

HDFS 的 Web 接口

作者：陆建华，刘东旭，李华盛，陈博洋，李建敦

版本：0.0.2-SNAPSHOT（第二次修订）

时间：2016 年 12 月 15 日

目录

- 在阅读此文章之前 3
- 更新亮点 3
- 开发环境 3
- 测试环境 4
- 使用者 4
- 开发者 5
 - 总体架构 5
 - 起步 6
 - 总线（ActiveMQ 消息队列） 7
 - HDFS 操作（Spring Data） 8
 - 网关（API Gateway） 10
- 故障排查 11
- 计划中的更新 12
- 附加说明 12
 - Kubernetes 的简单安装方法 12

在阅读此文章之前

如果你是使用者，您要会使用 REST 接口。

如果你是开发人员你需要对这些知识有大概的了解才能明白这篇文章在讲什么：

1. Java 8
2. Maven 的使用
3. Spring
4. HDFS 和 HBase
5. CentOS 7
6. docker
7. kubernetes

如果你要对网站前端显示有所修改，你还需要知道：

1. npm 的使用
2. glup 的使用
3. Bootstrap Material Design
4. JavaScript (Echarts)
5. Less

此外，到目前为止。如果你准备在类似中云滴水湖的办公室完成这些工作，你还需要熟练掌握这些：

1. Yum 私有仓库的建设和同步
2. Docker 私有仓库的建设
3. Crontab 计划任务
4. Socket 代理的使用

更新亮点

1. 上传文件请求方式改为 GET
2. 添加读出服务
3. 所有服务迁移到 kubernetes
4. 服务自我修复和弹性计算

开发环境

OS	Fedora release 25 (Twenty Five)
----	---------------------------------

架构	AMD64
Kernel	kernel-4.8.11-300.fc25.x86_64
JRE	1.8.0_112-release-408-b2 amd64
IntelliJ IDEA	2016.3
Apache Maven	3.3.9

测试环境

OS	CentOS Linux release 7.2.1511 (Core)
架构	AMD64
Kernel	3.10.0-327.36.2.el7.x86_64
Docker	1.10.3
Kubernetes	1.2.0

使用者

这个服务目前支持向 HDFS 和 HBase 增加数据和读出数据。

增加数据

发送一个附带文件的 GET 请求，请求头使用：

```
multipart/form-data
```

地址使用

```
/hdfs/add?hdfs_path={path}
```

例如使用 python 将一个名为 flask.pdf 的文件上传到 hdfs 的 /foo 的位置

```
import requests

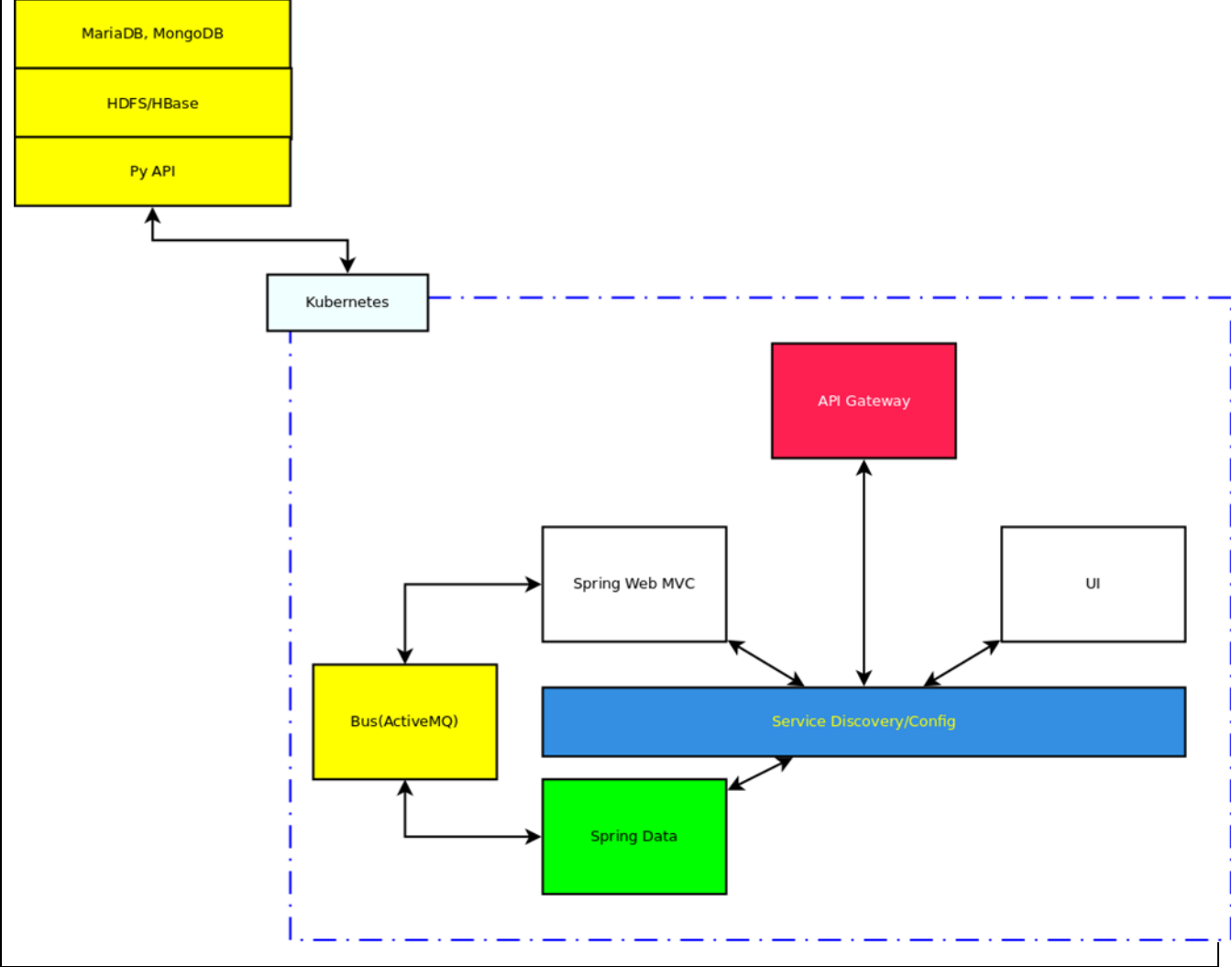
response = requests.post('http://hdfs-api.cloud.sinocbd.com/add?hdfs_path=/foo',
files=dict(file='/home/admin/Documents/flask.pdf'))

print(response.status_code)
```

如果返回的状态是 200 说明上传成功

开发者

总体架构



其中 API 接口（Spring Web MVC），总线（Bus），Apache Hadoop 的操作（Spring Data）以及前端页面属于这篇文档。其他三项服务（服务设置，服务发现和网关）也是系统中必不可少的组件。所有的服务均在 kubernetes 中运行以便出更少的问题。此外 HBase, HDFS 以及其他存储服务可以全部独立于 kubernetes 运行，部分的存储也可以加入 kubernetes 的持久化链统一管理。下表是各个服务的详细功能描述。

kubernetes	容器的管理平台
网关 API Gateway	系统内部与外部通信的接口，提供反向代理和负载均衡。
服务发现，服务设置 Service Discovery/Config	将新服务加入统一听从服务设置的管理。服务设置负责包括但不限于监控其它服务状态以及系统的自我修复。
API 接口 Spring Web MVC	提供向 Hadoop 添加文件的 REST 接口

网页 UI	使用 Echarts 在网页上显示数据
总线 Bus	提供服务之间的通信
Apache Hadoop 的操作 Spring Data	提供直接操作 Apache Hadoop 的服务
附加存储 HDFS/Hbase	这部分目前还不能被 kubernetes 的持久化链所管理，仍在设计之中。

起步

确保安装了第三代的 MAVEN

进入项目主目录可以看到三个目录



hdfs-api 以及 hdfs-op

进入你想要修改的项目，编辑 pom.xml 文件

```
<properties>
  <docker.image.prefix>192.168.1.50:5000</docker.image.prefix>
</properties>
<build>
  <plugins>
    <plugin>
      <groupId>com.spotify</groupId>
      <artifactId>docker-maven-plugin</artifactId>
      <version>0.4.11</version>
      <configuration>
        <imageName>${docker.image.prefix}/${project.artifactId}</imageName>
        <dockerDirectory>src/main/docker</dockerDirectory>
        <resources>
          <resource>
            <targetPath></targetPath>
            <directory>${project.build.directory}</directory>
            <include>${project.build.finalName}.jar</include>
          </resource>
        </resources>
      </configuration>
    </plugin>
  </plugins>
</build>
```

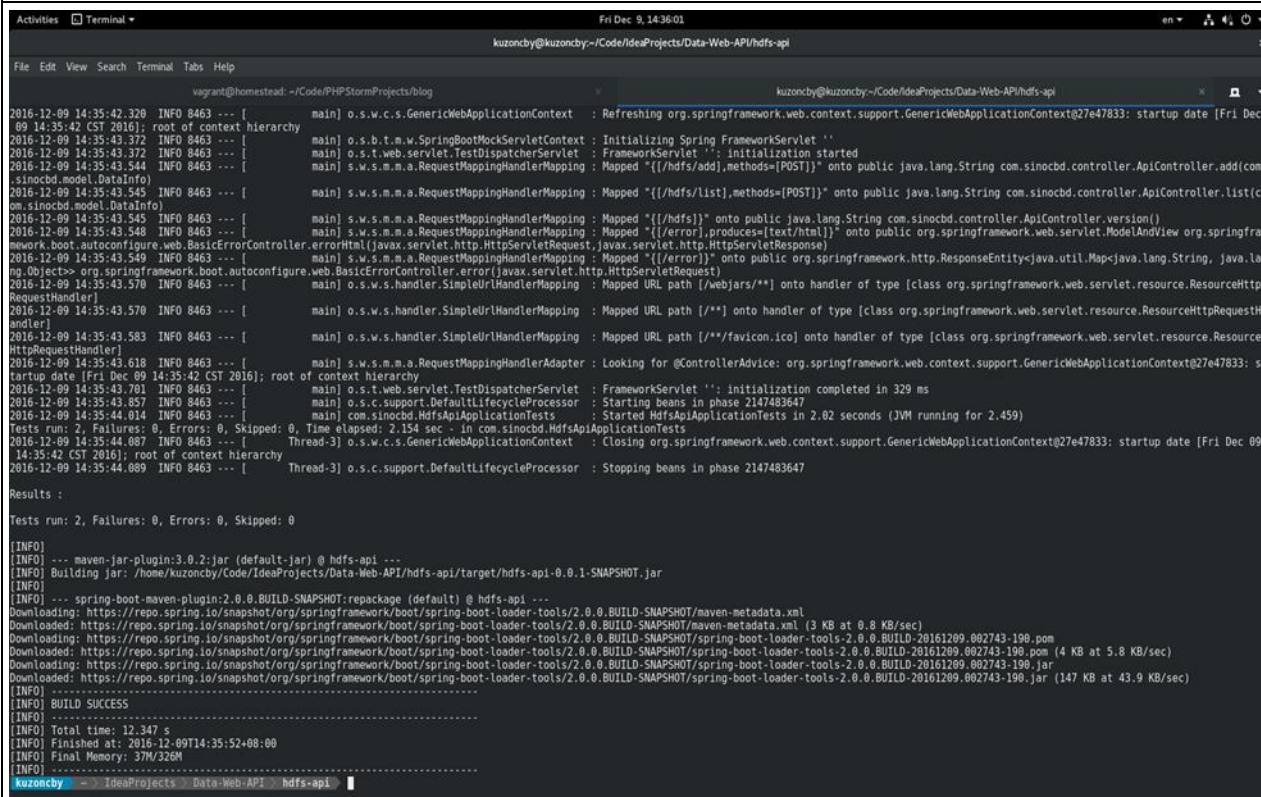
```

    </resources>
  </configuration>
</plugin>
</plugins>
</build>

```

将 docker.image.prefix 改为自己的镜像仓库地址。接下来执行：

```
$ mvn package
```



```

2016-12-09 14:35:42.326 INFO 8463 --- [main] o.s.w.c.s.GenericWebApplicationContext : Refreshing org.springframework.web.context.support.GenericWebApplicationContext@27e47833: startup date [Fri Dec 09 14:35:42 CST 2016]; root of context hierarchy
2016-12-09 14:35:43.372 INFO 8463 --- [main] o.s.b.t.m.w.SpringBootMockServletContext : Initializing Spring FrameworkServlet ''
2016-12-09 14:35:43.372 INFO 8463 --- [main] o.s.t.web.servlet.TestDispatcherServlet : FrameworkServlet '' : initialization started
2016-12-09 14:35:43.344 INFO 8463 --- [main] s.w.s.m.a.a.RequestMappingHandlerMapping : Mapped "{[/hdfs/add],methods=[POST]}" onto public java.lang.String com.sinocbd.controller.ApiController.add(com.sinocbd.model.DataInfo)
2016-12-09 14:35:43.345 INFO 8463 --- [main] s.w.s.m.a.a.RequestMappingHandlerMapping : Mapped "{[/hdfs/list],methods=[POST]}" onto public java.lang.String com.sinocbd.controller.ApiController.list(com.sinocbd.model.DataInfo)
2016-12-09 14:35:43.345 INFO 8463 --- [main] s.w.s.m.a.a.RequestMappingHandlerMapping : Mapped "{[/hdfs]}" onto public java.lang.String com.sinocbd.controller.ApiController.version()
2016-12-09 14:35:43.348 INFO 8463 --- [main] s.w.s.m.a.a.RequestMappingHandlerMapping : Mapped "{[/error],produces=[text/html]}" onto public org.springframework.web.servlet.ModelAndView org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter.handle(javax.servlet.http.HttpServletRequest, javax.servlet.http.HttpServletResponse)
2016-12-09 14:35:43.349 INFO 8463 --- [main] s.w.s.m.a.a.RequestMappingHandlerMapping : Mapped "{[/error]}" onto public org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter.handle(javax.servlet.http.HttpServletRequest, javax.servlet.http.HttpServletResponse)
2016-12-09 14:35:43.370 INFO 8463 --- [main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
2016-12-09 14:35:43.370 INFO 8463 --- [main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**/favicon.ico] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
2016-12-09 14:35:43.370 INFO 8463 --- [main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**/favicon.ico] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
2016-12-09 14:35:43.618 INFO 8463 --- [main] s.w.s.m.a.a.RequestMappingHandlerAdapter : Looking for @ControllerAdvice: org.springframework.web.context.support.GenericWebApplicationContext@27e47833: startup date [Fri Dec 09 14:35:42 CST 2016]; root of context hierarchy
2016-12-09 14:35:43.701 INFO 8463 --- [main] o.s.t.web.servlet.TestDispatcherServlet : FrameworkServlet '' : initialization completed in 329 ms
2016-12-09 14:35:43.857 INFO 8463 --- [main] o.s.c.support.DefaultLifecycleProcessor : Starting beans in phase 2147483647
2016-12-09 14:35:44.014 INFO 8463 --- [main] com.sinocbd.HdfsApiApplicationTests : Started HdfsApiApplicationTests in 2.02 seconds (JVM running for 2.459)
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 2.154 sec - in com.sinocbd.HdfsApiApplicationTests
2016-12-09 14:35:44.087 INFO 8463 --- [Thread-3] o.s.w.c.s.GenericWebApplicationContext : Closing org.springframework.web.context.support.GenericWebApplicationContext@27e47833: startup date [Fri Dec 09 14:35:42 CST 2016]; root of context hierarchy
2016-12-09 14:35:44.089 INFO 8463 --- [Thread-3] o.s.c.support.DefaultLifecycleProcessor : Stopping beans in phase 2147483647

Results :
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:3.0.2:jar (default-jar) @ hdfs-api ---
[INFO] Building jar: /home/kuzonchy/Code/IdeaProjects/Data-Web-API/hdfs-api/target/hdfs-api-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:2.0.0-BUILD-SNAPSHOT:repackage (default) @ hdfs-api ---
[INFO] Downloading: https://repo.spring.io/snapshot/org/springframework/boot/spring-boot-loader-tools/2.0.0-BUILD-SNAPSHOT/maven-metadata.xml
[INFO] Downloaded: https://repo.spring.io/snapshot/org/springframework/boot/spring-boot-loader-tools/2.0.0-BUILD-SNAPSHOT/maven-metadata.xml (3 KB at 0.8 KB/sec)
[INFO] Downloading: https://repo.spring.io/snapshot/org/springframework/boot/spring-boot-loader-tools/2.0.0-BUILD-SNAPSHOT/spring-boot-loader-tools-2.0.0-BUILD-SNAPSHOT-002743-190.pom
[INFO] Downloaded: https://repo.spring.io/snapshot/org/springframework/boot/spring-boot-loader-tools/2.0.0-BUILD-SNAPSHOT/spring-boot-loader-tools-2.0.0-BUILD-SNAPSHOT-002743-190.pom (4 KB at 5.8 KB/sec)
[INFO] Downloading: https://repo.spring.io/snapshot/org/springframework/boot/spring-boot-loader-tools/2.0.0-BUILD-SNAPSHOT/spring-boot-loader-tools-2.0.0-BUILD-SNAPSHOT-002743-190.jar
[INFO] Downloaded: https://repo.spring.io/snapshot/org/springframework/boot/spring-boot-loader-tools/2.0.0-BUILD-SNAPSHOT/spring-boot-loader-tools-2.0.0-BUILD-SNAPSHOT-002743-190.jar (147 KB at 43.9 KB/sec)
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 12.347 s
[INFO] Finished at: 2016-12-09T14:35:52+08:00
[INFO] Final Memory: 37M/320M
[INFO]

```

构建成功即可开始使用。

总线（ActiveMQ 消息队列）

这个程序可以起到服务之间的远程通讯，启动消息队列也非常容易。

进入到准备安装消息队列的机器输入创建一个 yamI 文件：

```

apiVersion: v1
kind: Service
metadata:
  name: activemq
labels:
  name: activemq
spec:
  type: NodePort

```

```
ports:
- port: 8161
  name: http
- port: 61616
  name: tcp
- port: 61613
  name: stomp
selector:
  name: activemq
```

```
starting node:
[kiosk@foundation0 ~]$ vim activemq.yaml
```

启动服务：

```
$ kubectl create -f activemq.yaml
```

```
[root@master ~]# kubectl create -o -
-f --filename=
[root@master ~]# kubectl create -f activemq.yaml
```

HDFS 操作 (Spring Data)

在启动这个服务之前，复制控制器要先启动，这样才能提供弹性计算服务。下面是复制控制器的 yaml 文件：

```
kind: ReplicationController
apiVersion: v1
metadata:
  name: hdfs-op-controller-latest
labels:
  name: hdfs-op
  version: latest
  visualize: "true"
spec:
  replicas: 2
  selector:
```



```
name: hdfs-op
version: latest
template:
  metadata:
    name: hdfs-op
  labels:
    name: hdfs-op
    version: latest
    visualize: "true"
    uses: "hdfs-op-service"
  spec:
    containers:
      - name: hdfs-op
        image: 192.168.1.50:5000/ hdfs-op
        imagePullPolicy: Always
        ports:
          - name: http
            containerPort: 9000
```

```
[root@master ~]# touch hdfs-op-replication-controller
[root@master ~]# vim hdfs-op-replication-controller
[root@master ~]# mv hdfs-op-replication-controller hdfs-op-replication-controller
[root@master ~]# kubectl create -f hdfs-op-replication-controller.yaml
```

接下来才能启动 Hadoop 操作的这项服务：

```
kind: Service
apiVersion: v1
metadata:
  name: hdfs-op
  labels:
    name: hdfs-op
    visualize: "true"
spec:
```

```
type: LoadBalancer
```

```
ports:
```

```
  - port: 9000
```

```
    targetPort: http
```

```
selector:
```

```
  name: hdfs-op
```

```
[root@master ~]# kubectl create -f hdfs-op.yaml
```

网关 (API Gateway)

在启动这个服务之前，和刚才一样复制控制器要先启动，这样才能提供弹性计算服务。下面是复制控制器的yaml文件：

```
kind: ReplicationController
```

```
apiVersion: v1
```

```
metadata:
```

```
  name: hdfs-api-controller-latest
```

```
  labels:
```

```
    name: hdfs-api
```

```
    version: latest
```

```
    visualize: "true"
```

```
spec:
```

```
  replicas: 2
```

```
  selector:
```

```
    name: hdfs-api
```

```
    version: latest
```

```
  template:
```

```
    metadata:
```

```
      name: hdfs-api
```

```
      labels:
```

```
        name: hdfs-api
```

```
        version: latest
```

```
        visualize: "true"
```

```
        uses: "hdfs-api-service"
```

```
spec:
  containers:
  - name: hdfs-api
    image: 192.168.1.50:5000/ hdfs-api
    imagePullPolicy: Always
    ports:
    - name: http
      containerPort: 9000
```

```
[root@master ~]# vim hdfs-api.yaml
```

接下来才能启动 REST（Spring Web MVC）这项服务：

```
kind: Service
apiVersion: v1
metadata:
  name: hdfs-api
  labels:
    name: hdfs-api
    visualize: "true"
spec:
  type: LoadBalancer
  ports:
  - port: 9000
    targetPort: http
  selector:
    name: hdfs-api
```

```
[root@master ~]# kubectl create -f hdfs-api.yaml
```

故障排查

Maven 无法编译：使用 aliyun 的仓库

池无法创建：检查仓库状态

服务无法启动：请先启动总线（ActiveMQ）再启动别的服务

计划中的更新

Kubernetes 加入持久化链

附加说明

Kubernetes 的简单安装方法

先要做一些准备：

- Windows 用户将需要有管理员权限，并需要在 Windows 10 或 Windows 8 上启用 Hyper-V。
- OS X 用户将需要安装 xhyve 驱动程序。
- Linux 将需要安装 kvm 驱动程序

接下来为您的平台下载 [gofabric8](#) 二进制包，并将其添加到您的 PATH. 接下来输入一个命令开启 kubernetes:

```
$ gofabric8 start
```