

1. System Administration Guide	2
1.1 Physical Architecture	2
1.1.1 Servers	3
1.1.2 Assumptions	3
1.1.3 API and Endpoint Exposure	4
1.1.4 Shared File Storage	4
1.2 Installation	4
1.2.1 Archive	5
1.2.2 Index	6
1.2.3 DC Services	7
1.2.4 User Interface	11
1.3 Exemplar Configuration - Johns Hopkins Data Management Service	11
1.3.1 Server Specification	11
1.3.2 Components	12
1.4 Backup & Recovery	12
1.4.1 Fedora	12
1.4.2 Solr	13
1.4.3 DCS Database	13

System Administration Guide

The System Administration Guide is targeted at systems administrators responsible for installing the DCS.

Physical Architecture

- Servers
- Assumptions
- API and Endpoint Exposure
- Shared File Storage

Installation

- Archive
- Index
- DC Services
- User Interface

Exemplar Configuration - Johns Hopkins Data Management Service

- Server Specification
- Components

Backup & Recovery

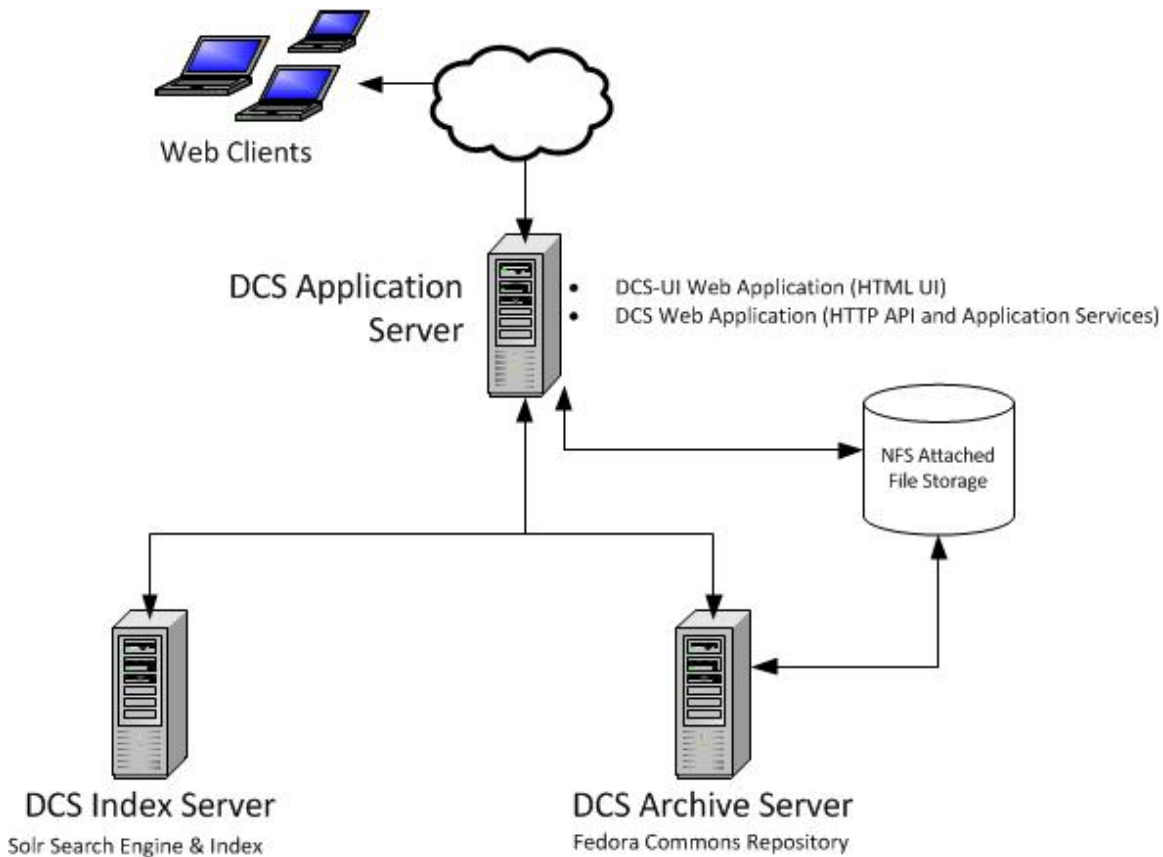
- Fedora
- Solr
- DCS Database

Physical Architecture

- Servers
- Assumptions
- API and Endpoint Exposure
- Shared File Storage

This section and sub-sections describe the physical architecture of a generic DCS system deployment including the systems involved, their roles, and how they interact (i.e. entities can only be retrieved from the archive via DCS APIs if the index is current and up-to-date; the `dcservice` machine and the `dcarchive` machine must mount a shared storage pool). The documentation includes how storage is shared between the archive and `dcservice` machine.

The following diagram is used to visualize the physical layout of the DCS system.



Servers

Assumptions

API and Endpoint Exposure

Shared File Storage

Servers

A typical Data Conservancy (DC) system deployment consists of 3 servers (virtual or physical). These systems, taken together, are known as a "Data Conservancy Instance" (or simply, an "Instance"):

1. **DC Services Server (a.k.a. "dcservice", "DC Service")**
 - a. Provides the runtime environment for the DC web APIs and applications.
 - b. Receives all HTTP requests for data inputs or outputs.
 - c. Communicates with the DC Archive and Index servers as needed.
2. **DC Archive Server (a.k.a. "dcarchive", "DC Archive")**
 - a. Archival content
 - b. Runs an instance of Fedora Commons Repository server with locally or Network File System (NFS) attached remote content storage.
3. **DC Index Server (a.k.a. "dcindex", "DC Index")**
 - a. Search engine and index server running Solr search engine.
 - b. Supports the DC query API, indexes content in the DC Archive.

The Johns Hopkins Data Management instance of the Data Conservancy system provides an example of the server specifications and components ([Exemplar Configuration - Johns Hopkins Data Management Service](#)).

Assumptions

The DC Instance makes assumptions about its runtime environment:

1. The dcservice server “owns” the dcindex and dcarchive servers, their services, and their data. This means that the Fedora and Solr instances cannot be shared with other applications; the dcservice server requires its own instance of each*.
2. The dcservice server requires *exclusive access* to the dcindex and dcarchive applications. This means, for example, that system administrators or non-DC applications should not be modifying data in the Fedora archive, except through APIs exposed by the dcservice server.
3. The dcarchive and daccess server require access to a file system *shared* between the two systems (more on that below). It is assumed that the DC will manage the data that resides under that file system, and it should remain opaque to system administrators.
4. That the server environment is a *Linux variant*. DC developers use multiple environments, including Mac, Windows, and Linux. However, our production builds and tests are executed in a Linux environment. Running an Instance on the Windows or Mac platform isn’t recommended, but may be possible.

* Advanced users of Fedora and Solr may realize configurations to share dcindex and dcarchive installations between Data Conservancy and other services (e.g. Hydra, Fez, etc.). This is discouraged, because future versions of the Data Conservancy will operate under the aforementioned assumptions, and undefined behavior (e.g. data loss) may result.

API and Endpoint Exposure

User agents interact with a Data Conservancy Instance through the public DCS APIs, which are exposed by the dcservice server. APIs hosted by dcarchive and dcindex are presumed to be private, and not exposed to user agents of the Instance. System administrators should take care to restrict access to the Fedora and Solr API endpoints to **only** the servers that make up the Instance (for example, the Fedora APIs should be accessible to the dcservice and dcindex servers, while the Solr interface should be accessible to the dcservice and dcarchive servers). Fedora and Solr APIs should **not** be publicly available.

Shared File Storage

The DC Services and DC Archive Server must share a common storage pool for content files. Commonly, the Network File System (NFS) will be used to share the storage, but alternate technologies may be employed, as long as each server references the common storage pool using the same mount path.

The common file storage location is specified in the dcs configuration files under the property name dcs.home (see DCS Services Installation documentation on [DC Services Configuration](#)). For example `dcs.home = /mnt/storage/dcs`

Within the common file storage location DCS creates two sub-directories: "files" and "staging". The "files" directory is used as the common storage pool location between DC Services and DC Archive. The dcs services application places content files in the "files" directory when they are uploaded as part of a deliverable unit. The associated metadata and package information included in the deliverable unit are placed in the "staging" directory.

Once the deliverable unit is processed through the dcs ingest workflows the deposited content is persisted by Fedora on the DC Archive Server which stores a URL to the content files stored in the Shared Storage Pool.

```
.
|-- files (shared by dcservices and dcarchive)
|   |-- 02
|   |-- 06
|   |-- ...
|   `-- f6
`-- staging (used by dcservices)
    |-- metadata
    `-- sips
```

Installation

The following pages provide information on how to install the major components of the DCS system. Before carrying out the installation you should read the documentation and subsections on the [Physical Architecture](#) of the system so you have an understanding of how and where the components should be installed.

Archive

Index

DC Services

User Interface

Archive

- 1. Install
- 2. Configure
- 3. Start & Stop
- 4. Verify
- 5. Re-initialize

1. Install

Install Fedora 3.5 server using the [Fedora install documentation](#)

The DC Service and DC Index servers require access to Fedora's REST interface. Therefore when installing Fedora note any passwords required to connect to the Fedora APIs or if SSL is required to connect to the APIs.

2. Configure

Configure Fedora using the [Fedora configuration documentation](#).

The Data Conservancy does not currently rely on any of Fedora's features (such as the Resource Index, or FeSL), nor does it rely on any of the Fedora-supplied local services or demos. So these do not need to be installed or enabled.

3. Start & Stop

Assuming you have installed Fedora per the Fedora installation instructions above then login using the account used to install Fedora. Start Fedora by starting Tomcat:

```
$FEDORA_HOME/tomcat/bin/startup.sh
```

If you selected the custom install option, ensure that your database server is running.

Likewise to stop Fedora stop Tomcat:

```
$FEDORA_HOME/tomcat/bin/shutdown.sh
```

4. Verify

To verify that Fedora is up and running start Fedora and connect to Fedora's /describe URL. For example if Fedora is installed on dcarchive.dataconservancy.org using Tomcat port 8980 then connect to <http://dcarchive.dataconservancy.org:8980/fedora/describe>

You should see something similar to below, indicating that Fedora is up and running



The dcservice and dcindex servers need to be able to access Fedora, so you should login to each server, and use 'curl' (or similar) to insure that they can talk to Fedora by retrieving its /describe URL.

Remember that no other servers or programs should be accessing Fedora except for the dcservice and dcindex servers. The dcservice server assumes it has exclusive access to the Fedora repository.

5. Re-initialize

This describes how to “clean out” your Fedora archive; these steps will erase all data in your Fedora archive! Typically this step will be followed by re-initializing the Solr index (on the dcindex server) as well, because the Solr index will contain references to archival objects that no longer exist. You should be logged in as root or the account used to install Fedora.

Stop Fedora as described above.

Delete the files in the objectStore and resourceIndex directories in the Fedora data directory. For example if you installed Fedora under /usr/local/fcrepo :

1. `rm -rf /usr/local/fcrepo/data/objectStore/*`
2. `rm -rf /usr/local/fcrepo/data/resourceIndex/*`

Run the fedora-rebuild command line utility to rebuild the SQL database and Resource Index (for additional information see the [Fedora Server Command Line Utilities](#) documentation).

1. `fedora-rebuild.sh` (rebuild the SQL database first)
 2. `fedora-rebuild.sh` (then rebuild the Resource Index)
- Then restart the Fedora Tomcat server.

Index

- [1. Install](#)
- [2. Start & Stop](#)
- [3. Verify](#)
- [4. Re-initialize](#)

1. Install

To install Solr:

1. Download the Solr 3.6.1 distribution, extract it, and install the Solr distribution war to your preferred servlet container's webapps directory.
2. Determine your Solr home directory, \$SOLR_HOME.
3. Define your \$SOLR_HOME per the [Solr install documentation](#) and your preferred servlet container documentation. Solr provides detailed instructions for installing Solr under several different application containers including [installing Solr within Tomcat](#). You will either need to define the system property `solr.solr.home`, or define it using JNDI. For example, to start Jetty defining the `solr.solr.home` system property: `java -jar -Dsolr.solr.home=$SOLR_HOME start.jar`
4. Extract the DCS Solr configuration to \$SOLR_HOME:
 - a. `curl 'http://maven.dataconservancy.org/public/snapshots/org/dataconservancy/index/dcs-index-dcp-solr-resources/1.1.0-SNAPSHOT/' > $SOLR_HOME/dcs-index-dcp-solr-resources-1.1.0-SNAPSHOT.jar`
 - b. `jar -xf dcs-index-dcp-solr-resources-1.1.0-SNAPSHOT.jar org/dataconservancy/dcs/index/dcpsolr/`
 - c. `mv org/dataconservancy/dcs/index/dcpsolr/* .`
 - d. `rm -rf org/ support/`
5. Your \$SOLR_HOME directory structure should look like this:

```

$SOLR_HOME/
|-- default
|   |-- conf
|       |-- elevate.xml
|       |-- mapping-ISOLatin1Accent.txt
|       |-- protwords.txt
|       |-- schema.xml
|       |-- scripts.conf
|       |-- solrconfig.xml
|       |-- spellings.txt
|       |-- stopwords.txt
|       |-- synonyms.txt
|-- solr.xml

```

For more information on how to configure Solr please see the [Configuring Solr documentation](#).

2. Start & Stop

Start and stop Solr by starting or stopping the application server Solr is running in. **Note that you need to configure the `solr.solr.home` system property appropriately for your servlet container for these commands to work!**

For example if you are using Tomcat then:

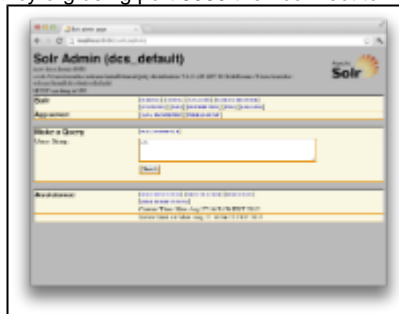
- `$TOMCAT_HOME/tomcat/bin/startup.sh` (to start Solr)
- `$TOMCAT_HOME/tomcat/bin/shutdown.sh` (to stop Solr)

If you are you using Jetty, then:

- `java -jar $JETTY_HOME/start.jar`
- `java -jar $JETTY_HOME/start.jar --stop`

3. Verify

Test the Solr installation by starting the application server under which Solr is installed, and connecting to the Solr `/admin` page URL. For example if Solr is installed on `dcindex.dataconservancy.org` using port 8080 then connect to <http://dcindex.dataconservancy.org:8080/solr/admin/>.



You should see a web page similar to the following:

Note that the DCS schema (`dcs-default`) is referenced in the page heading, and that the Solr Home should be pointing at `$SOLR_HOME`. This URL should be accessible from the `dcservice` server, and should be tested by using `'curl'` to retrieve the Solr `/admin` page URL. Remember, the `dcservice` machine assumes it has exclusive access to the `dcindex` server running Solr. no other applications should be adding to or updating the indexes managed by the `dcindex` server.

4. Re-initialize

Re-initializing Solr will mean deleting all of the data in the index. The only time you want to do this is if you have also re-initialized the Fedora archive. There is no supported way to re-index an existing Fedora archive, so be sure this is a step you want to take.

First, stop Solr by shutting down the application server it is deployed in (Tomcat, Jetty, etc.). Next, delete all of the files under `$SOLR_HOME/data/index`, then re-start Solr. Solr will initialize an empty, fresh, index.

DC Services

- 1. Install

- 1.1. Prerequisites
 - 1.1.1. Sun Java JRE, 1.6.0_31+
 - 1.1.2. Jetty Application Server, version 7.6
 - 1.1.2.1. Installing Jetty
 - 1.1.2.2. Helpful Jetty URLs
 - 1.1.3. Derby Relational Database, version 10.8.2.2
 - 1.1.3.1. Installing Derby
 - 1.1.3.2. Helpful Derby URLs
- 1.2. Shared File Storage
- 1.3. Install DC Services Application
- 2. Configure
 - 2.1. Jetty Configuration
 - 2.2. Derby Configuration
 - 2.3. DC Services Configuration
- 3. Verify
 - 3.1. External Dependencies
 - 3.2. Start Jetty
- 4. Start & Stop
- 5. Re-initialize

1. Install

1.1. Prerequisites

1.1.1. Sun Java JRE, 1.6.0_31+

Install Java JRE: Sun Java 1.6.0_31 <= x < Sun Java 1.7 (At least Sun Java 1.6.0_31 but less than Java 1.7)

- Insure that your JAVA_HOME environment variable is pointing to the directory you installed the JRE into
- Insure that \$JAVA_HOME/bin/java is on your command path (e.g. PATH=\$JAVA_HOME/bin:\$PATH)
- Verify your Java installation by
 - Insuring you are pointing to the correct java binary by running: `which java`
 - Insure you installed the correct version of java by running: `java -version`
 - Verifying the Java installation can be a bit tricky on the Mac OS platform, due to the way Java is installed

1.1.2. Jetty Application Server, version 7.6

1.1.2.1. Installing Jetty

Install the Jetty Application Server, version 7.6

- Jetty is required due to the way Tomcat 6.x processes certain request URLs
 - This requirement will be relaxed in the future to support any Servlet 2.5/JSP 2.1 container.
- The installation directory for Jetty will be referred to as \$JETTY_HOME
- Simply download the Jetty 7 stable distribution, and unpack it into \$JETTY_HOME
- Verify your installation of Jetty by starting it using: `$JETTY_HOME/bin/jetty.sh start` and loading the Jetty homepage in your browser

You should see something similar in your browser, indicating you have installed and started Jetty successfully.



Once verifying it is installed successfully, stop Jetty by typing `$JETTY_HOME/bin/jetty.sh stop`

- Systems administrators may note that the `$JETTY_HOME/bin/jetty.sh` will need to be properly invoked from `/etc/init.d` in order to start and stop Jetty during system reboots.

1.1.2.2. Helpful Jetty URLs

- Jetty home page is here: <http://www.eclipse.org/jetty/about.php>
- Jetty 7 can be downloaded here: <http://download.eclipse.org/jetty/stable-7/dist/>

- Jetty documentation is available here: <http://wiki.eclipse.org/Jetty>

1.1.3. Derby Relational Database, version 10.8.2.2

1.1.3.1. Installing Derby

Install the Derby Relational Database: version 10.8.2.2 (we plan on supporting Postgres and potentially other RDBMSs in future releases).

1. Download and unpack Derby 10.8.2.2
 - a. The installation directory for Derby will be referred to as \$DERBY_HOME

That's it!

1.1.3.2. Helpful Derby URLs

- Derby home page is here: <http://db.apache.org/derby/>
- Derby downloads are available here: <http://db.apache.org/derby/releases/release-10.8.2.2.cgi>
- Derby documentation is available here: http://db.apache.org/derby/manuals/index.html#docs_10.8
 - Derby Reference Manual: <http://db.apache.org/derby/docs/10.8/ref/>
 - System Properties
 - Connection URL syntax
 - Connection URL attributes
 - Data Types
 - Derby Developers Guide: <http://db.apache.org/derby/docs/10.8/devguide/>
 - Derby & JDBC basics

1.2. Shared File Storage

Set up a common storage location for the DCS Services and DCS Archive servers ([Shared File Storage](#)). The dcservices and dcarchive must share access to \$DCS_HOME/files. This means that one of the two servers will host the \$DCS_HOME/files directory, and export it using NFS, and the other system will mount the exported \$DCS_HOME/files directory. The important thing to understand is that the file path to \$DCS_HOME/files must be the same on both servers.

If \$DCS_HOME is defined as /usr/local/dcs on the dcservice machine, and dcservice exports the /usr/local/dcs/files filesystem, then that means that \$DCS_HOME must be /usr/local/dcs on the dcarchive machine, and /usr/local/dcs/files on dcarchive machine must be (or symlink to) the mounted NFS filesystem from the dcservice machine.

The dcservice machine needs read/write access to \$DCS_HOME/files, while the dcarchive machine only requires read access.

Verify the correct setup by logging into the dcservice machine, and typing

- echo "Hello DCS world!" >> \$DCS_HOME/files/sample_data_file.txt

Next, login to the dcarchive machine and read the file by typing:

- cat \$DCS_HOME/files/sample_data_file.txt

1.3. Install DC Services Application

1. Determine where your DCS Home directory will be.
 - a. These instructions will use /usr/local/dcs as \$DCS_HOME.
 - b. \$DCS_HOME should be readable, writable, and executable by the dcs user.
1. Download the dcs.war file from the Data Conservancy distribution website ([Download DCS Version 1.1.0-alpha](#)).
2. Save it as dcs.war
3. Copy the dcs.war file into the Jetty application server webapps directory:

```
cp dcs.war $JETTY_HOME/webapps/dcs.war
```

2. Configure

2.1. Jetty Configuration

1. Edit \$JETTY_HOME/etc/webdefault.xml and change welcomeServlets and redirectWelcome parameters to true.
2. Edit \$JETTY_HOME/start.ini and uncomment --exec, and add the following parameters each on their own line below --exec:
 - a. -Xmx1024m
 - b. -XX:MaxPermSize=256m

2.2. Derby Configuration

The jar containing the Derby JDBC driver needs to reside in a shared classloader on your application server:

- Copy \$DERBY_HOME/lib/derby.jar to \$JETTY_HOME/lib/ext

2.3. DC Services Configuration

Create the DC services configuration file (\$JETTY_HOME/resources/dcs.properties) with the following content:

```
# Shared stable filesystem and http paths for dcs
dcs.home=/usr/local/dcs
dcs.baseurl=http://localhost:8080/dcs
dcs.connector.port=8080

# Properties to connect to the Derby database server for the ID service
dcs.idservice.hibernate.dialect=org.hibernate.dialect.DerbyDialect
dcs.idservice.hibernate.driverClassName=org.apache.derby.jdbc.EmbeddedDriver
dcs.idservice.hibernate.url=jdbc:derby:/usr/local/dcs/id/dcsidservice;create=true
# This should match the derby.user.dcsid in derby.properties above
dcs.idservice.hibernate.username = dcsid
dcs.idservice.hibernate.password = dcsid

# connection properties for the Solr installation
dcs.access.solr = org.apache.solr.client.solrj.impl.CommonsHttpSolrServer
dcs.access.solr.url = http://dcindex.dataconservancy.org:8080/solr/

# Access access details for dcarchive server and Fedora
dcs.archive = org.dataconservancy.archive.impl.fcrepo.FcrepoArchiveStore
dcs.archive.fcrepo.url = http://dcarchive.dataconservancy.org:8980/fedora/
dcs.archive.fcrepo.user = fedoraAdmin
dcs.archive.fcrepo.password = fedoraAdmin
```

You will need to customize the the values of the following properties to suite your environment:

1. dcs.home: This should be the value of \$DCS_HOME, e.g. /usr/local/dcs
2. dcs.baseurl: This should be the URL to the DCS application, the URL your users will use to access the DCS. When Jetty is up and running, you should be able to retrieve this URL **TODO: Verify**
3. **TODO: dcs.connector (do we even need this)**
4. dcs.idservice.hibernate.url: This URL should include the path to the \$DCS_HOME directory, with /id/dcsidservice appended to it
a. If your \$DCS_HOME was /var/dcs, then the value of dcs.idservice.hibernate.url would be
jdbc:derby:/var/dcs/id/dcsidservice;create=true
5. dcs.access.solr.url: This should be the URL for your Solr installation. For example, you should be able to append /admin to this value, and **successfully load** the Solr administration page.
6. dcs.archive.fcrepo.url: This should be the URL for your Fedora installation. For example, you should be able to append /describe to this value, and **successfully load** the Fedora "describe" page.

3. Verify

To start the DCS, first ensure that all of the external dependencies are available, then start Jetty (which launches the DCS application).

3.1. External Dependencies

Be sure that the dcindex and dcarchive servers are up, and that their services are available from the dcservice server:

- Use curl from the dcservice server to load the Solr /admin URL
- Use curl from the dcservice to load the Fedora /describe URL

3.2. Start Jetty

Start Jetty by following the instructions in the section below.

Verify that the DCS is available by performing a search of the DCS using the query api (in this example, dcs.baseurl == <http://localhost:8080/dcs/>):

- curl 'http://localhost:8080/dcs/query/?q=entityType:*'

If this is an initial install of the DCS, then the query will simply return an empty DCP package:

```
<dcp xmlns="http://dataconservancy.org/schemas/dcp/1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://dataconservancy.org/schemas/dcp/1.0
http://dataconservancy.org/schemas/dcp/1.0" />
```

4. Start & Stop

Start the Jetty application server:

- `$JETTY_HOME/bin/jetty.sh start`

To stop the Jetty application server:

- `$JETTY_HOME/bin/jetty.sh stop`

5. Re-initialize

Re-initializing the DCS means that you will delete all of the data it manages, the indexes it maintains, and the identifiers it has allocated. You might want to do this if you are simply "kicking the tires" of the DCS. Do not re-initialize the DCS if you are concerned that you might lose valuable data.

1. Shutdown Jetty
2. Delete the contents of the "staging" and "files" directories in `dc.home`, for example (if `$DCS_HOME` == `{/usr/local/dcs}`):
 - a. `rm -rf /usr/local/dcs/staging/*`
 - b. `rm -rf /usr/local/dcs/access/*`
3. Delete the `dcidservice` database
 - a. `rm -rf /usr/local/dcs/id`
4. [Re-initialize Fedora](#)
5. [Re-initialize Solr](#)
6. Start Jetty
7. Verify your DCS is empty by performing the following query:
 - a. `curl 'http://localhost:8080/dcs/query/?q=entityType:*'`
 - b. The query should return an empty XML document

User Interface

For installation instructions of the Data Conservancy Reference User Interface, application please see the [User Interface Administration Guide](#).

Exemplar Configuration - Johns Hopkins Data Management Service

This section outlines the configuration used in the current Johns Hopkins Data Management Data Conservancy instance. It provides an example of the server specifications (architecture and platform) as well as the major components (database components, application servers etc) of an instance of DCS.

Server Specification

Components

Server Specification

The server requirements for a specific DC Instance will depend on the scope of the data curation services being implemented. For example, the data volumes collected from project-to-project in all disciplines is highly variable. Correspondingly, the amount of hard disk drive (HDD) and tape storage space needed for any particular instance will vary widely depending on the data to be managed and curated.

The Data Conservancy software has no explicit hardware requirements. DC Instances can be installed with varying amounts of RAM space, storage space, and processor speeds.

Hardware requirements should be assessed prior to installing an Instance. Hardware needs can then be assessed on an ongoing basis as the Instance's performance is evaluated and any bottlenecks are identified.

As an example of a possible server and hardware configuration, the hardware being used for the current Johns Hopkins Data Management Data Conservancy instance are shown below:

- 1. DC Service machine (dcservice-dataconservancy.org)**
 - This machine runs the application servers that host the DC services and user interface applications.
 - Linux CentOS 5.8
 - 2 GB RAM,
 - the staging machine currently has 30 GB HDD to use in the ingest pipeline,
 - using 1.7 TB mass storage for staging.
 - Dual CPUs @ 2.40 GHz.
- 2. DC Index machine (dcindex.dataconservancy.org)**
 - This machine runs the Solr search engine and index server and associated components.
 - Linux CentOS 5.8
 - 8 GB RAM,
 - single CPU @ 2.40 GHz,
 - 30 GB HDD on local disk.
- 3. DC Archive machine (dcarchive.dataconservancy.org)**
 - This machine runs the Fedora Commons Repository server and components.
 - Linux CentOS 5.8
 - 8 GB RAM,
 - 30 GB HDD,
 - 1.7TB mass storage.
 - Dual CPUs @ 2.40 GHz.

Components

The following section documents the 3rd party software requirements installed on the Johns Hopkins Data Management instance of Data Conservancy.

These include such things as Java Application Servers to run the DCS web applications; the relational databases required to operate DCS; major 3rd party components such as Fedora Commons Repository used for content storage management and Solr search engine and index server.

1. DC Service machine (dcservice-dataconservancy.org)

- Sun Java JDK 1.6.0_31
- Jetty 7.5.4
- Apache Derby 10.8.2.2

2. DC Index machine (dcindex.dataconservancy.org)

- Sun Java JDK 1.6.0_31
- Apache Tomcat 6.0.29
- Solr 1.4.1

3. DC Archive machine (dcarchive.dataconservancy.org)

- Sun Java JDK 1.6.0_31
- Fedora Commons Repository 3.5

Backup & Recovery

The following section provides an outline and links to system documentation in order to ensure that the various components of the DCS installation are considered as part of your institutions backup and recovery plan.

[Fedora](#)

[Solr](#)

[DCS Database](#)

Fedora

Please see the [Fedora documentation](#) for information on Fedora Backup and Recovery.

Solr

Please refer to the Solr documentation for [backup and recovery](#) information.

DCS Database

Please see the Derby database documentation on [Backup and Recovery](#).