YARN Service Framework的编译、安装和使用

# 编译、配置和启动

1）编译

从gerrit上下载源码，打开后执行如下：

*mvn package -Pdist,native,yarn-ui -DskipTests -Dtar -Dhbase.profile=2.0*

编译出yarn-ui依赖的webapp，见目录target/webapp

2）配置

在yarn-site.xml中添加配置

*<property>*

*<name>yarn.webapp.api-service.enable</name>*

*<value>true</value>*

*</property>*

*<property>*

*<name>yarn.webapp.ui2.enable</name>*

*<value>true</value>*

*</property>*

重启YARN

# 创建Yarn Service

Yarn Service Framework有两种使用方式，这两种方式都是直接调用api-service来实现

1. Hadoop客户端

使用以下提供的JSON文件运行命令：

*$yarn app -launch my-sleeper sleeper.json*

*{*

*"name": "sleeper-service",*

*"version": "1.0.0",*

*"components" :*

*[{*

*"name": "sleeper",*

*"number\_of\_containers": 2,*

*"launch\_command": "sleep 900000",*

*"resource": {*

*"cpus": 1,*

*"memory": "256" } }]*

*}*

获取应用程序的状态

*$ yarn app -status my-sleeper*

*{*

*"name": "my-sleeper",*

*"id": "application\_1535536235958\_0001",*

*"lifetime": -1,*

*"components": [{*

*"name": "sleeper",*

*"dependencies": [],*

*"resource": {*

*"cpus": 1,*

*"memory": "256"*

*},*

*"state": "STABLE",*

*"configuration": {*

*"properties": {},*

*"env": {},*

*"files": []*

*},*

*"quicklinks": [],*

*"containers": [{*

*"id": "container\_e08\_1535536235958\_0001\_01\_000002",*

*"ip": "10.139.4.107",*

*"hostname": "fys2.cmss.com",*

*"state": "READY",*

*"launch\_time": 1535536941077,*

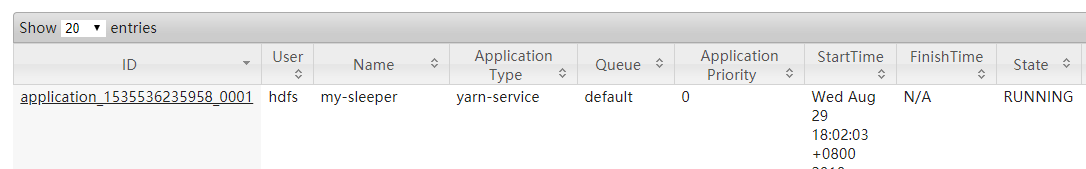
*"bare\_host": "fys2.cmss.com",*

*"component\_instance\_name": "sleeper-0"*

*}, ......*

*}*

查看YARN Web UI中可以查找到对应的服务



其他命令

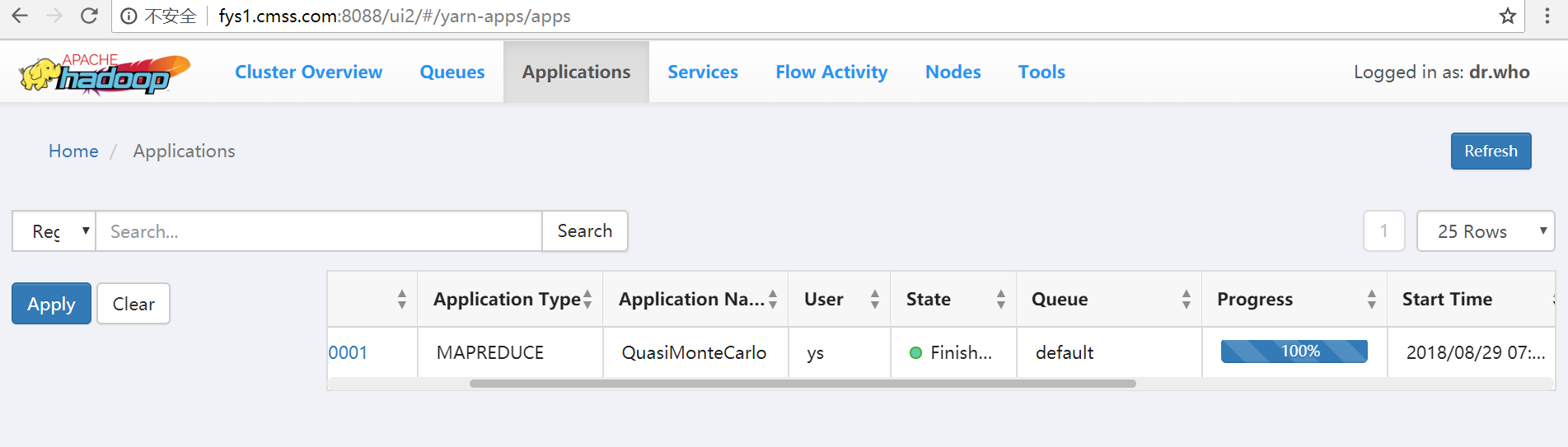
*$yarn app -flex my-sleeper -component sleeper 3 //把容器数量增加到3*

*$yarn app -stop my-sleeper //停止这个服务*

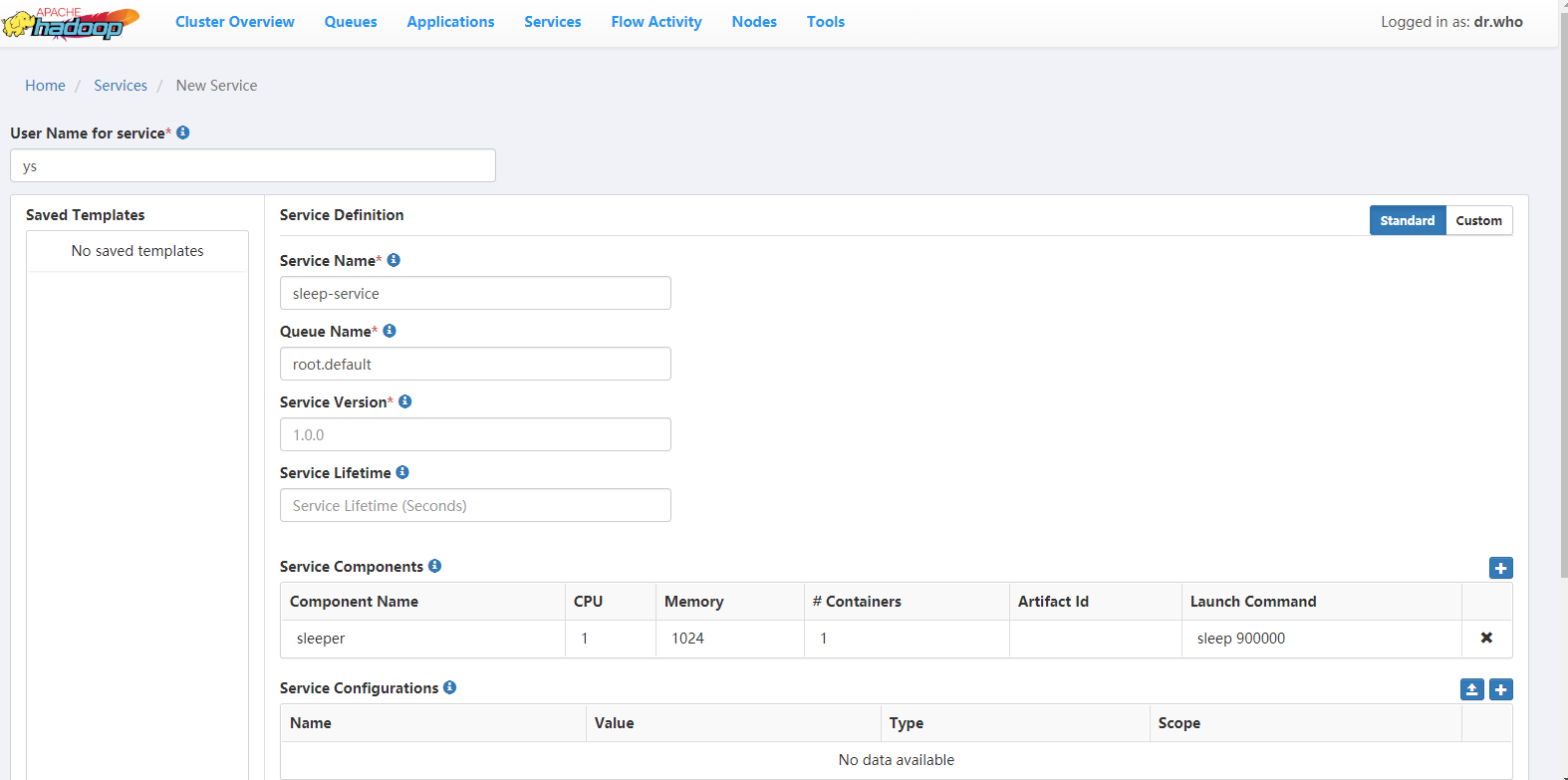
*$yarn app -start my-sleeper //启动停止的服务*

1. YARN Web UI2

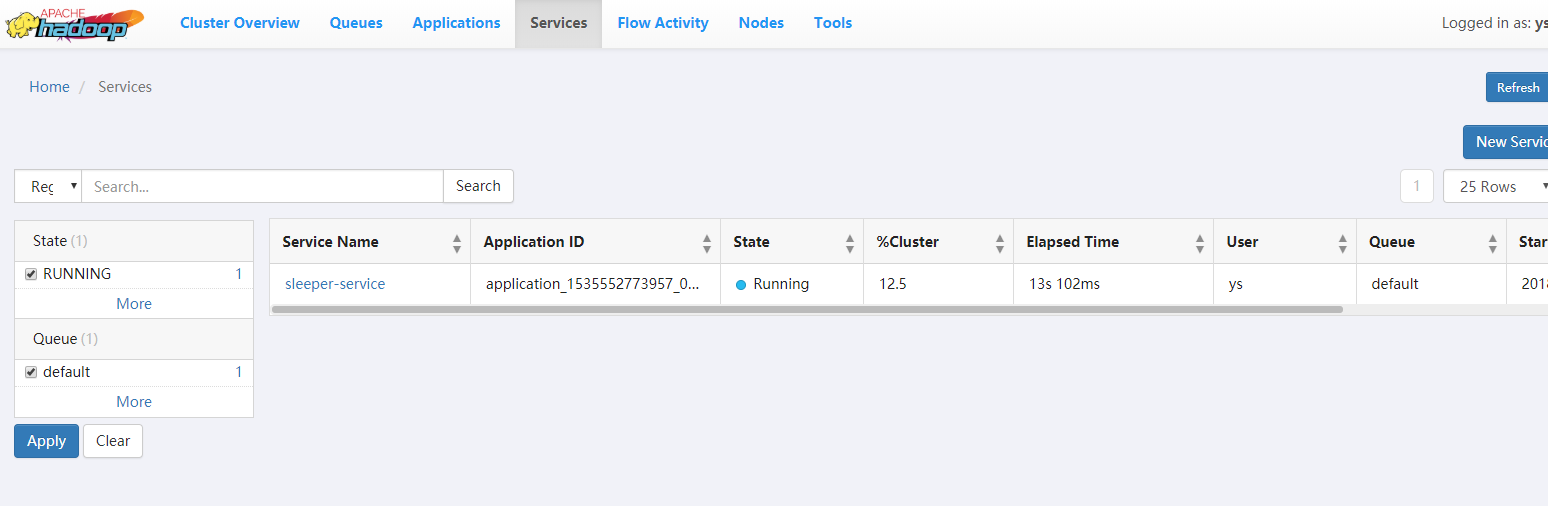
主页面打开如下：



通过YARN Web UI创建Service的页面如下：



配置Service Name， Component使用的资源及启动命令，然后启动，执行后yarn 中信息如下：



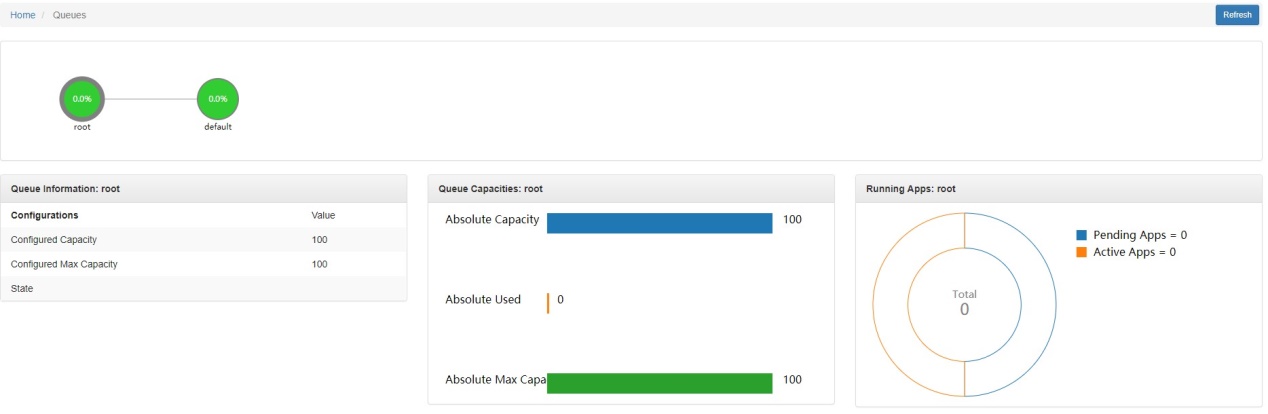
执行效果与通过命令行提交程序的结果相同。

# Yarn UI2界面

下面介绍各tab功能

## 3.1 Queues

界面如下：

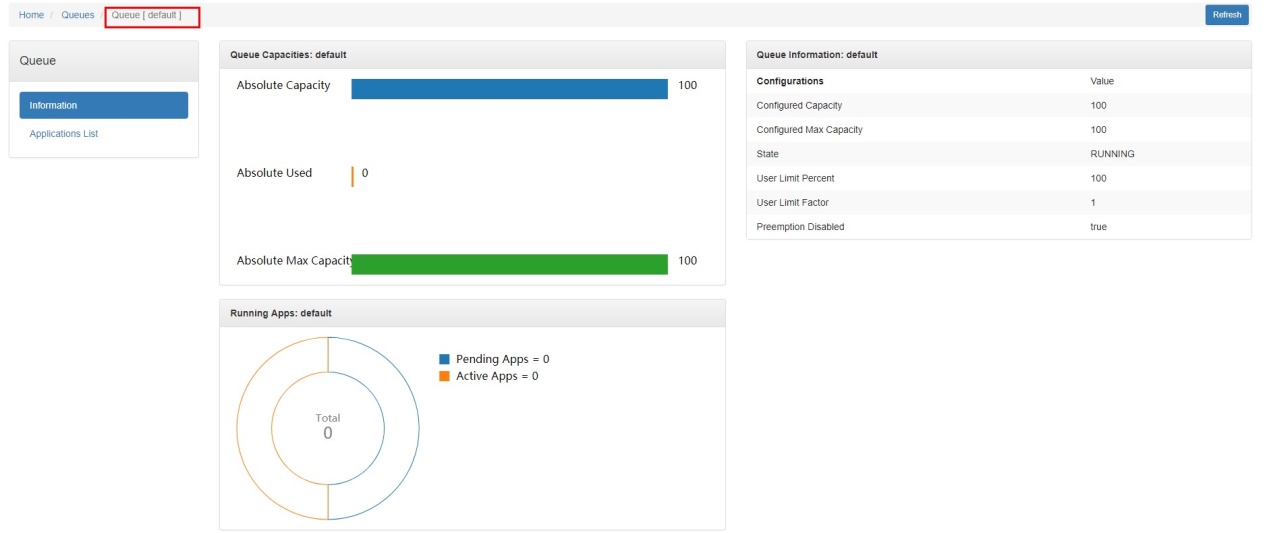


有下面几个功能模块

1. 队列概览图

包括与哪些队列，队列信息，队列容量和提交到这些队列的应用信息

1. 点进去该队列后，可以看到具体的队列数据，如下图所示：



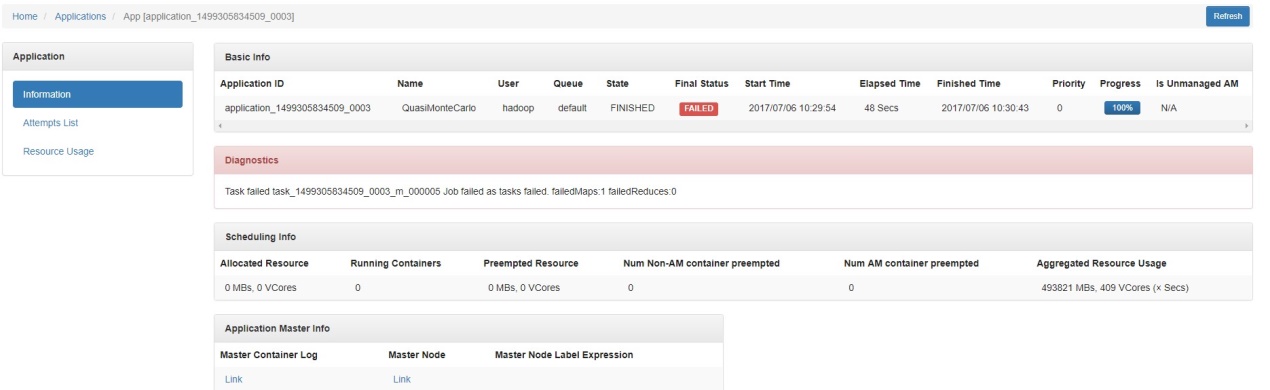
包括：

队列information，容量，提交的应用及配置

在该队列中运行的app，Application List，子界面如下：

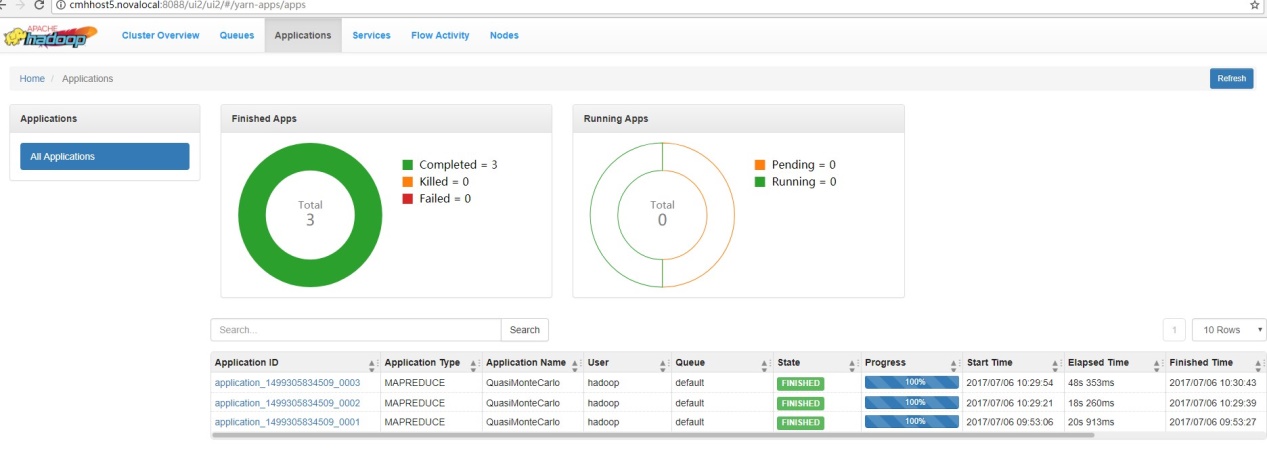


进入Application history Info后，可以看到详细信息：



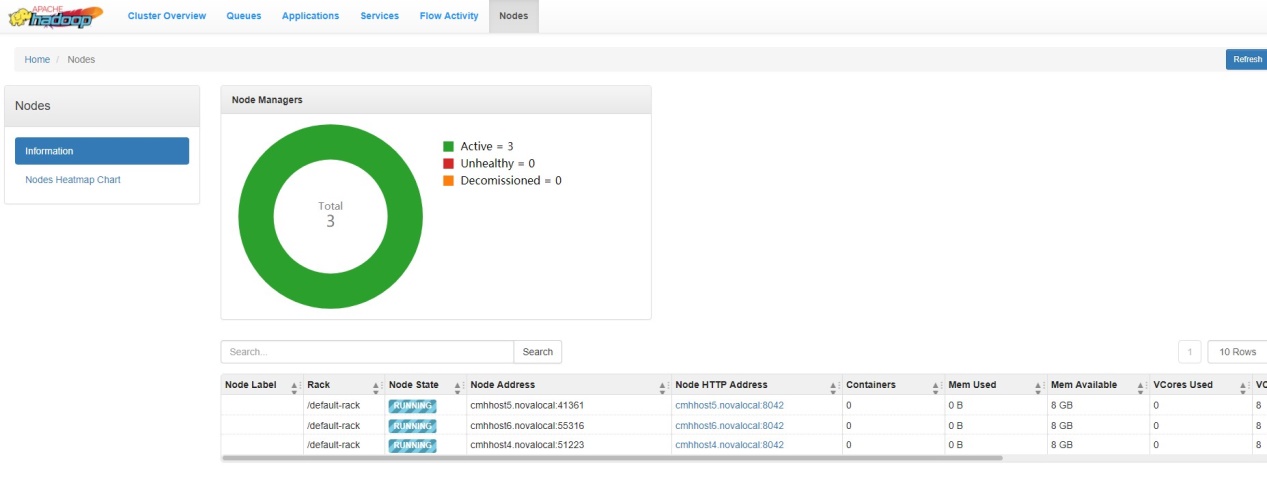
## 3.2 Applications

列举所有的应用，如下图所示：



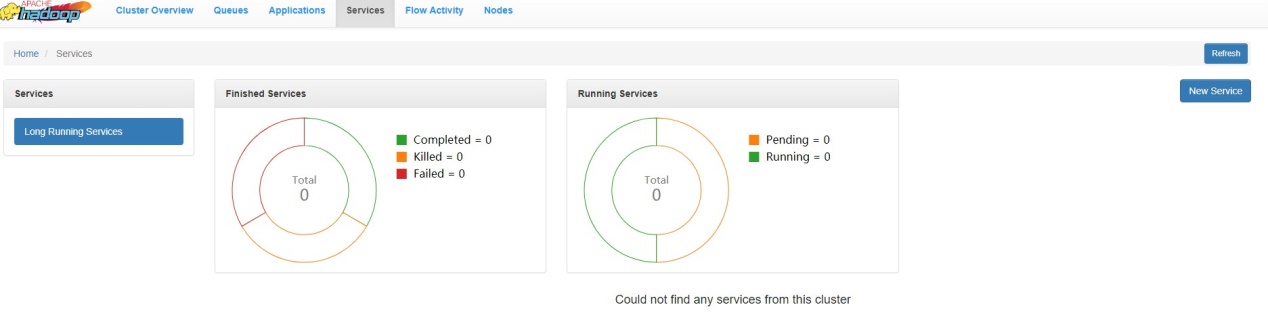
## 3.3 Nodes

所有的节点信息，如下图所示：

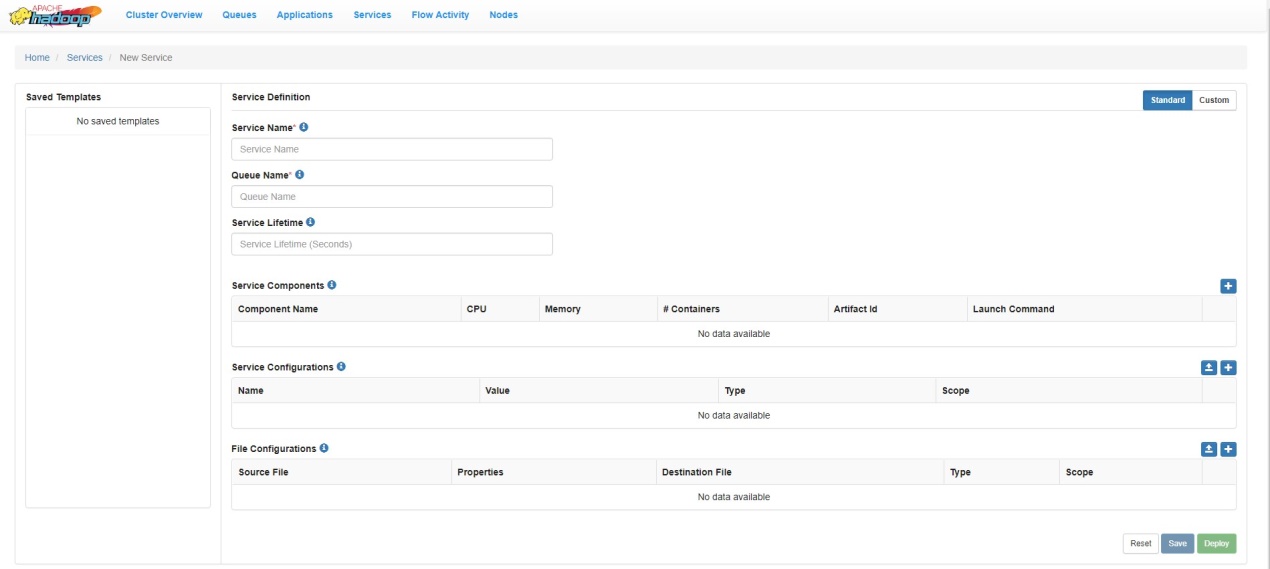


## 3.4 Services

该界面用于创建Long Runnign Service，界面如下所示：

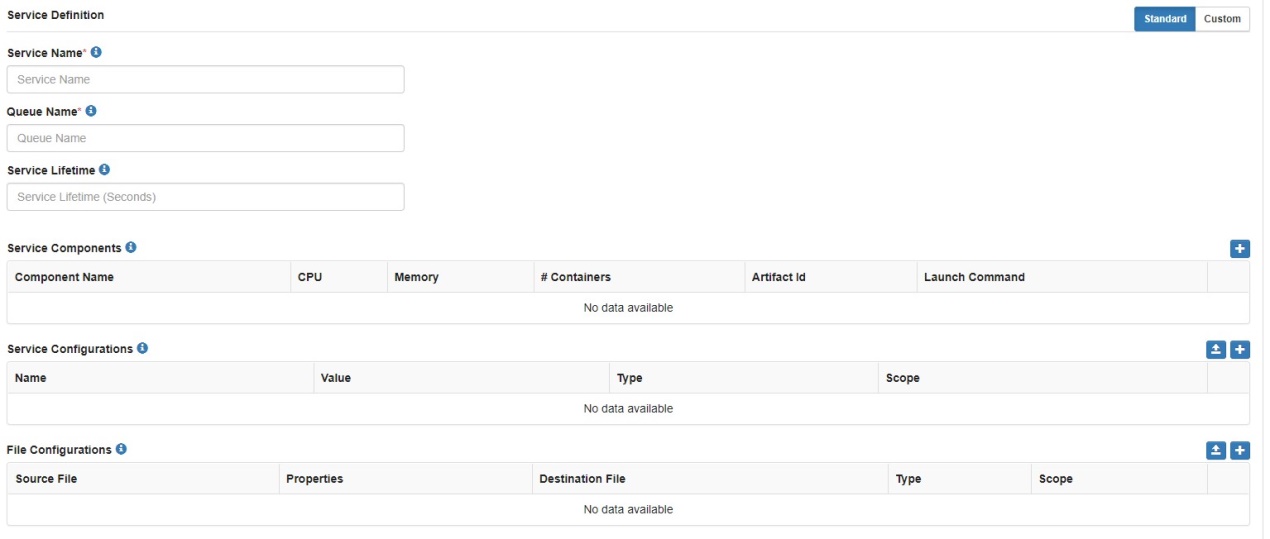


在总览中可以看到所有Finished Service和Running Services。在右上角有创建Service按钮，进入界面，如下所示：



从图中可以看到几个部分：

1. Saved Templates，已经保存好的模板
2. Service Definition，分成两类Standard，在界面上勾选服务，如下：



可以配置服务名，使用队列，生命周期，服务的组件和配置等

Custom，使用JSON配置格式。