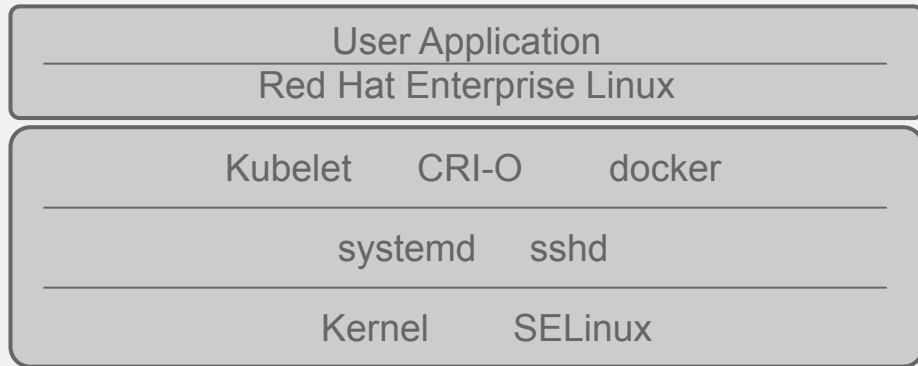
Red Hat CoreOS - 设计目标

保留两个产品的最佳特性

- Container Linux
 - 用户体验
 - 最小的构成
 - 容器化所有的工作负载
- Atomic Host
 - 红帽的生态系统

不会遗漏任何用户或者社区

- Red Hat为容器主机提供稳定性和集成。
 - RHEL ABI 放在容器内部
 - 为容器负载提供稳定性和兼容



Red Hat CoreOS - 设计目标

保留两个产品的最佳特性

- Container Linux
 - 用户体验
 - 最小的构成
 - 容器化所有的工作负载
 - Atomic Host
 - 红帽的生态系统
- 不会遗漏任何用户或者社区

- Red Hat CoreOS 会基于Openshift产品。
- CoreOS团队, Red Hat和Fedora会共同努力
- 上游社区的发行版本会涵盖 Container Linux今天所解决的所有用例



Red Hat CoreOS - 容器主机的进化

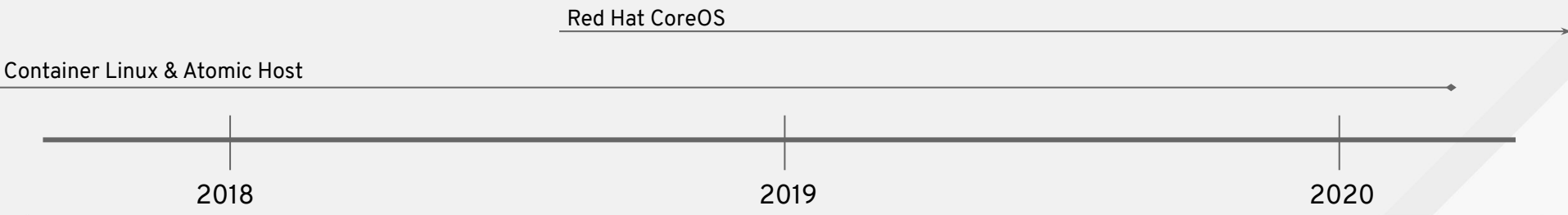
- **OpenShift的基础**
 - 一致的生命周期
 - 一致的发布频率
- 集成容器和操作系统堆栈
- 一键式部署
 - 支持云环境/虚拟化
 - 支持Bare metal
- 自动的、集群感知更新
 - 通过cluster operator实现全新的、智能的控制模式
 - 符合OCI标准

Kubelet	CRI-O	docker
systemd		sshd
Kernel	SELinux	

对现有部署的影响

- Red Hat CoreOS是Atomic Host和Container OS的接班人，是OpenShift的基础
- 在Openshift中依然可以使用RHEL
- Red Hat CoreOS将作为上游涵盖Atomic和Container Linux的所有用例

- 提供 充足的时间和产品重叠期以适应迁移
 - RHEL Atomic Host 会继续发布7.7版本
 - Container Linux 的更新和维护会持续到2019年以后。
- 迁移文档和工具已经在计划中



问题解答(基于CoreOS Container Linux)

补丁升级：因为CoreOS采用AB内核，那操作系统内核升级、操作系统补丁升级，能不能做到不重启操作系统，或者对业务无感知-包括不需要应用改造，不需要容器漂移什么的；

- **自动重启或者手工重启：**首先 Container Linux采用的是AB分区，默认启动在A分区。当检测到有升级时，会将升级版部署到B分区，然后根据相应的升级策略（etcd-lock、reboot和off）进行自动重启或者手工重启
- **无需应用改造：**在Container Linux中，所有的应用都是以容器方式运行的，因此Container Linux本身的升级不会对应用有任何的兼容的影响
- **容器会漂移：**Container Linux的设计理念就是以集群替代单机，因此重启时应用会在其他节点上重新部署，结合负载均衡可以将对应用的影响降到最低

问题解答(基于CoreOS Container Linux)

统一管理:有没有统一管理工具,实现对所有CoreOS的统一管理,查看所有OS的部署情况、配置情况、资源使用情况、支持统一下发配置等;

- 有:Container Linux的设计初衷就是自动化,轻松部署,安全,可靠,规模化。为了实现这个目标提供了CoreOS Ignition部署工具来实现对任意环境下Container Linux的一键式部署,包括用户、口令、服务以及集群配置等。
- Container Linux和Red Hat CoreOS都属于Immutable Linux, 在部署结束后无需做过多的管理。
- Prometheus是CoreOS名下的开源项目,也是红帽Openshift平台的重要组件,可用于实现对服务器、容器和各种应用的监控。

问题解答(基于CoreOS Container Linux)

资源节省:

和完整的发行版的Linux相比, 有什么 资源上的优势: 磁盘空间, CPU, 内存, 启动速度, 单机部署的容器的稳定性, 单机支持的容器数量增长多少等等。。。。。

- 前面我们已经提到Container Linux是精简的Linux, 只包含基本操作系统以及运行和配置容器和容器集群所需的最基本的组件, 没有多余的服务和进程, 所以对资源的占用和开销是非常少的。
- 精简Container Linux的首要目的是为了便于维护 and 安全性, 红帽的CoreOS会默认针对容器运行环境进行优化, 一定有性能提升。由于产品暂未发布, 性能数据不详 (目前服务器的处理能力都非常强, 而且原生的Linux和上面的应用对资源的占用本身就不高, 精简掉的应用对服务器的资源占用其实可以忽略不计, 我们不必过分纠结性能提升问题)。
- 同上, 此外单机上支持的容器数量和容器内部运行的 应用对资源的占用有关, 无法一概而定
- 未来的Red Hat CoreOS是基于RHEL开发的, 会继承RHEL所有的特性, 包括认证、安全、以及稳定性

问题解答(基于CoreOS Container Linux)

部署场景:

是否支持裸机部署?

- 支持裸机部署
- 此外还支持Amazon EC2、微软Azure、Google Compute Engine、Digital Ocean、VMware、KVM等云计算和虚拟化环境

问题解答(基于CoreOS Container Linux)

是否支持在其上部署 KVM？安装部署的方法是否便捷？

- 目前的社区版本不支持：他的主要目的是运行容器
- 红帽的Openshift在未来的版本会支持容器原生虚拟化，通过kubevirt(kubevirt.io)来实现。鉴于Openshift底层是Red Hat CoreOS，因此合并后的Red Hat CoreOS版本应该是支持在其上面部署KVM的，具体细节需要等产品发布之后。



THANK YOU



plus.google.com/+RedHat



facebook.com/redhatinc



linkedin.com/company/red-hat



twitter.com/RedHatNews



youtube.com/user/RedHatVideos