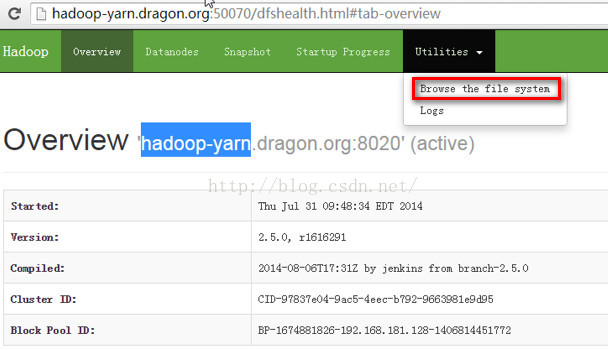
HDFS HA监控WEB UI页面分析  
和查看NN与JN存储的编辑日志

到现在为止，我们已经配置了hadoop的HA，让我们通过页面去查看下hadoop的文件系统。

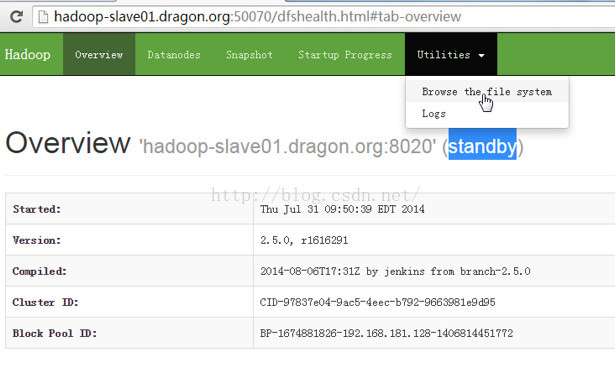
# 1. 分析active namenode和standby namenode对客户端服务的情况。



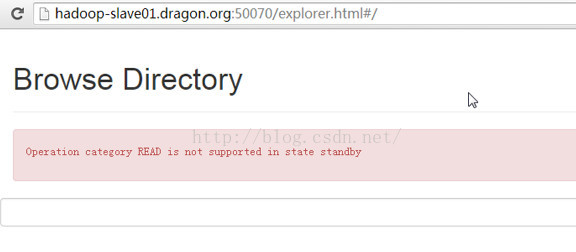
我们可以清楚看到hadoop文件系统的目录结构：



以上我们都是通过active的namenode访问hadoop的，那么如果我们通过standby namenode可不可以访问hadoop呢？



接下来我们看到，通过standby namenode是无法访问hadoop的文件系统的。根据提示，我们知道，standby namenode是不支持读操作的。



这也就证明了，在HA机制中，standby namenode是不能给客户端提供服务的。

# 2. 分析active namenode和standby namenode的datanode的情况。

在我们的HA配置中，总共有3个datanode，这3个datanode同时为active namenode和standby namenode服务。

## 第一：我们首先分析下standby namenode下的datanode个数情况以及standby namenode是从哪里读取编辑日志的。



通过上图我们知道，standby namenode下面有3个datanode, 并且standby namenode是通过QJM的方式读取编辑日志的。

## 第二，我们来查看一下active namenode是如何向三个journal node写编辑日志的。



我们可以看到当前的active namenode正在向3个journal node写编辑日志，当前的编辑日志的transaction id 是 41。

## 第三，我们来对比下active namenode和standby namenode的cluster ID，必定是相同的。





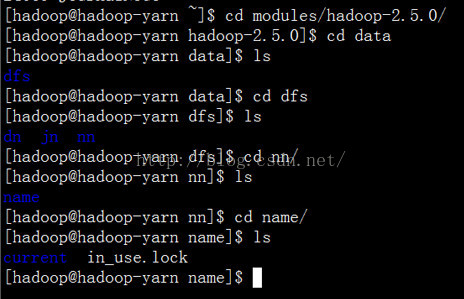
## 第四：对比Block ID，我们知道Block ID存储的是hadoop分布式文件系统的元信息，因为standby namenode是active namenode的备份节点，

他们管理的元信息应该是一样的，所以他们的Block ID应该也是相同的。



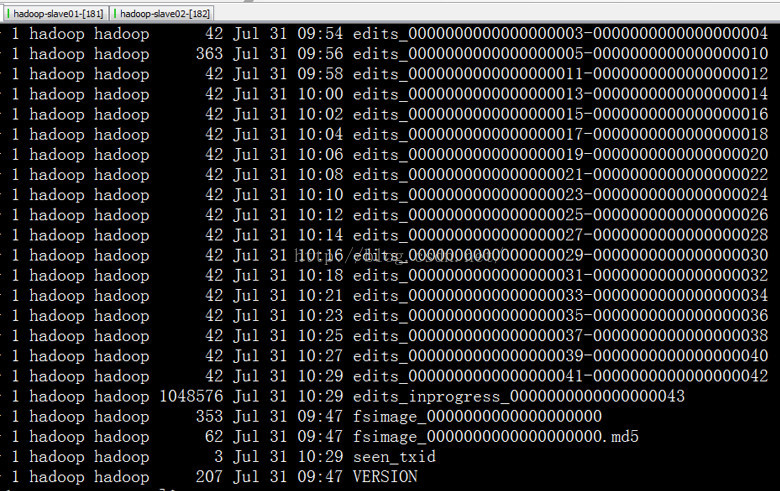


## 第五：现在我们去active namenode的存储目录中去看下。



有一个in\_user.lock的文件，证明当前这个文件正在被使用。current 目录下存放的是当前的active namenode下存放的映像文件和编辑日志文件。

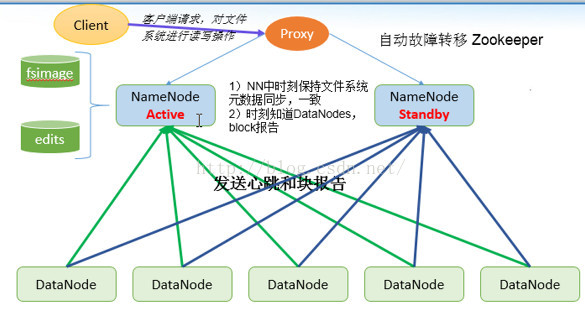
我们查看下current 目录下的文件：



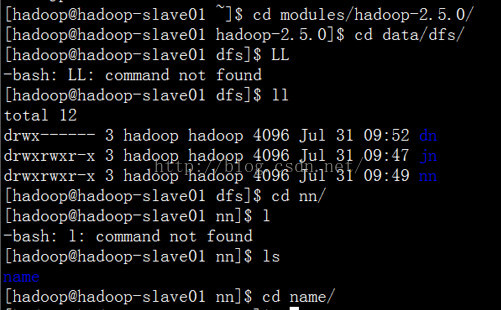
有人肯定会有疑问，我们的编辑日志文件不是存放在journal node里面吗？为什么active namenode里面也会有编辑日志文件呢？

让我们来看下面的一张图，来解释下。我们发现，编辑日志文件在active namenode端始终是有的。当客户端请求namenode的时候，

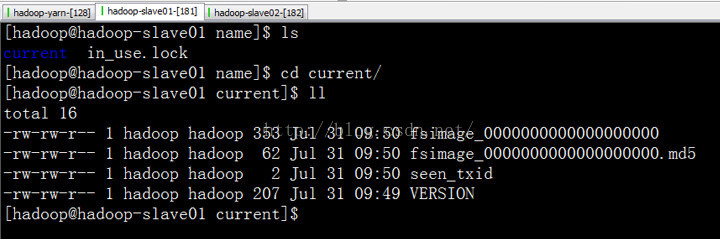
namenode不仅将编辑日志写到本地，还会将编辑日志文件写QJM的节点中。



## 第六：现在我们去standby namenode的存储目录中去看下。



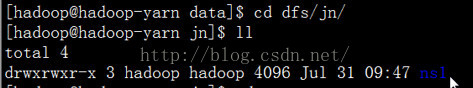
进入到current目录下面：



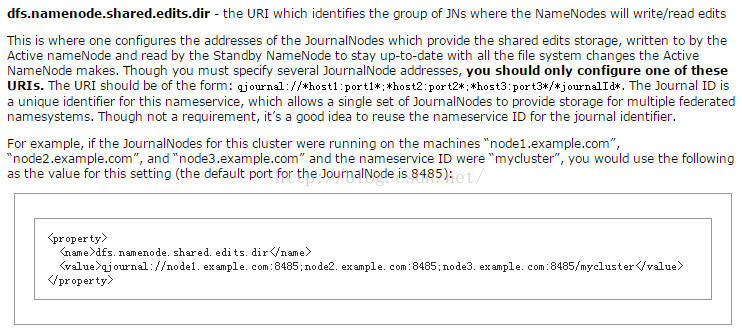
我们发现，这里只有映像文件，没有编辑日志文件。这里的映像文件，我们在启动的时候，使用的一个命令，将active namenode的映像文件完全复制过来的。

http://img.blog.csdn.net/20160925152833364?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQv/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center

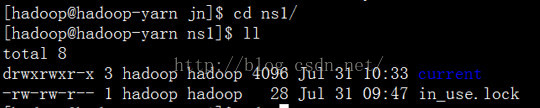
## 第七：现在我们去journal node为当前的namespace(ns1)创建的存储目录中去看下。



至于为什么要这么配置，官方文档有一个很好的说明：



我们先进去看下这个目录下存放的是什么？里面存放的应该是编辑日志。



首先是current 文件夹，然后current下面存放的是编辑日志。

