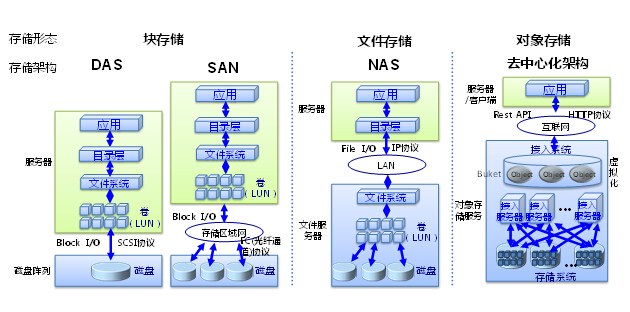
对象存储相关概念

针对不同的应用场景，选择的分布式存储方案也会不同，因此有了对象存储、块存储和文件系统存储，这三者的主要区别在于存储接口：

* 块存储（硬盘），这种接口通常以Kernel Module的方式存在，这种接口需要实现Linux的Block Device的接口，如Sheepdog、AWS的EBS等，其IO特点与传统硬盘是一致的
* 文件存储（文件系统），通常是支持POSIX接口，和传统的文件系统如Ext4是一个类型，但区别在于分布式存储提供了并行化的能力，如CephFS、GFS等，HDFS这种非POSIX接口的类文件存储接口也可以归入此类。文件系统中由文件目录、子目录、文件等形成一个自上而下的文件结构，文件系统下的文件通过应用程序进行打开和修改等
* 对象存储（键值数据库），通常指键值存储，其接口是简单的GET、PUT、DEL和其他扩展，如Swift、S3，Linkedin Ambry等。在对象存储中，一个对象可以看成是一个文件，只能全读全写，通常以大文件为主。对象存储面向对象/文件的海量互联网存储，对象是被封装文件，不能直接打开/修改文件，但可以想FTP一样上传/下载文件等。在对象存储中，没有文件系统的层级文件结构，而是只有一个桶的概念（存储空间）。桶其中存储对象，这是一种扁平化的存储方式。其最大的特点是对象名称就是一个域名地址。

这三种存储形态和存储架构的示意图如下所示：

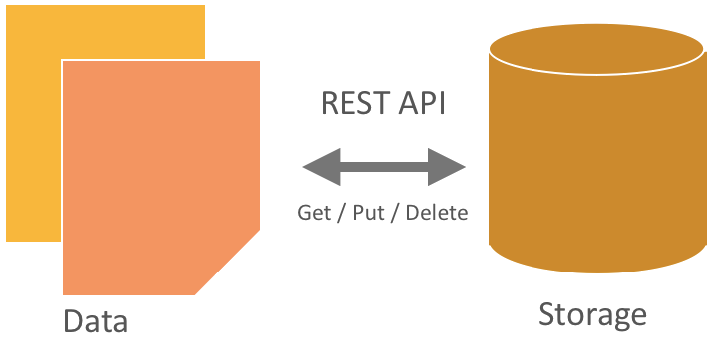


对象存储是无层次结构的数据存储方法，通常用于云中，不同于其他数据存储，基于对象的存储不使用目录树。各个单独的数据（对象）单元存在于存储池中的同一级别。每个对象都有唯一的识别名称，供应用进行检索，此外每个对象可包含有助于检索的元数据。对象存储具有如下特点：

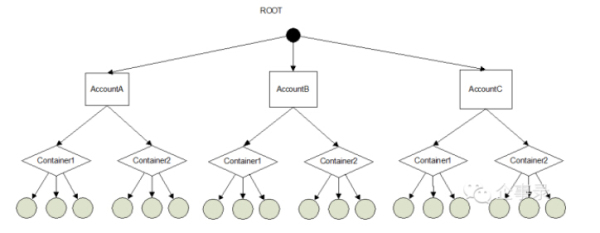
* 数据作为单独的对象进行存储
* 数据并不放置在目录层次结构中，而是存在于平面地址空间内
* 应用通过唯一地址来识别每个单独的数据对象
* 可用代客泊车作类比，每个数据对象就像是一辆车，地址是收据
* 专为使用API在应用级别（而非用户级别）进行访问而设计

使用对象存储可容纳几乎任意数量的数据，而无需对数据集进行分区，由于对象存储缺少层次结构，因此目录简单，数据的读取比较高效。在对象存储中具有一些高级特性，能够确保数据一致性，实现自动数据复制和滚动更新，以及消除宕机情况。

对象存储的接口是REST风格，通常是基于HTTP协议的RESTful Web API，通过HTTP请求中PUT和GET等操作进行文件的上传即写入和下载，通过DELETE操作删除文件，如下图所示：



对象存储采用扁平的数据组织结构，往往是两层或者三层，每个用户可以把它的存储空间划分为容器（Bucket），如下图所示：



在OpenStack的Swift中将整个存储分为三个层次：Account、Container和Object，必须使用一个或者多个名称，系统才能为数据提供独一无二的存储位置，描述如下：

*/account*

账户存储位置是一个特殊的存储区域，包含账户本身的元数据（ 信息）以及账户中的容器列表。Account不是用户身份，而是指存储区域。

*/account/container*

容器存储位置账户中由用户定义的存储区域，容器中存储的是本身的元数据和对象列表

//account/container/object

对象存储位置的内容是该对象和其本身的元数据

在集群中由账户、容器和对象组成的完整的路径不可重复，但是不同账户下的容器和对象名可以重复。

# Account

账户是数据存储的根目录，在系统中可以将账户类必成文件系统中的卷，账户就是卷名。每个账户都由一个数据库来存储账户的缘数据以及账户中所有容器的列表。账户的元数据头是以键值对的形式存储，这些数据与其他所有Swift数据一样，有相同的持久性。

账户数据库如果不存储对象及其元数据是没有任何意义的，当一个用户请求账户的元数据或者账户中所有的容器列表时，就需要访问账户数据库。可以根据需求设置账户的访问权限开发给单用户还是多用户，比较常见的是，单用户访问权限一般用于个人或服务账户，如backups；多用户访问权限权限一般应用于项目账户，账户中相应权限的用户可以创建、修改、删除容器或则对象。

在对象存储中，账户是存储账户，而非用户身份认证。当看到账户时，应该想到这是存储位置。

# 容器

容器是账户的下一级存储空间，由用户定义，可以存储对象。容器不可以嵌套，从概念上来将类似于文件系统中的目录或文件夹。因为容器是账户（/account/container）的下一级，所有不存在全局唯一名称的限制，两个相同名称容器存储在两个不同的账户中时完全可行的。

每个账户都由一个数据库，容器也是一样。每个容器数据库中都存储容器的元数据及其包含的对象列表。容器元数据头也是以键值对的形式存储的，这些数据也是持久性。例如一个容器元数据头-X-Container-Read，包含了有权访问容器对象的授权列表。

像账户数据库一样，容器数据库中并不存储实际的对象或对象元数据，当用户请求容器中所有对象列表或者容器中元数据时就会访问容器数据库。在一个账户中，可以创建和使用容器合理的管理对象，也可以为容器增加元数据。

# 对象

对象是任何数据，可以是照片、视频、文档、日志文件、数据库备份、文件系统快照或者任何其他类型的数据。每个对象通常包括对象本身及其元数据。

元数据头部是以键值对（名称和值）的方式存储的，有着与对象数据库一样的持久性保障，检索时不需要额外的延迟。元数据可以提供对象的重要信息，例如一个视频创作室可以将录像格式和录像内容一起存储。一个文档可以包含作者信息，一个基因序列可以包含它的产生信息。

每个对象必须从属于一个容器，如果一个对象被存入集群，那么用户总是使用存储位置（/account/container/object）来引用它。每个容器中可以存储的对象数量不受限制。

从用户角度上看，对象存储位置是用来定位一个对象的，例如Swift实际上为每个对象存储多个副本并且跨集群分布，以保障数据可靠性和可用性,Swift将数据放置成为一个分区（partition）的逻辑组中，一个分区映射到多个驱动器。

https://www.ibm.com/cloud-computing/cn-zh/object-storage/