Copying Data Between Two Clusters Using Distcp (#topic_7_2)

The Distcp Command (#topic_7_2_1)

The distributed copy command, distcp (http://hadoop.apache.org/docs/current/hadoop-distcp/DistCp.html) , is a general utility for copying large data sets between distributed filesystems within and across clusters. The `distcp` command submits a regular MapReduce job that performs a file-by-file copy.

To see the `distcp` command options, run the built-in help:

```
$ hadoop distcp
```

Important:

- Do not run `distcp` as the `hdfs` user which is blacklisted for MapReduce jobs by default.
- Do not use **Hadoop shell commands (http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/FileSystemShell.html)** (such as `cp`, `copyfromlocal`, `put`, `get`) for large copying jobs or you may experience I/O bottlenecks.

Distcp Syntax and Examples (#distcp_examples)

You can use `distcp` to copy files between compatible clusters in either direction, from or to the source or destination clusters.

For example, when upgrading, say from CDH 4 to CDH 5, you should run `distcp` *from* the CDH 5 cluster in this manner:

```
$ hadoop distcp hftp://cdh4-namenode:50070/ hdfs://CDH5-nameservice/
$ hadoop distcp s3a://bucket/ hdfs://CDH5-nameservice/
```

You can also use a specific path, such as `/hbase` to move HBase data, for example:

```
$ hadoop distcp hftp://cdh4-namenode:50070/hbase hdfs://CDH5-nameservice/hbase
$ hadoop distcp s3a://bucket/file hdfs://CDH5-nameservice/bucket/file
```

HFTP Protocol (#distcp_and_hftp)

The HFTP protocol allows you to use FTP resources in an HTTP request. When copying with `distcp` across *different versions of CDH*, use `hftp://` for the source filesystem and `hdfs://` for the destination filesystem, and run `distcp` from the destination cluster. The default port for HFTP is `50070` and the default port for HDFS is `8020`.

Example of a source URI: `hftp://namenode-location:50070/basePath`

- `hftp://` is the source protocol.
- `namenode-location` is the CDH 4 (source) NameNode hostname as defined by its configured `fs.default.name`.
- `50070` is the NameNode's HTTP server port, as defined by the configured `dfs.http.address`.

Example of a destination URI: `hdfs://nameservice-id/basePath` or `hdfs://namenode-location`

- `hdfs://` is the destination protocol
- `nameservice-id` or `namenode-location` is the CDH 5 (destination) NameNode hostname as defined by its configured `fs.defaultFS`.
- `basePath` in both examples refers to the directory you want to copy, if one is specifically needed.

Important:

- HFTP is a read-only protocol and can only be used for the source cluster, not the destination.
- HFTP cannot be used when copying with `distcp` from an insecure cluster to a secure cluster.

S3 Protocol (#distcp_and_s3)

Amazon S3 block and native filesystems are also supported with the `s3a://` protocol.

Example of an Amazon S3 Block Filesystem URI: `s3a://`*`bucket_name`*`/path/to/file`

S3 credentials can be provided in a configuration file (for example, core-site.xml):

```
<property>
    <name>fs.s3a.access.key</name>
    <value>...</value>
</property>
<property>
    <name>fs.s3a.secret.key</name>
    <value>...</value>
</property>
```

or run on the command line as follows:

```
hadoop distcp -Dfs.s3a.access.key=... -Dfs.s3a.secret.key=... s3a://
```

Kerberos Setup Guidelines for Distcp between Secure Clusters (without Cross-realm Authentication) (#concept_fx2_t1q_3x)

The guidelines mentioned in this section are only applicable for the following sample deployment:

- Let's assume you have two clusters with the realms: `SOURCE` and `DESTINATION`
- You have data that needs to be copied from `SOURCE` to `DESTINATION`
- Trust exists between `SOURCE` and Active Directory, and `DESTINATION` and Active Directory.
- Both `SOURCE` and `DESTINATION` clusters are running CDH 5.3.4 or higher

If your environment matches the one described above, use the following table to configure Kerberos delegation tokens on your cluster so that you can successfully `distcp` across two secure clusters. Based on the direction of the trust between the `SOURCE` and `DESTINATION` clusters, you can use the `mapreduce.job.hdfs-servers.token-renewal.exclude` property to instruct ResourceManagers on either cluster to skip or perform delegation token renewal for NameNode hosts.

Environment Type	Kerberos Delegation Token Setting

SOURCE trusts DESTINATION	Distcp job runs on the DESTINATION cluster	You do not need to set the mapreduce.job.hd servers.token-renewal.exclude property.
	Distcp job runs on the SOURCE cluster	Set the mapreduce.job.hdfs-servers.toke renewal.exclude property to a comma-separ hostnames of the NameNodes of the DESTINA
DESTINATION trusts SOURCE	Distcp job runs on the DESTINATION cluster	Set the mapreduce.job.hdfs-servers.toke renewal.exclude property to a comma-separ hostnames of the NameNodes of the SOURCE c
	Distcp job runs on the SOURCE cluster	You do not need to set the mapreduce.job.hd servers.token-renewal.exclude property.
Both SOURCE and DESTINATION trust each other	You do not need to set the mapreduce.job.hdfs-servers.token-renewal.exclude propert	
Neither SOURCE nor DESTINATION trusts the other	If a common realm is usable (such as Active Directory), set the mapreduce.job.hdfs-server renewal.exclude property to a comma-separated list of hostnames of the NameNodes of the *not* running the distcp job. For example, if you are running the job on the DESTINATION cluste 1. kinit on any DESTINATION YARN Gateway host using an AD account that can be used on bo DESTINATION. 2. Run the distcp job as the hadoop user: `$ hadoop distcp -Ddfs.namenode.kerberos.principal.pattern=* \ -Dmapreduce.job.hdfs-servers.token-renewal.exclude=SOURCE-nn-host1,SOURCE- hdfs://source-nn-nameservice/source/path \ /destination/path`	

Distcp between Secure Clusters in Distinct Kerberos Realms (#concept_hcs_srr_sr)

Note: JDK version 1.7.x is required on both clusters when copying data between Kerberized clusters that are in different realms. For information about supported JDK versions, see **Supported JDK Versions (cdh_ig_req_supported_versions.html#concept_pdd_kzf_vp)**.

Specify the Destination Parameters in krb5.conf (#concept_txx_dtr_sr)

Edit the krb5.conf file on the client (where the distcp job will be submitted) to include the destination hostname and realm.

```
[realms]
HADOOP.QA.domain.COM = { kdc = kdc.domain.com:88 admin_server = admin.test.com:749
default_domain = domain.com supported_enctypes = arcfour-hmac:normal des-cbc-crc:normal
des-cbc-md5:normal des:normal des:v4 des:norealm des:onlyrealm des:afs3 }

[domain_realm]
.domain.com = HADOOP.test.domain.COM
domain.com = HADOOP.test.domain.COM
test03.domain.com = HADOOP.QA.domain.COM
```

Configure HDFS RPC Protection and Acceptable Kerberos Principal Patterns (#concept_vwg_wnj_55)

Set the `hadoop.rpc.protection` property to `authentication` in both clusters. You can modify this property either in `hdfs-site.xml`, or using Cloudera Manager as follows:

1. Open the Cloudera Manager Admin Console.
2. Go to the HDFS service.
3. Click the Configuration tab.
4. Select Scope > HDFS-1 (Service-Wide)
5. Select Category > Security.
6. Locate the **Hadoop RPC Protection** property and select `authentication`.
7. Click Save Changes to commit the changes.

The following steps are not required if the two realms are already set up to trust each other (cm_bdr_replication_and_kerberos.html) , or have the same principal pattern. However, this isn't usually the case.

Set the `dfs.namenode.kerberos.principal.pattern` property to `*` to allow distcp irrespective of the principal patterns of the source and destination clusters. You can modify this property either in `hdfs-site.xml` on both clusters, or using Cloudera Manager as follows:

1. Open the Cloudera Manager Admin Console.
2. Go to the HDFS service.
3. Click the Configuration tab.
4. Select Scope > Gateway
5. Select Category > Advanced.
6. Edit the **HDFS Client Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml** property to add:

```
<property>
  <name>dfs.namenode.kerberos.principal.pattern</name>
  <value>*</value>
</property>
```

7. Click Save Changes to commit the changes.

(If TLS/SSL is enabled) Specify Truststore Properties (#concept_o5v_dtr_sr)

The following properties must be configured in the `ssl-client.xml` file on the client submitting the distcp job to establish trust between the target and destination clusters.

```
<property>
<name>ssl.client.truststore.location</name>
<value>path_to_truststore</value>
</property>

<property>
<name>ssl.client.truststore.password</name>
```

```
<value>XXXXXX</value>
</property>

<property>
<name>ssl.client.truststore.type</name>
<value>jks</value>
</property>
```

Set HADOOP_CONF to the Destination Cluster (#concept_ixz_h5r_sr)

Set the HADOOP_CONF path to be the destination environment. If you are not using HFTP, set the HADOOP_CONF path to the source environment instead.

Launch Distcp (#concept_vgk_pxs_sr)

Kinit on the client and launch the distcp job.

```
hadoop distcp hdfs://test01.domain.com:8020/user/alice hdfs://test02.domain.com:8020/user/alice
```

If launching distcp fails, force Kerberos to use TCP instead of UDP by adding the following parameter to the krb5.conf file on the client.

```
[libdefaults]
udp_preference_limit = 1
```

Enabling Fallback Configuration (#concept_whd_kb2_5v)

To enable the fallback configuration, for copying between secure and insecure clusters, add the following to the HDFS configuration file, core-default.xml, by using an advanced configuration snippet if you use Cloudera Manager, or editing the file directly otherwise.

```
<property>
  <name>ipc.client.fallback-to-simple-auth-allowed</name>
  <value>true</value>
</property>
```

Protocol Support for Distcp (#concept_hfx_tqr_rp)

The following table lists the protocols supported with the distcp command on different versions of CDH. "Secure" means that the cluster is configured to use Kerberos.

Note: Copying between a secure cluster and an insecure cluster is only supported with CDH 5.1.3 and higher (CDH 5.1.3+) in accordance with **HDFS-6776 (https://issues.apache.org/jira/browse/HDFS-6776)** .

Source	Destination	Where to Issue distcp Command	Source Protocol	Source Config	Destination Protocol	Destination Config	Fallback Config (cdh_admin_distcp_ Required
CDH 4	CDH 4	Destination	hftp	Secure	hdfs or webhdfs	Secure	

Source	Destination	Where to Issue distcp Command	Source Protocol	Source Config	Destination Protocol	Destination Config	Fallback Config (cdh_admin_distcp_ Required
CDH 4	CDH 4	Source or Destination	hdfs or webhdfs	Secure	hdfs or webhdfs	Secure	
CDH 4	CDH 4	Source or Destination	hdfs or webhdfs	Insecure	hdfs or webhdfs	Insecure	
CDH 4	CDH 4	Destination	hftp	Insecure	hdfs or webhdfs	Insecure	
CDH 4	CDH 5	Destination	webhdfs or hftp	Secure	webhdfs or hdfs	Secure	
CDH 4	CDH 5.1.3+	Destination	webhdfs	Insecure	webhdfs	Secure	Yes
CDH 4	CDH 5	Destination	webhdfs or hftp	Insecure	webhdfs or hdfs	Insecure	
CDH 4	CDH 5	Source	hdfs or webhdfs	Insecure	webhdfs	Insecure	
CDH 5	CDH 4	Source or Destination	webhdfs	Secure	webhdfs	Secure	
CDH 5	CDH 4	Source	hdfs	Secure	webhdfs	Secure	
CDH 5.1.3+	CDH 4	Source	hdfs or webhdfs	Secure	webhdfs	Insecure	Yes
CDH 5	CDH 4	Source or Destination	webhdfs	Insecure	webhdfs	Insecure	
CDH 5	CDH 4	Destination	webhdfs	Insecure	hdfs	Insecure	
CDH 5	CDH 4	Source	hdfs	Insecure	webhdfs	Insecure	
CDH 5	CDH 4	Destination	hftp	Insecure	hdfs or webhdfs	Insecure	
CDH 5	CDH 5	Source or Destination	hdfs or webhdfs	Secure	hdfs or webhdfs	Secure	
CDH 5	CDH 5	Destination	hftp	Secure	hdfs or webhdfs	Secure	
CDH 5.1.3+	CDH 5	Source	hdfs or webhdfs	Secure	hdfs or webhdfs	Insecure	Yes
CDH 5	CDH 5.1.3+	Destination	hdfs or webhdfs	Insecure	hdfs or webhdfs	Secure	Yes
CDH 5	CDH 5	Source or Destination	hdfs or webhdfs	Insecure	hdfs or webhdfs	Insecure	

Source	Destination	Where to Issue distcp Command	Source Protocol	Source Config	Destination Protocol	Destination Config	Fallback Config (cdh_admin_distcp_ Required
CDH 5	CDH 5	Destination	hftp	Insecure	hdfs or webhdfs	Insecure	

Categories: Administrators (../categories/hub_administrators.html) | Clusters (../categories/hub_clusters.html) | Data Analysts (../categories/hub_data_analysts.html) | Disk Storage (../categories/hub_disk_storage.html) | ETL (../categories/hub_etl.html) | Ingest (../categories/hub_ingest.html) | Kerberos (../categories/hub_kerberos.html) | Migrating (../categories/hub_migrating.html) | Security (../categories/hub_security.html) | All Categories (../categories/hub.html)

Page generated December 2, 2016.