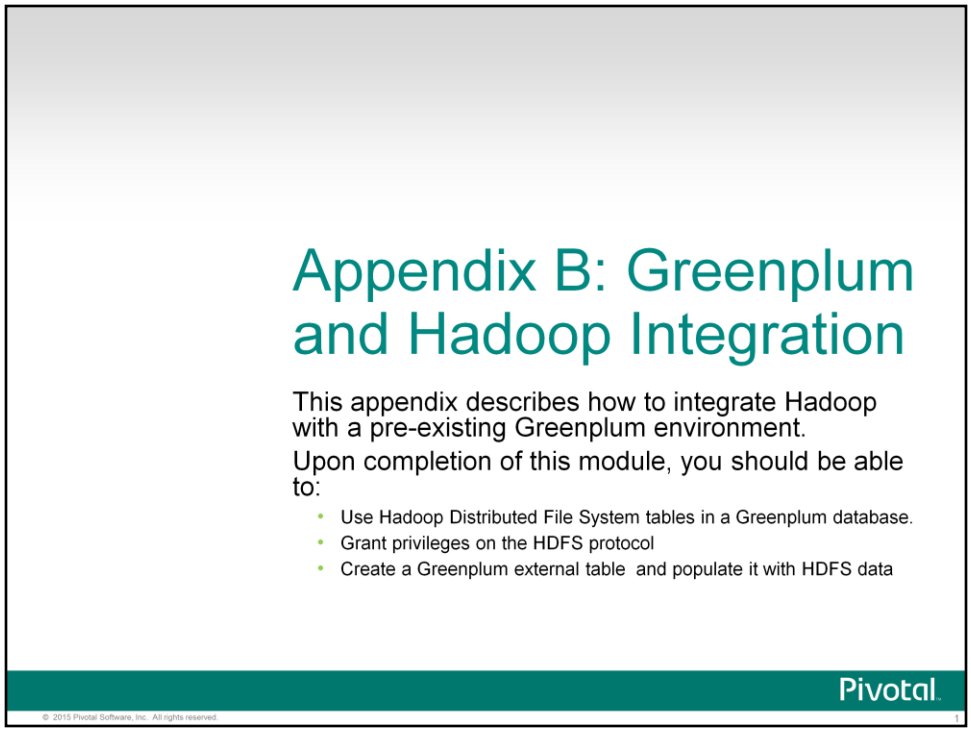


Appendix B: Greenplum and Hadoop Integration

Data co-processing of structured and unstructured data with Greenplum Database and Hadoop respectively is possible by integrating the two databases and taking advantage of the parallel processing architecture inherent in Greenplum Database.

Greenplum Database supports integration of a Hadoop Distributed File System by providing access to the `gphdfs` protocol to read and write unstructured data.

In this module, you will:

- Integrate Greenplum Database to the Hadoop Distributed File System.
- Create an external table from Greenplum Database to files on a pre-existing Hadoop file system.

Summary of Steps for Hadoop Integration

To integrate a pre-configured Hadoop environment with a pre-existing Greenplum Database, the following must be performed:

- Set environmental variables required for integration process
- Set Greenplum configuration parameters for working with Hadoop
- Grant privileges to HDFS protocol

Once configured, you can create a Greenplum external table to content in Hadoop

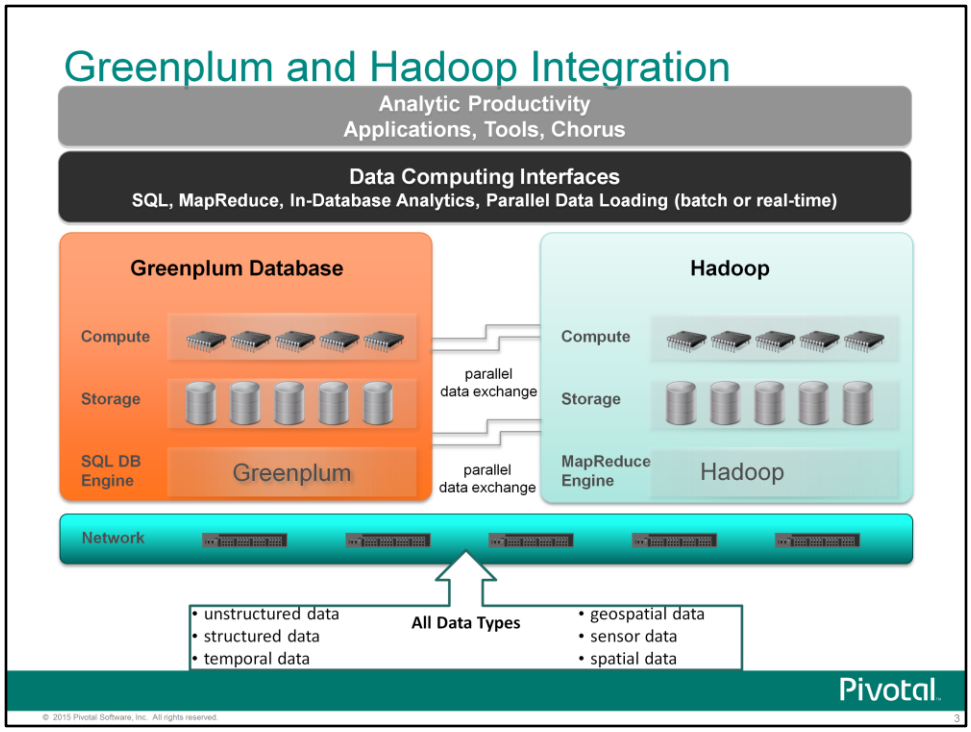
Pivotal

2

To integrate Greenplum to Hadoop Distributed File System you must:

- Configure several environmental variables
- Modify several Greenplum configuration parameters required for Hadoop integration
- Perform a one-setup and grant privileges for the HDFS protocol.
- Identify and execute on-time setup
- Grant privileges for the HDFS protocol

Once the integration between the two databases has been configured, you can create a Greenplum external table to content that resides on the Hadoop Distribute File System.

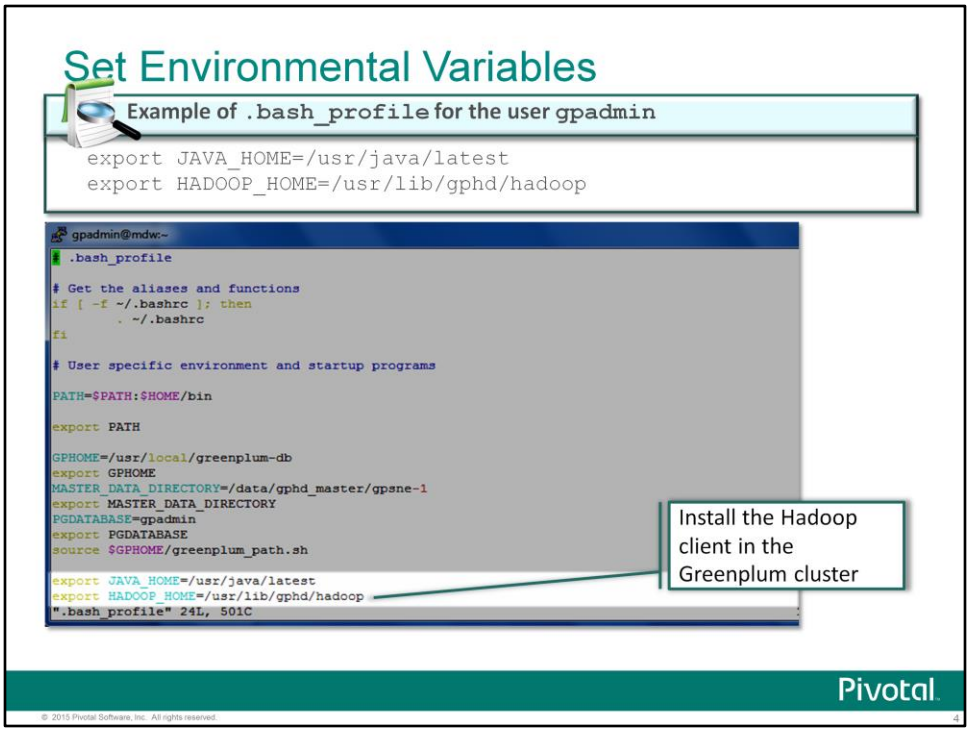


Greenplum and Hadoop Integration

Greenplum Database and Hadoop are complementary technologies that deliver a powerful solution for the analytics of structured, semi-structured, and unstructured data.

Some benefits of integrating the two technologies are listed below.

- Perform complex, high-speed, interactive analytics using Greenplum Database.
- Analytic Productivity, applications, Tools, and Chorus can be used on HDFS (Hadoop Distributed File System) data
- Data computing interfaces such as SQL and MapReduce
- Stream the data directly from Hadoop into Greenplum Database to incorporate unstructured or semi-structured data in the above analyses within Greenplum Database
- Hadoop can also be used to transform unstructured and semi-structured data into a structured format that can then be fed into Greenplum Database for high speed, interactive querying



Set Environmental Variables

You will need to set several environmental variables for the `gpadmin` user in your Greenplum environment.

From the Greenplum master server, add the following lines to the `.bash_profile` of the `gpadmin` user.:

- `export JAVA_HOME=/usr/java/latest`
- `export HADOOP_HOME=/usr/lib/gphd/hadoop`

The `JAVA_HOME` variable should be set to the location of java on the master server. The integration requires Java 1.6, which you will need to download and install on all Greenplum database hosts, including the master, standby, and segment hosts.

The `HADOOP_HOME` variable must point to the location of the Hadoop libraries, which are available from the Hadoop client. You will need to install the Hadoop client in the Greenplum cluster on all hosts before proceeding. This ensures access to all Hadoop libraries.

Server Configuration Parameters for Hadoop Targets

Parameter	Available Value	Hadoop Distribution	Default Value
gp_hadoop_target_version	gphd-1.0	Greenplum HD 1.1/1.2	gphd-1.1
	gphd-1.1		
	gphd-1.2		
	gphd-2.0	Pivotal HD 1.0/2.0	
	gpmr-1.0	MapR 1.x, 2.x, 3.x	
	gpmr-1.2	MapR 4.x	
	cdh3u2	Cloudera 4.1-4.7	
	cdh4.1	Cloudera 5.0/5.1	
	hdp2	Hortonworks Data Platform 2.1	
gp_hadoop_home	The value stored in the $HADOOP_HOME variable		NULL

Pivotal

5

Server Configuration Parameters for Hadoop Targets

After setting the parameters, you will need to update the following master server configuration parameters in the `postgresql.conf` file with the `gpconfig` command:

- `gp_hadoop_target_version`
- `gp_hadoop_home`

If not specified in the `postgresql.conf`, Greenplum will use the `gphd-1.1` library for the integration. If your distribution differs, select the appropriate value from the Available Value column.

The `gp_hadoop_home` parameter must be the same as the `HADOOP_HOME` variable set in the `.bash_profile` file. In this case, it is set to `/usr/lib/gphd/hadoop`.

After modifying the `postgresql.conf` file, you will need to re-read the `postgresql.conf` file with the `gpstop -u` command.

Setting the Parameter Values

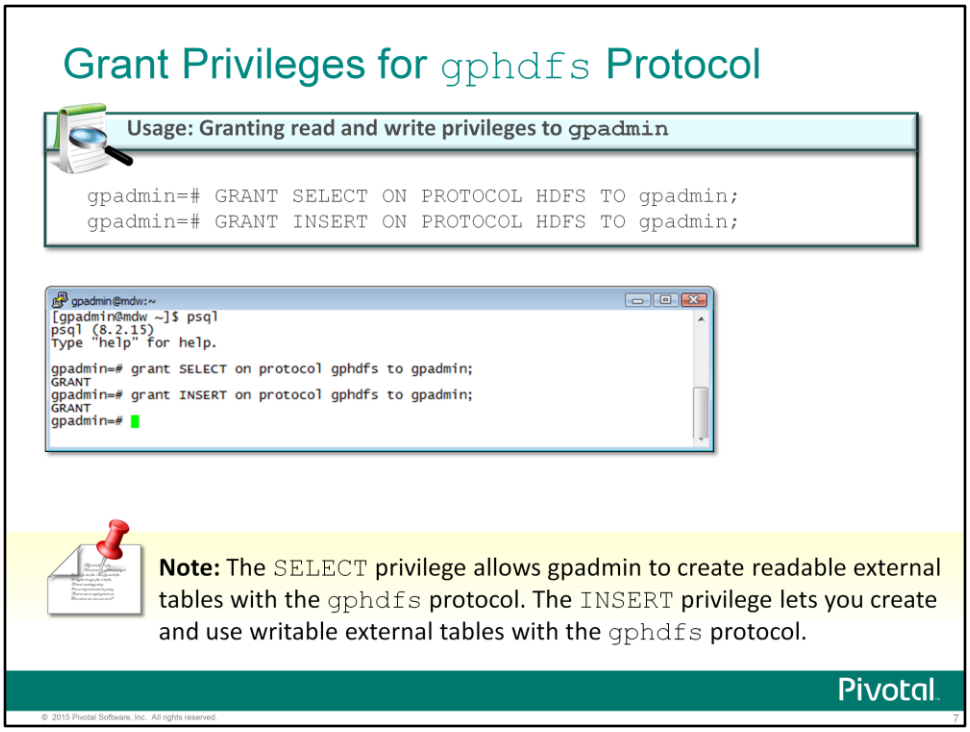
gpadmin@mdw:~

```
[gpadmin@mdw ~]$ gpconfig -s gp_hadoop_home
Values on all segments are consistent
GUC          : gp_hadoop_home
Master  value:
Segment value:
[gpadmin@mdw ~]$ gpconfig -s gp_hadoop_target_version
Values on all segments are consistent
GUC          : gp_hadoop_target_version
Master  value: gphd-1.1
Segment value: gphd-1.1
[gpadmin@mdw ~]$ gpconfig -c gp_hadoop_home -v "'/usr/lib/gphd/hadoop'"
20150413:12:09:52:010657 gpconfig:mdw:gpadmin-[INFO]:-completed successfully
[gpadmin@mdw ~]$ gpstop -u
20150413:12:09:56:010720 gpstop:mdw:gpadmin-[INFO]:-Starting gpstop with args: -u
20150413:12:09:56:010720 gpstop:mdw:gpadmin-[INFO]:-Gathering information and valid
ating the environment...
20150413:12:09:56:010720 gpstop:mdw:gpadmin-[INFO]:-Obtaining Greenplum Master cata
log information
20150413:12:09:56:010720 gpstop:mdw:gpadmin-[INFO]:-Obtaining Segment details from
master...
20150413:12:09:57:010720 gpstop:mdw:gpadmin-[INFO]:-Greenplum Version: 'postgres (G
reenplum Database) 4.3.4.0 build 1'
20150413:12:09:57:010720 gpstop:mdw:gpadmin-[INFO]:-Signalling all postmaster proce
sses to reload

[gpadmin@mdw ~]$ gpconfig -s gp_hadoop_home
Values on all segments are consistent
GUC          : gp_hadoop_home
Master  value: /usr/lib/gphd/hadoop
Segment value: /usr/lib/gphd/hadoop
[gpadmin@mdw ~]$
```

Update the parameters to the appropriate values

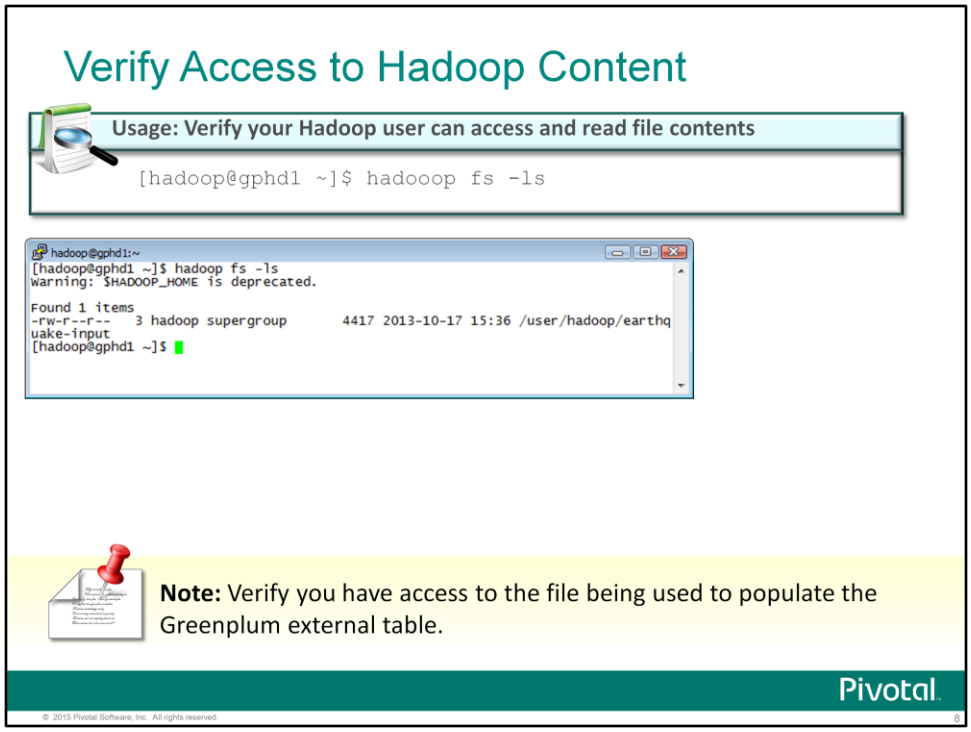
Pivotal



Grant Privileges for `gphdfs` Protocol

You must now provide read and write privileges on the gphdfs protocol to the gpadmin user. This lets the gpadmin user create readable and writeable external tables to content on HDFS using the gphdfs protocol.

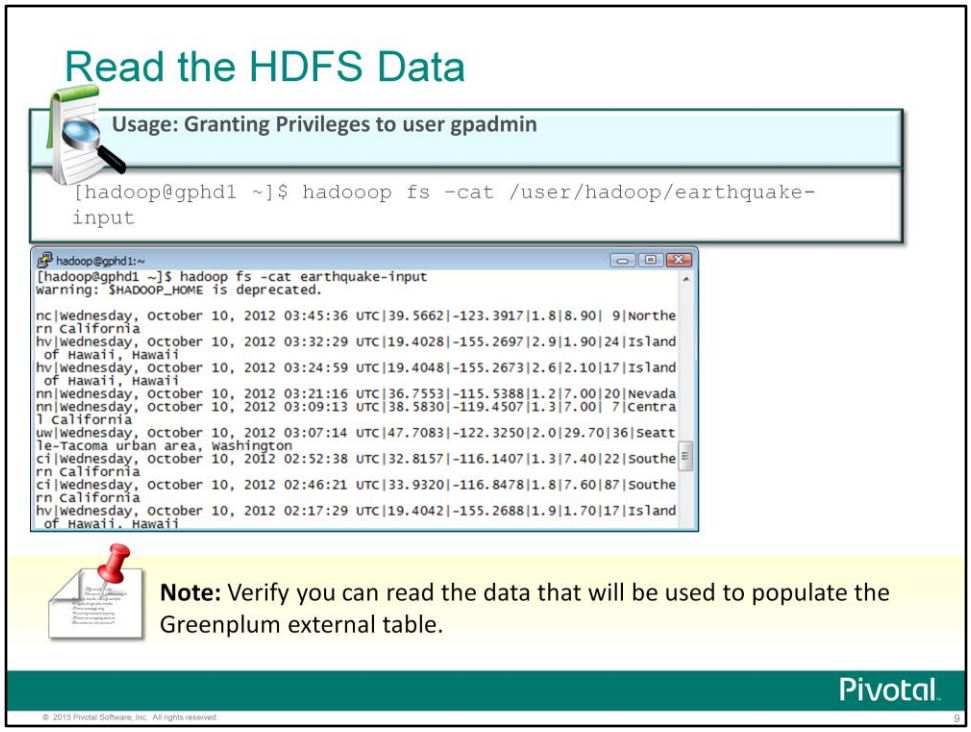
From a `pSQL` session on to the Greenplum environment you are configuring for integration, execute the grant command for `SELECT` and `INSERT` privileges for the **`gpadmin`** user.



Verify Access to Hadoop Content

Before attempting to create an external table and read or write to the HDFS system, you should verify that your Hadoop user can access and read content from HDFS.

In this example, the `hadoop fs -ls` command is used to list the user-related content for the Hadoop user, hadoop.



Read the HDFS Data

Read the HDFS data using the `hadoop fs -cat` command. By viewing the content of your data, you can verify how many and what type of columns you will use when creating the external table.

Creating an External Table

Usage: Example of an external table creation

```
gpadmin=# CREATE EXTERNAL TABLE TABLE_NAME (FIELD NAMES)
LOCATION ('gphdfs://hdfs_host:port/path/file_name')
FORMAT 'TEXT' (DELIMITER '|');
```

Port number

Hadoop IP Host

```
gpadmin=#
gpadmin=#
gpadmin=# create external table ext_hdfs (
gpadmin(# f1 text,
gpadmin(# f2 text,
gpadmin(# f3 float,
gpadmin(# f4 float,
gpadmin(# f5 float,
gpadmin(# f6 float,
gpadmin(# f7 integer,
gpadmin(# f8 text)
gpadmin-# location
gpadmin-# ('gphdfs://172.16.1.21:9000/user/hadoop/earthquake-input') format 'TEXT' (delimiter '|');
CREATE EXTERNAL TABLE
gpadmin=#
gpadmin=#
```

Note: Know the field types you are going to create on your external table. A field omitted or wrongly defined can compromise your data retrieval.

Pivotal

10

Creating an External Table

The **`gphdfs`** protocol allows for a link between Greenplum and HDFS. Thus, a Greenplum external table can be populated using getting HDFS unstructured data. The data is read in parallel from HDFS into the Greenplum segments for a fast process.

Connect to your Greenplum Database environment to create an external table as shown.

Accessing HDFS Data Loaded into Greenplum

```
gpadmin@mdw:~
[gpadmin@mdw ~]$ psql
psql (8.2.15)
Type "help" for help.

gpadmin=# select count(*)from ext_hdfs ;
 count
-------
    47
(1 row)
```

Once the HDFS data has been loaded into Greenplum, you can use it as regular Greenplum data

```
gpadmin@mdw:~
[gpadmin@mdw ~]$ psql
psql (8.2.15)
Type "help" for help.

gpadmin=# select * from ext_hdfs limit 10;
 f1 |                    f2                    |    f3    |    f4     | f5  |  f6  | f7 |                    f8
----+------------------------------------------+----------+-----------+-----+------+----+-------------------------------------
 nc | Wednesday, October 10, 2012 03:45:36 UTC | 39.5662  | -123.3917 | 1.8 |  8.9 |  9 | Northern California
 hv | Wednesday, October 10, 2012 03:32:29 UTC | 19.4028  | -155.2697 | 2.9 |  1.9 | 24 | Island of Hawaii, Hawaii
 hv | Wednesday, October 10, 2012 03:24:59 UTC | 19.4048  | -155.2673 | 2.6 |  2.1 | 17 | Island of Hawaii, Hawaii
 nn | Wednesday, October 10, 2012 03:21:16 UTC | 36.7553  | -115.5388 | 1.2 |    7 | 20 | Nevada
 nn | Wednesday, October 10, 2012 03:09:13 UTC |  38.583  | -119.4507 | 1.3 |    7 |  7 | Central California
 uw | Wednesday, October 10, 2012 03:07:14 UTC | 47.7083  |  -122.325 |   2 | 29.7 | 36 | Seattle-Tacoma urban area, Washington
 ci | Wednesday, October 10, 2012 02:52:38 UTC | 32.8157  | -116.1407 | 1.3 |  7.4 | 22 | Southern California
 ci | Wednesday, October 10, 2012 02:46:21 UTC |  33.932  | -116.8478 | 1.8 |  7.6 | 87 | Southern California
 hv | Wednesday, October 10, 2012 02:17:29 UTC | 19.4042  | -155.2688 | 1.9 |  1.7 | 17 | Island of Hawaii, Hawaii
 ak | Wednesday, October 10, 2012 02:06:25 UTC | 60.5271  | -152.2992 | 1.3 | 83.9 |  7 | Southern Alaska
(10 rows)

gpadmin=#
```

Pivotal

11

Accessing HDFS Data Loaded into Greenplum

Once the data has been loaded into the Greenplum external table you can use it as regular Greenplum data.

Pivotal

A NEW PLATFORM FOR A NEW ERA