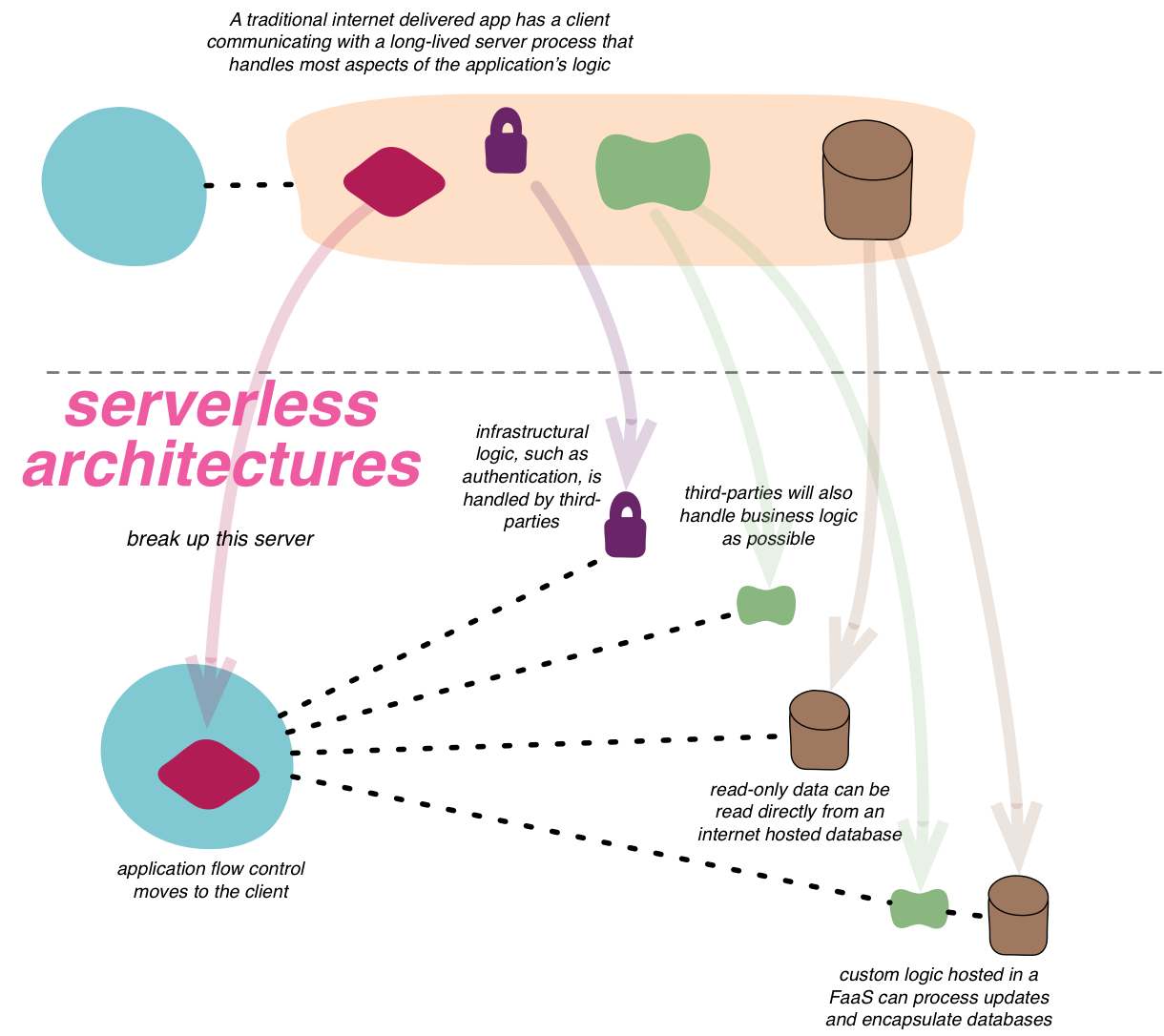
Serverless架构

目前大多数应用程序部署在服务器上时，无论是公有云/私有云，都需要提前规划服务器规模、多大存储及数据库功能，并且需要将应用程序和依赖的软件部署到基础设施上。Serverless(无服务器)架构简化了以上的精力，其定义描述如下：无服务架构是基于互联网的系统，其中应用开发不使用常规的服务进程，相反，它们依赖于第三方服务（AWS Lambda服务等），客户端逻辑和服务托管远程调用的组合，目前各大厂商都提供了的Serverless平台，包括AWS Lambda、Microsoft Azure、Google Cloud Platform,IBM Bluemix,Kubeless等。

Serverless是一个无服务器的架构，与传统架构不同，由开发者实现的服务端逻辑运行在无状态的计算容器中，由事件触发、短暂（可能只存在一次请求过程），完成被第三方管理，可以和Faas(Function as a Service)进行类比，但是这两个概念有重要的区别：

* FaaS是一个计算服务，主要功能有部署、可伸缩性、执行和结算
* Serverless架构属于平台即服务（PaaS），针对事件驱动、短暂性的工作负载，其结合了FaaS与其他云服务来构建复杂的系统，同时提供可伸缩性、可操作性、综合安全性以及按使用率来计算，其系统架构图如下所示：



Serverless代表着未来云服务正在走向越来越分离的趋势，业务系统不直接与硬件、操作系统和容器打交道，而是通过一个更高级的容器运行业务系统。业务系统向容器的管理中心申请各种资源，其只需要关心自身业务。在该架构中为每个事件/请求启动一个全新的进程运行代码是Serverless的核心，其无状态是其基本特点。

# Serverless定义

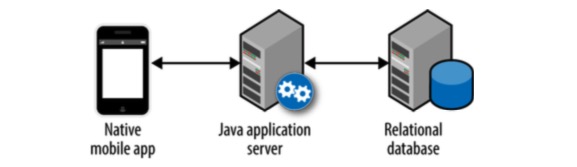
Serverless架构可以理解为云的延伸，通常包含两个领域BaaS(Backend as a Service)和FaaS( Function as a Service)

1. BaaS，后端即服务，通过API调用后端实现好的程序逻辑，其旨在为移动和Web应用提供后端元服务，比如身份验证服务Auth、数据管理、应用数据分析等。
2. FaaS，函数即服务，无服务器计算的一种形式，本质上是一种事件驱动的由消息触发的服务，FaaS中集成各种同步和异步的事件源，通过订阅这些事件源，触发函数的执行。这些函数运行在无状态的容器上，FaaS不需要关心后台服务器或者应用服务，值需要关系自己的代码即可。

下面以一个游戏应用为例，其包含如下特性：

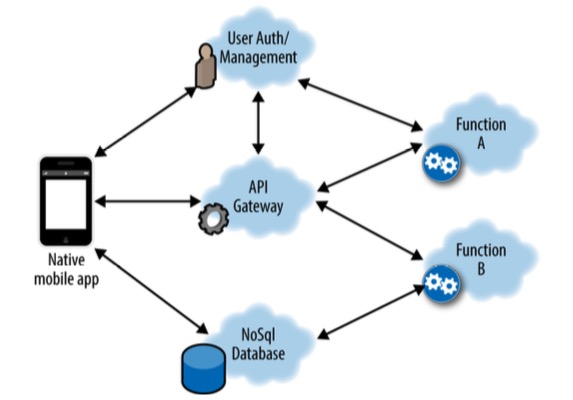
* 移动端友好的用户体验
* 用户管理和权限认证
* 关卡、升级等游戏逻辑、游戏排行，玩家等级、任务等消息

在传统的应用程序架构如下图所示：



前端只需要渲染用户界面，将请求通过HTTP发送给后端，所有的数据操作都是有后端的Java程序来完成。这种架构维护复杂，前后端都需要专业人员进行维护。

Serverless架构如下图：



在服务端代码中不需要存储任何会话状态，而是存储在NoSQL中，这样使应用程序无状态，有助于弹性扩展。前端可以直接使用BaaS而减少后端的开发需求。这种架构减少应用程序开发的人力成本，降低自己维护基础设施的成本，重复利用云的能力更便于扩展和快速迭代。

Serverless中，BaaS提供业务依赖服务，FaaS提供业务的部署和生命周期管理，其角色和PaaS相同，但是PaaS以程序为粒度管理应用的生命周期，而Serverless是以函数粒度管理应用生命周期，在PaaS中的应用为常驻内存的进程，而Serverless应用运行完成即销毁。

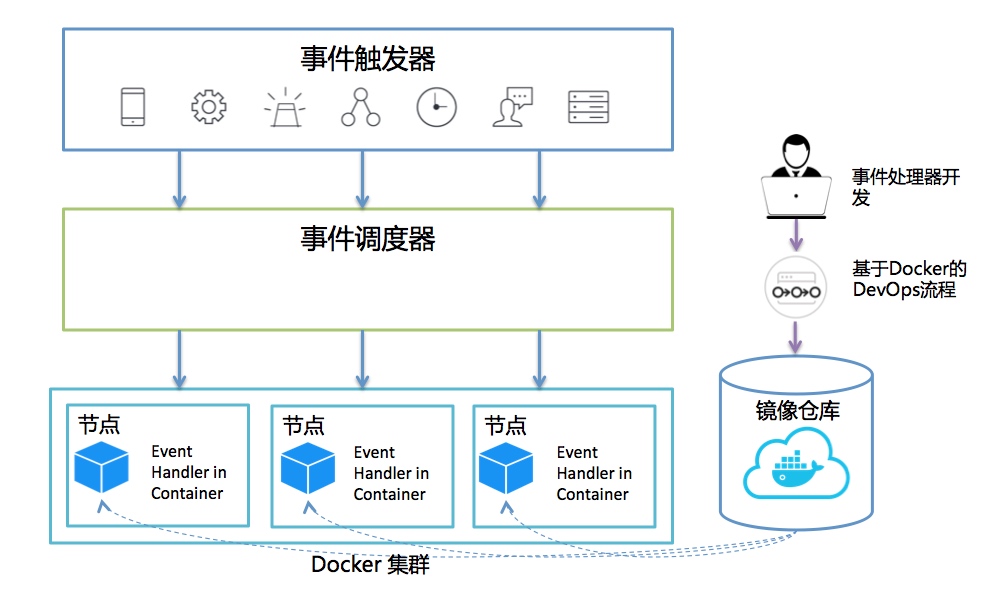
# Serverless架构分析

Serverless应用通常基于Event-Driven编程范型，其开发方式和经典的ECA( Event-condit

ion-Action)类似，其通常包含如下方面：

* 事件（Event）触发器，用于描述触发应用逻辑
* 事件处理器，无状态、原子化的任务，并能够从系统的上下文中进行数据交换
* 事件的派发和调度，开发者可以声明事件处理器对底层计算需求，由系统根据需求自动分配计算资源并调度执行

可以使用多种技术提供Serverless计算环境，每次系统接收到事件，动态启动容器来执行业务逻辑即可，下面介绍一个基于Docker的Serverless平台的一个高层次参考架构，如下图所示：



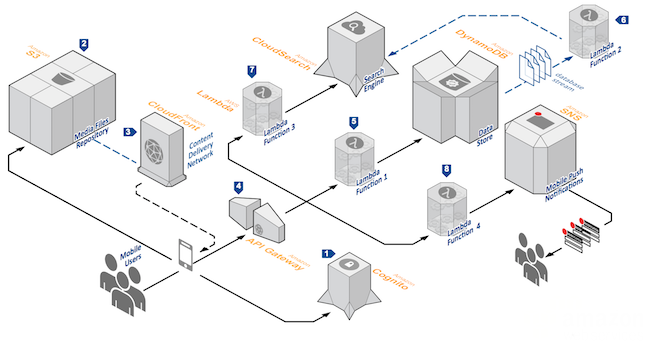
在该架构中，将函数化事件逻辑打包成Docker镜像存储在镜像仓库中，事件调度器通过Docker集群来调度容器来处理调度事件，Docker容器为事件处理提供计算环境。

使用Docker能够敏捷的创建事件执行环境，并提供基本的资源、安全隔离能力，容器之间可以自由选择使用不同语言和框架进行事件处理，不互相干扰。使用Docker镜像仓库，可以简单标准地对事件处理逻辑进行打包和分发，而且Docker编排技术可以提供集群管理和编排能力，简化事件调度器的实现。

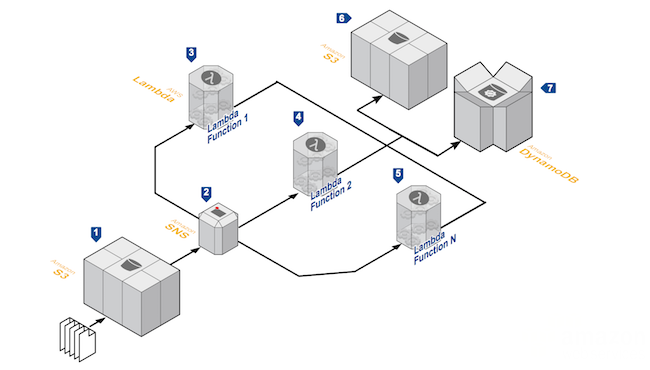
# Serverless的架构范式

下面介绍几种Serverless架构范式：

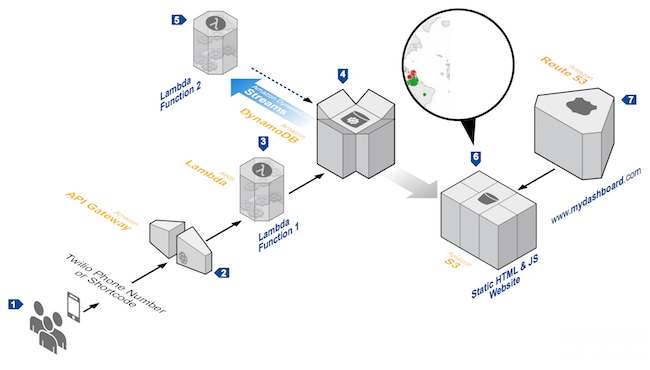
1. 移动应用后台Serverless参考架构



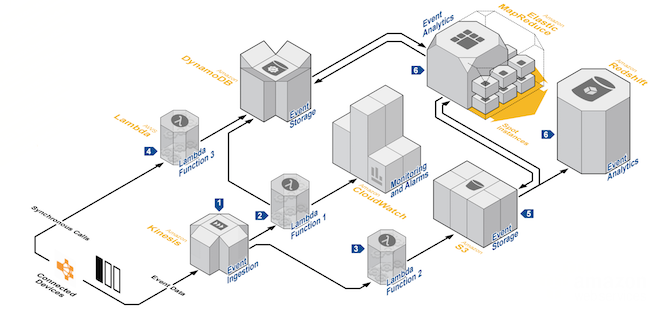
1. 实时文件处理Serverless参考架构



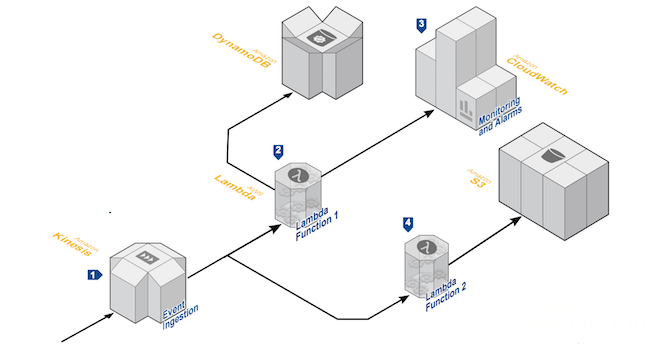
1. Web应用Serverless参考架构



1. 物联网应用后台参考架构



1. 实时流处理Serverless参考架构



参考文献：

https://aws.amazon.com/cn/blogs/china/iaas-faas-serverless/

https://blog.csdn.net/chdhust/article/details/71250099?utm\_medium=referral&utm\_source=itdadao