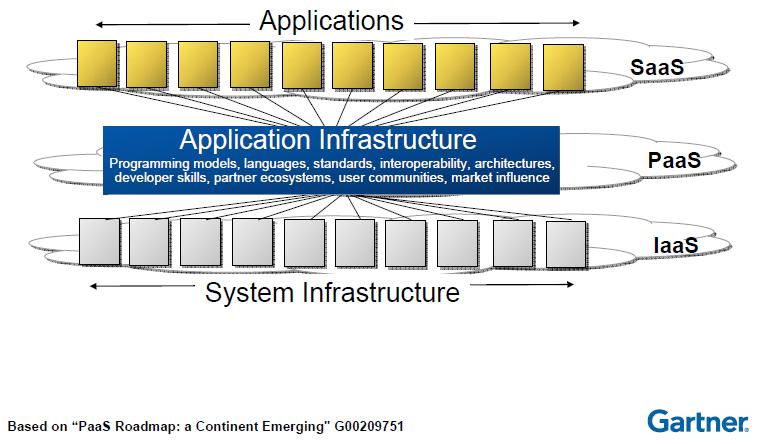
Tomcat应用路由转发

# 1、PASS平台简介

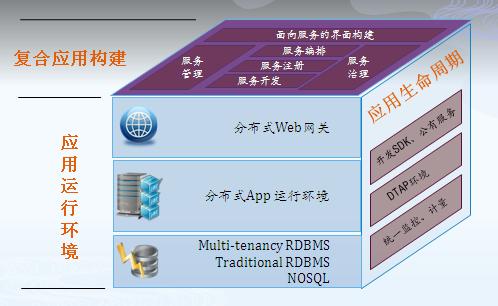
PASS，平台即服务，把服务器平台作为一种服务提供的商业模式。PASS实际上是指将软件研发的平台作为服务，以SAAS的模式提交给用户，因此PASS也是SASS模式的一种应用。

从传统角度来看，PASS实际上就是云环境下的应用基础设施，可理解成中间件即服务，如下图所示：



PASS为部署和运行应用系统提供所需的基础设置资源应用基础设施，所以应用开发人员无需要关心应用的底层硬件和应用基础设施，并且可以根据应用需求动态扩展应用系统所需的资源。

PASS的全局功能视图如下：



PASS应该包括：

1）应用运行环境：分布式运行环境，多种类型的数据存储及动态资源伸缩

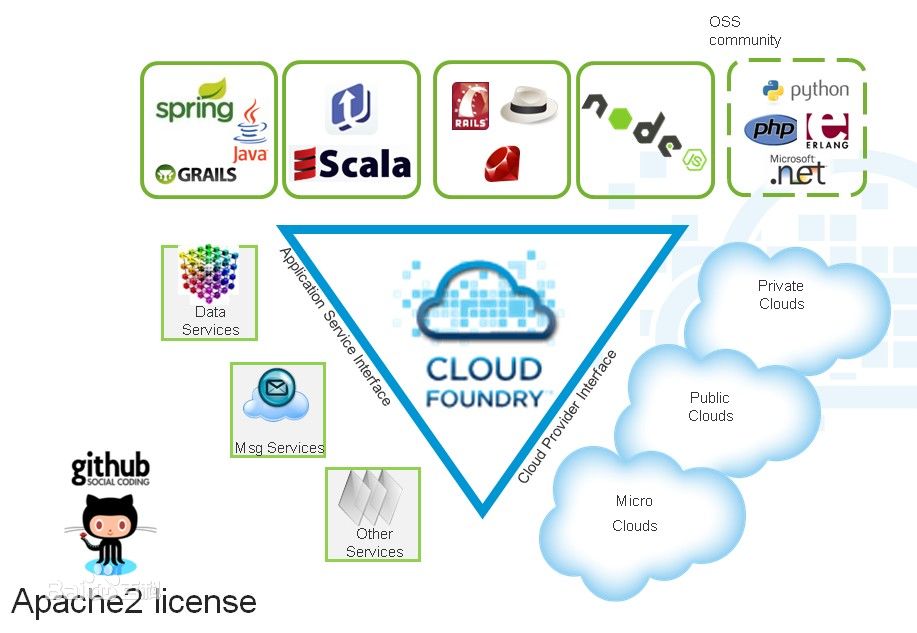
2）应用全生命周期支持：提供开发SDK、IDE等加快应用的开发、测试和部署；以API形式提供公共服务，如队列服务、存储服务和缓存服务；监控、管理和计量，提供资源池、应用系统的管理和监控功能，精确计量。应用使用所消耗的计算资源。

3）集成、复合应用构建能力，除了提供应用运行环境外，还需要提供连通性服务、整合服务、消息服务和流程服务等用于构建SOA架构风格的复合应用

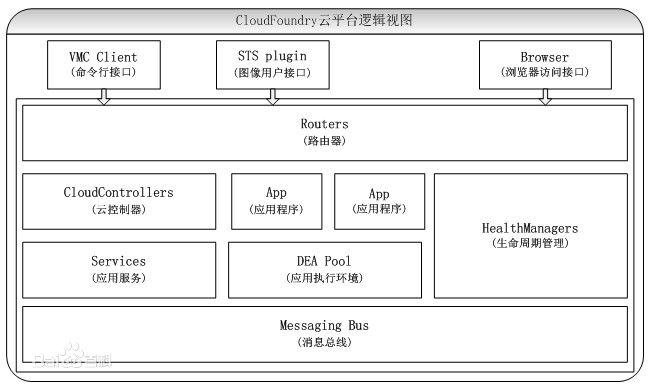
PASS的特性有多租户、弹性、统一运维、自愈、细粒度资源计量、SLA保障等。多租户是指一个软件系统可以被多个实体所使用，每个实体之间是逻辑隔离、互不影响的。多租户弹性是指租户或者租户的应用可以根据自身需求动态的增加、释放其所使用的计算资源，基本的实现方式：Shared-Nothing、Shared-Hardware、Shared-OS及Shared-Everything等。

# 2、VMware's Cloud Foundry

Cloud Foundry是VMWare推出的业界第一个开源PASS云平台，支持多种框架、语言、运行时环境、云平台即应用服务，使开发人员能走在几秒内进行应用程序的部署和扩展，无需担心任务基础架构的问题。

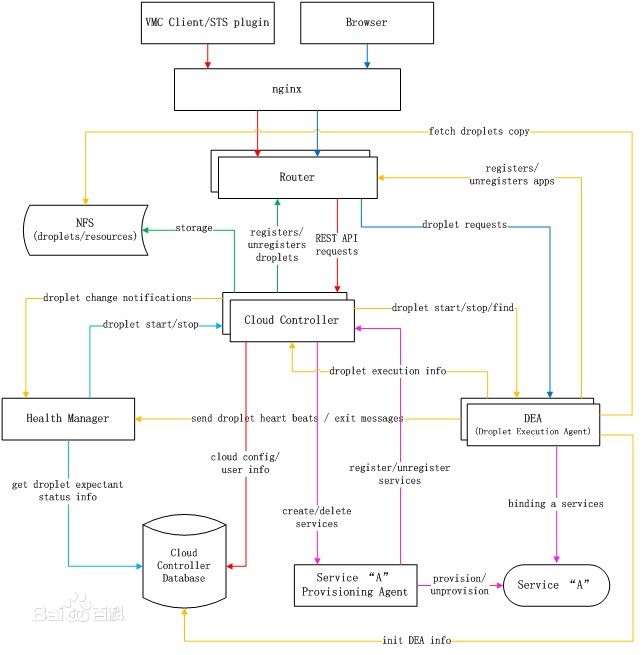


Cloud Foundry是由相对独立的多个模块构成的分布式系统，每个模块单独存在和运行，各模块之间通过消息机制进行通信，整体逻辑组成如下图所示：



这种模块化的分布式系统，使耦合度降低，使系统功能容易扩充。开发人员可通过VMC命令行工具或STS插件方便的部署应用程序到Cloud Foundry云平台上，最终用户可以通过浏览器访问运行在Cloud Foundry云平台上的应用。所有的访问请求都通过Router进行转发，分别由云控制器Cloud Controller和应用运行代理DEA模块进行请求响应。

从下图中可以看出，Cloud Foundry云平台主要有Router、Cloud Controller、Health Manager、DEA、NFS、NATS、Cloud Controller Database以及Service等模块组成。



这些模块协同合作，通过特定的消息传输机制和API接口进行通信，就可以使整个云平台正常运行。由于在集群环境下，每个模块都有多个部署节点，保证了云平台的可靠性和弹性动态扩展的需求，使得应用程序可以稳定可靠的运行在Cloud Foundry云平台上。

# 3、Hadoop Yarn as PAAS

## 3.1 YARN 作为PASS具有的特性

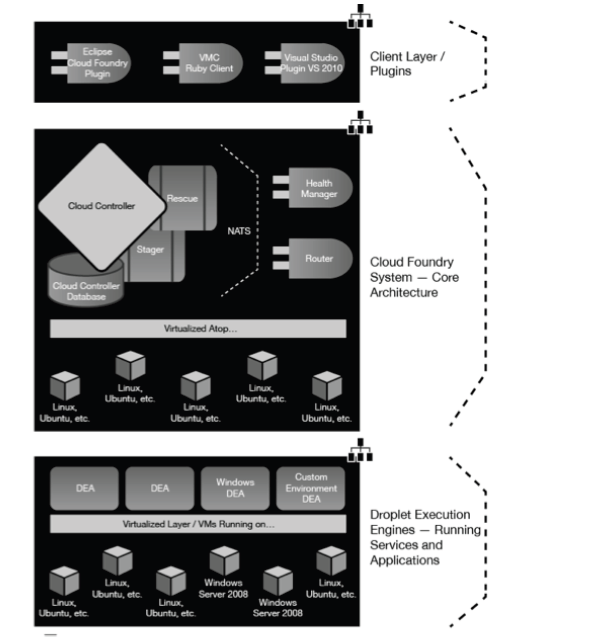
YARN具有以下特点：

* 1. Hadoop是在大规模集群上运行应用的框架，这种特点减少开销和避免依赖于某个厂商的产品
  2. 很多公司部署了Hadoop集群，可以同时用作大规模数据处理和PASS
  3. YARN三层框架（Client/YARN core/App Instance）和PASS比较贴近
  4. YARN监控和资源管理能力（RM及NM），在PASS中比较有前途
  5. Hadoop是JAVA框架
  6. HBASE是Hadoop的NoSQL数据库，可以作为PASS Service

因此，YARN作为PASS，具有可行性，而且YARN本身已经提供了很多PASS实现所需要的技术。

## 3.2 Cloud Foundry与 YARN as PASS对比

Cloud Foundry将PASS分成三层典型框架，Client层，Core层和服务/应用层。三个层次与Hadoop Yarn架构很类似，其架构如下所示：



Cloud Foundry 组件包括Self-service应用执行引擎，应用部署及生命周期自动管理引擎，脚本执行客户端及开发工具，为了扩展的需要，还包括增加框架、应用服务及cloud provider的接口。

1）Client Layer

|  |  |  |
| --- | --- | --- |
| Feature | Cloud Foundary | Hadoop PASS 建议 |
| Management Client | Vmc:命令行管理客户端，上传/更新/管理服务，及获取应用的信息 | 基于YARN Client，来处理命令 |
| Application Provisioning | 基于应用上传，支持war file | 在节点上运行jetty及tomcat，通过Yarn Client将war上传 |

2）Cloud Foundry System Layer — Core Architecture

|  |  |  |
| --- | --- | --- |
| Feature | Cloud Foundary | Hadoop PASS 建议 |
| Request Routing | Router Component基于Nginx | 在YARN中没有类似的Router，需要类似Nginx的http proxy |
| Health Management | Health Manager，周期的进行健康检测 | 在Yarn中ApplicationMaster可以起着Health Manager的作用 |
| Messaging | 使用NAT作为消息通信的核心，用于不同的组件之间同步协作或者状态更新 | HBase使用Zookeeper |
| Main Controller | Cloud Controller | RM和NM作为控制器 |

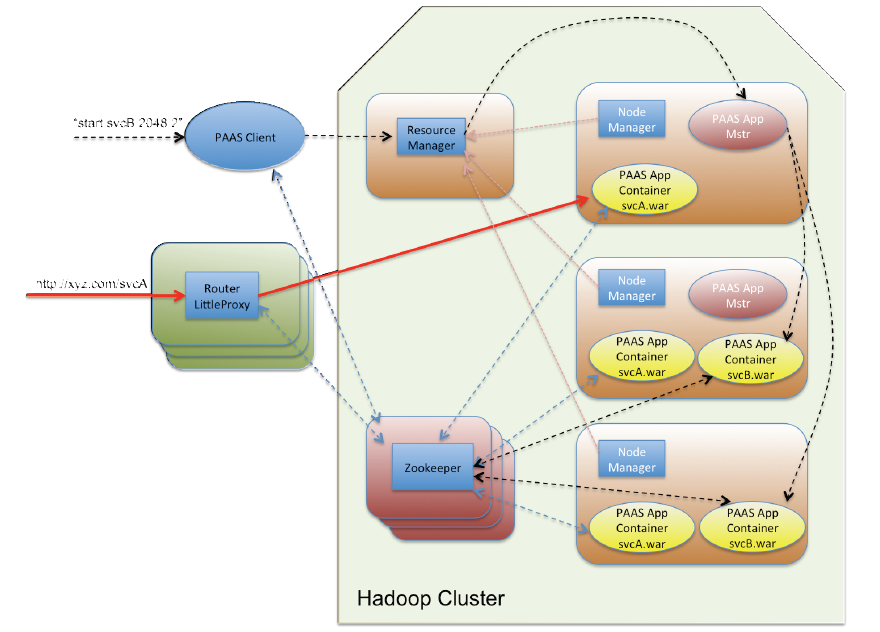
3）Droplet Execution Engines Layer — Running Service And Applicatins

|  |  |  |
| --- | --- | --- |
| Feature | Cloud Foundary | Hadoop PASS 建议 |
| App Running Env | Droploet Execution Engine | Container作为shell执行环境 |
| Java Web Container | 默认在DEE中运行Tomcat作为Web Container | 可以使用内嵌jetty作为web container |
| Service support | 运行注册/注销服务（MySQL，NoSQL）等 | 还没支持 |
| Multi-tenancy | 支持服务层的多租户，应用可以为应用提供数据库服务 | 仅container-level的多租户被支持 |

## 3.3 YARN AS PAAS

通过比较，Hadoop YARN缺少一些PASS组件，但是总体符合PASS需要。其中Router使用LittleProxy(修改)，Router的主要功能是应用在正运行的应用实例中route及分别应用请求。下面介绍Hadoop PASS 的原型系统，主要提供以下几个PASS 功能:YARN Client命令行，提供初始化应用、启动App实例及停止一定数目的应用实例；仅支持Web APP，可以管理WAR文件。

结构流程图如下所示：



为了支持PASS，YARN框架中，会创建4个JAVA项目：

1. PAAS Client(PassClient)：hadoop-yarn-applications-pass-client

定制的YARN Client，使用方法类似于命令行。用户可以通过命令行提供APP文件，停止或启动应用，获取实例的信息及状态。

1. PAAS Application Master(PaasAppMaster): hadoop-yarn-applications-pass-master

PASS App Master，用于管理PASS应用的生命周期。对于每个Start命令，创建一个PaasAppMaster，然后创建PaasAppContainer实例。整个生命持续到PassAppContainer实例的停止或失败。

1. PASS Application Container(PassAppContainer)：hadoop-yarn-applications-pass-container，以Jetty Web Container为案例示例，PassAppMaster启动PassAppContainer，container有资源限制(例如Memory)。仅封装了内嵌的Jetty Web Container，加载War file，并与Zookeeper通信完成注册和撤销功能。Router可以根据注册信息更新Zookeeper中的routing table。
2. PASS Zookeeper Client Library:hadoop-yarn-applications-zkClient

集中式服务，提供分布式同步，获取主要的配置和提供组服务。Zookeeper主要提供两个作用：记录运行应用实例，PassAppContainer实例启动或者停止时，将Container向Zookeepr中注册或注销。这些信息都会提供给Router; YARN不提供单独Containers的管理，因此与PassAppContainer实例的协作，来停止该应用实例。

1. Router

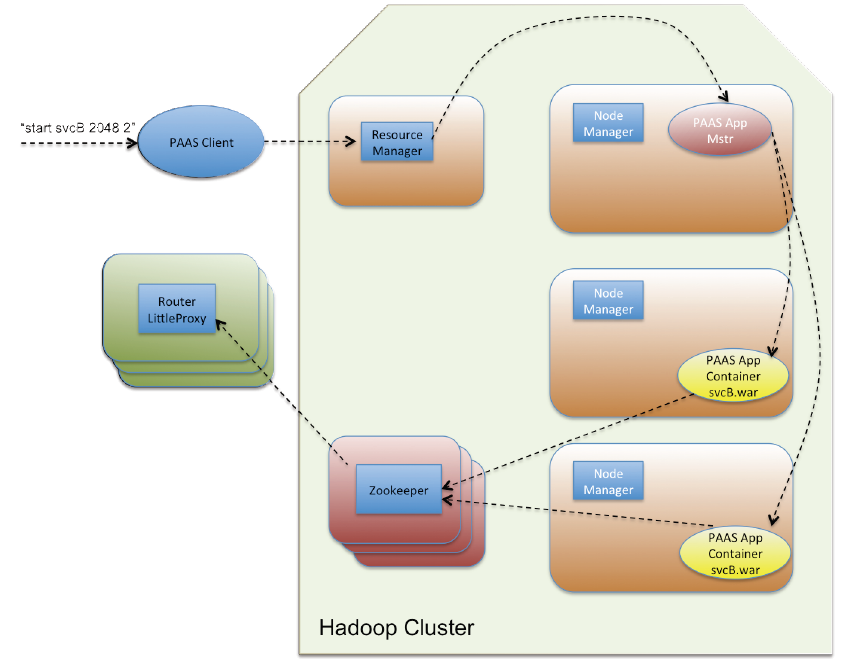
主要功能是路由功能，应用请求分发到正在运行的应用实例中。Hadoop YARN PASS中，使用LittleProxy，Java实现的Http Proxy，使用Netty异步网络框架。为了支持Hadoop YARN PASS，修改LitterProxy，可以从Zookeeper中获取rounting table，然后给予routing table分发请求。

## 3.4 应用启动流程

### 3.4.1 App启动

简单的来说，Application war文件已经上传到了HDFS，当start命令执行后，AppMaster从HDFS中获取war文件。

顺序如下：管理用户通过Client端发送"push /User/PAAS/svcA.war类似的命令，然后Client将文件上传到Hadoop Server（使用scp或者ssh）。在Hadoop文件系统的目录/PASS中存储war文件，同时non-application lib及jar 文件存储到不同的目录中。流程如下图所示：



应用的启动，先判断pass 文件及war file是否上传成功，然后进行start操作：

1) 在client端，执行start命令

2）Pass Client，与RM通信，启动AppMaster，参数中会有启动的components数目及消耗资源

3）AppMaster请求RM，分配containers，启动AppContainers

4）AppContainers，从hdfs中获取war文件，启动Jetty Web Container

5）Container注册到Zookeer中，（Jetty web continaer的service URL）

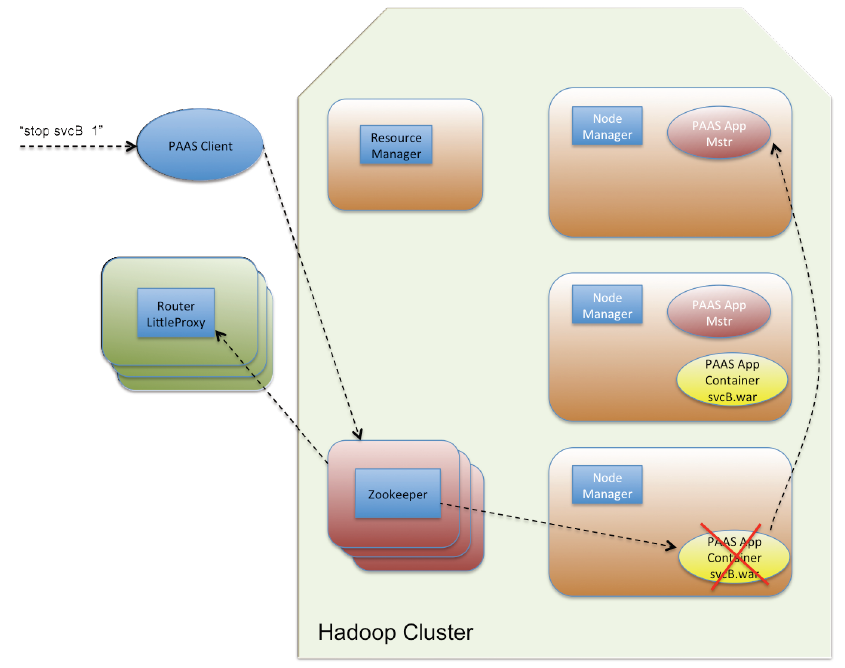
6）注册事件会通知给Router，然后更新Routing tables。至此，所有的操作完成

注：

start命令执行时，可能YARN中已经PassAppMaster，虽然有，但是会重新执行（用YarnRPC协议来解决）。

### 3.4.2 App停止

YARN不能够停止单Container，YARN as PASS原型使用Zookeeper的同步机制来执行stop命令。App停止的流程如下所示：



1. 管理用户，执行stop命令，类似于stop svcA 2，含义是停止2个svcA应用
2. PASS-Client，从启动的实例中随机选择请求数量的实例，并从Zookeeper中删除对应的注册信息
3. un-registrations事件会通知给对应的PaasAppContainer实例，然后AppContainer自己停止工作(可以选择更合理的方式)

# 4.Slider与Yarn As PASS

仿照YARN AS PASS，Slider结构流程图（目前还没有Router）如下所示：

缺少Router

缺少REST

# 5.Router的作用

5.1 维护路由表

5.2 转发请求

# 6. Router开发

6.1 Service注册component的时候，同时注册端口

6.2 开发ConnectionWatcher，用于维护service信息

Map<String,ServiceNode>，ServiceNode:name,path,list<String> instances

6.3 添加Zookeeper事件，修改Zookeeper数据时，更新service信息

6.4 增加RouterClient，获取Zookeeper中的Service信息，主要是ip:host

6.5 增加Router，维护一个RouteTable，当访问某个链接从routtable中找该IP:PORT，然后进行操作

6.6 维护一个代理，LittleProxy，如果写到Router中

# 1、简介及需求

BDOC中，用户可以提交创建Tomcat的请求，创建后访问方式为：http://ip:port/app

但是不能将ip及port暴露给用户，最好是提供类似http://bdoc-ip/user/app类似的链接，因此需要提供一个路由的机制。

# 2、LittleProxy

用Java实现的HTTP代理（使用Netty 异步网络框架），可以修改后将Table放到Zookeeper中，然后根据routing table进行路由请求。当实例在Zookeeper中注册或者注销，该事件会通知Router，同时修改routing table。

# 3、Nginx

Nginx是一个高性能的HTTP和反向代理服务器。