**一、elasticsearch所在主机安装**repository-hdfs插件**：**

./bin/elasticsearch-plugin install repository-hdfs

修改配置

修改elasticsearch的config路径下的config/jvm.options文件

增加如下配置：

-Djava.security.policy=/opt/elasticsearch-5.6.0/plugins/repository-hdfs/plugin-security.policy  
注：请将/opt/elasticsearch-5.6.0/plugins/repository-hdfs/plugin-security.policy替换为repository-hdfs插件路径下plugin-security.policy的真实路径。

修改java安全策略

vim /opt/elasticsearch-5.6.0/plugins/repository-hdfs/plugin-security.policy

替换内容如下：

grant {

// Hadoop UserGroupInformation, HdfsConstants, PipelineAck clinit

permission java.lang.RuntimePermission "getClassLoader";

// UserGroupInformation (UGI) Metrics clinit

permission java.lang.RuntimePermission "accessDeclaredMembers";

permission java.lang.reflect.ReflectPermission "suppressAccessChecks";

// org.apache.hadoop.util.StringUtils clinit

permission java.util.PropertyPermission "\*", "read,write";

// org.apache.hadoop.util.ShutdownHookManager clinit

permission java.lang.RuntimePermission "shutdownHooks";

// JAAS is used always, we use a fake subject, hurts nobody

permission javax.security.auth.AuthPermission "getSubject";

permission javax.security.auth.AuthPermission "doAs";

permission javax.security.auth.AuthPermission "modifyPrivateCredentials";

permission java.lang.RuntimePermission "accessDeclaredMembers";

permission java.lang.RuntimePermission "getClassLoader";

permission java.lang.RuntimePermission "shutdownHooks";

permission java.lang.reflect.ReflectPermission "suppressAccessChecks";

permission javax.security.auth.AuthPermission "doAs";

permission javax.security.auth.AuthPermission "getSubject";

permission javax.security.auth.AuthPermission "modifyPrivateCredentials";

permission java.security.AllPermission;

permission java.util.PropertyPermission "\*", "read,write";

permission javax.security.auth.PrivateCredentialPermission "org.apache.hadoop.security.Credentials \* \"\*\"", "read";

};

重启elasticsearch服务

**配置使用：**

curl -XPUT 'http://10.10.100.99:9200/\_snapshot/backup' -d '{

"type": "hdfs",

 "settings": {

"uri": "hdfs://l172.17.0.3:9000",

"path": "/data/es",

"conf\_location": "/usr/local/hadoop/etc/hadoop/hdfs-site.xml, /usr/local/hadoop/etc/hadoop/core-site.xml "

}

}

'

关于配置：

uri                 :    hdfs的uri在 /usr/local/hadoop/etc/hadoop/core-site.xml 下可以看到

path              :    数据存储/加载的路径

load\_defaults:    是否加载hadoop默认配置（默认开启）

config.<key>:    是否要将内联配置参数加载到hadoop（可选）

compress      :    是否压缩元数据（默认关闭）

chunk\_size    :    覆盖块大小（默认关闭）

security\_principle:连接到hadoop时是否使用kerberos

备份快照：

curl -XPUT 'http://localhost:9200/\_snapshot/backup/snapshot\_logstash-2017.10.26' -d '{"indices":"logstash-2017.10.26"}'

查看备份状态：

curl -XGET 'http://localhost:9200/\_snapshot/backup/snapshot\_logstash-2017.10.26?pretty'

恢复快照：

要先关闭所有的索引才能恢复

curl -XPOST localhost:9200/\_all/\_close

恢复

url -XPOST localhost:9200/\_snapshot/backup/ snapshot\_logstash-2017.10.26/\_restore

原本想把索引全部导入HDFS后做分析的，不过这种方法只能建立一个快照，数据没导过去

暂时还没找到能将索引导入HDFS的好方法，可能要重新设计才行了> <