**密级：D**



**大数据组件容器化趋势分析**

# 数据统计



上述统计从存储组件、计算组件、调度组件、周边组件选择了21个常用组件，从以下三个方面统计了他们的容器化程度。

|  |  |
| --- | --- |
| 官方镜像 | 指组件的官方安装方式中有Docker部署方案，并且版本持续更新； |
| 第三方镜像 | 指hub.docker.com上的第三方镜像。 |
| Helm Chart | Helm Chart实际上是Kubernetes一种应用打包方式，类似于Linux下的yum，apt。 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | 有官方镜像 | 有第三方镜像 | 有Helm安装包 |
| 比例（21个组件） | 9/21 | 21/21 | 9/21 |

**官方镜像：**

有9个组件提供持续更新的官方镜像，这类组件有以下的特点：

* 轻量、简单（如ZK，Redis）
* 大规模应用时Docker已经成为潮流（如Flink、Sentry）
* 非密切依赖Hadoop（如Mongodb、ES）

**第三方镜像：**

以现如今的Docker的发展态势，几乎所有组件都能在docker hub找到有人做好的镜像。

这一类镜像存在以下特点：

* 没有详细使用说明文档
* 绝大多数镜像作者是以Demo、QuickStart为目的制作的，组件以最简单的方式部署，不具备HA、扩容等特性。

**Helm Chart：**

[Helm Chart](https://github.com/helm/charts)是一个打造Kubernetes生态的项目，目的是将大量生产中的一些基础组件从裸机运行迁移到Kubernetes下容器运行。Helm Chart提供的组件，容器化方案一般情况下优于上面两种。但是使用过程可能并不Easy（用户需要了解Kubernetes的一些基本知识）。

Helm Chart有近百个组件的安装包，但是和大数据相关的组件并不多（只有9个）。

**个人认为**：Helm+Kubernetes是一个容器大数据平台的框架，我们可以在这个基础上丰富组件、简化使用难度、并且引进高级特性（如伸缩性），形成自己的大数据平台。

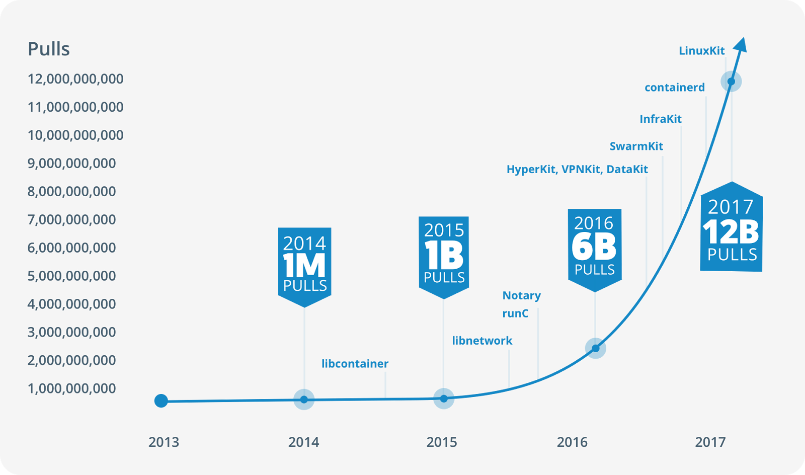
# 哪些组件不适合容器化？

**大数据生态中，并非所有组件都适合运行在容器中。**

尤其以Hadoop生态为代表的大集群“重型武器”，以及存储、数据库相关的基础服务，包括：HDFS（底层文件系统）、HBase（NoSql数据库）、Yarn（资源管理框架）。

包括以下几个原因：

* 从使用场景上看，这类组件是**基础服务，**一般独立部署在一个单独的集群中，规模很大，部署完成不会随意变动**。此类组件即使容器化部署，也无法发挥Docker快速部署、快速升级的优势（几年部署一次）。**
* 从功能上看，Hadoop的组件全是分布式，**天生支持HA、动态扩展**。无需Kubernetes等Docker编排工具来实现上述功能。容器化是重复造轮子！！
* 从架构上看，Hadoop生态中的核心组件大都诞生于2008年之前，并在2013年之前高速发展。此时，容器化的浪潮还没有开始（docker大规模普及应该是15年后），**一些组件在设计架构上和云原生应用的设计原则相违背**，勉强容器化也发挥不了容器化带来的优势。



# 哪些部分需要容器化？

**代码频繁变动，需要版本迭代的用户代码需要容器化。**

典型例子包括：Web应用、Spark的业务代码。

原因包括以下几点：

* 在开发过程中业务代码会反复变动，反复部署，容器化能够最大程度减小部署时间。
* 利用Kubernetes的编排能力，能够实现业务弹性伸缩、高可用。
* 业务代码是自己设计的，在设计过程中能够考虑设计是否符合云化、容器化要求，容器化效果会更好。

合理的架构应该如下图：

基础服务集群依然是非容器化的（或者说不一定非要要求完全容器化），而业务应该完全运行Kubernetes的容器中。



# 大数据容器化的公司SequenceIQ

# 大数据平台是否有云化的趋势？