NASA_Logs_Analysis

default

```
%md

## Analyzing Unstructured Data & leveraging spark sql functions

* This tutorial analyzes apache web server logs released by NASA and available [here](http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html)
* This tutorial shows how to parse log files
* How to leverage spark sql functions to analyze
```

FINISHED

Analyzing Unstructured Data & leveraging spark sql functions

- This tutorial analyzes apache web server logs released by NASA and available here (http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html)
- This tutorial shows how to parse log files
- How to leverage spark sql functions to analyze

Took 0 seconds

```
sc
```

FINISHED

```
res0: org.apache.spark.SparkContext = org.apache.spark.SparkContext@5a3e01b8
```

Took 30 seconds

```
// Let's read the logs file available at /home/hadoop/lab/data
var base_df = sqlContext.read.text( "file:///home/hadoop/lab/data/access_log_Aug95" )
```

FINISHED

```
base_df: org.apache.spark.sql.DataFrame = [value: string]
```

Took 0 seconds

```
// Let's look at the schema
base_df.printSchema()
```

FINISHED

```
root
 |-- value: string (nullable = true)
```

NASA_Logs_Analysis

READY default

READY

FINISHED

```
// Let's print first few line. Show by default shows 20 lines
base_df.show( truncate = false )
```

```
+-----------------------------------------------------------------------------------------------------------------------+
|value                                                                                                                  |
+-----------------------------------------------------------------------------------------------------------------------+
|in24.inetnebr.com - - [01/Aug/1995:00:00:01 -0400] "GET /shuttle/missions/sts-68/news/sts-68-mcc-05.txt HTTP/1.0" 200 1839     |
|uplherc.upl.com - - [01/Aug/1995:00:00:07 -0400] "GET / HTTP/1.0" 304 0                                                |
|uplherc.upl.com - - [01/Aug/1995:00:00:08 -0400] "GET /images/ksclogo-medium.gif HTTP/1.0" 304 0                       |
|uplherc.upl.com - - [01/Aug/1995:00:00:08 -0400] "GET /images/MOSAIC-logosmall.gif HTTP/1.0" 304 0                     |
|uplherc.upl.com - - [01/Aug/1995:00:00:08 -0400] "GET /images/USA-logosmall.gif HTTP/1.0" 304 0                        |
|ix-esc-ca2-07.ix.netcom.com - - [01/Aug/1995:00:00:09 -0400] "GET /images/launch-logo.gif HTTP/1.0" 200 1713           |
|uplherc.upl.com - - [01/Aug/1995:00:00:10 -0400] "GET /images/WORLD-logosmall.gif HTTP/1.0" 304 0                      |
|slppp6.intermind.net - - [01/Aug/1995:00:00:10 -0400] "GET /history/skylab/skylab.html HTTP/1.0" 200 1687              |
|piweba4y.prodigy.com - - [01/Aug/1995:00:00:10 -0400] "GET /images/launchmedium.gif HTTP/1.0" 200 11853               |
|slppp6.intermind.net - - [01/Aug/1995:00:00:11 -0400] "GET /history/skylab/skylab-small.gif HTTP/1.0" 200 9202         |
|slppp6.intermind.net - - [01/Aug/1995:00:00:12 -0400] "GET /images/ksclogosmall.gif HTTP/1.0" 200 3635                 |
|ix-esc-ca2-07.ix.netcom.com - - [01/Aug/1995:00:00:12 -0400] "GET /history/apollo/images/apollo-logo1.gif HTTP/1.0" 200 1173 |
|slppp6.intermind.net - - [01/Aug/1995:00:00:13 -0400] "GET /history/apollo/images/apollo-logo.gif HTTP/1.0" 200 3047     |
|uplherc.upl.com - - [01/Aug/1995:00:00:14 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 304 0                       |
|133.43.96.45 - - [01/Aug/1995:00:00:16 -0400] "GET /shuttle/missions/sts-69/mission-sts-69.html HTTP/1.0" 200 10566     |
|kgtyk4.kj.yamagata-u.ac.jp - - [01/Aug/1995:00:00:17 -0400] "GET / HTTP/1.0" 200 7280                                  |
|kgtyk4.kj.yamagata-u.ac.jp - - [01/Aug/1995:00:00:18 -0400] "GET /images/ksclogo-medium.gif HTTP/1.0" 200 5866         |
|d0ucr6.fnal.gov - - [01/Aug/1995:00:00:19 -0400] "GET /history/apollo/apollo-16/apollo-16.html HTTP/1.0" 200 2743      |
|ix-esc-ca2-07.ix.netcom.com - - [01/Aug/1995:00:00:19 -0400] "GET /shuttle/resources/orbiters/discovery.html HTTP/1.0" 200 6849|
|d0ucr6.fnal.gov - - [01/Aug/1995:00:00:20 -0400] "GET /history/apollo/apollo-16/apollo-16-patch-small.gif HTTP/1.0" 200 14897 |
+-----------------------------------------------------------------------------------------------------------------------+
only showing top 20 rows
```

Took 1 seconds (outdated)

FINISHED

```
%md

The logs follow [Common Log Format](https://www.w3.org/Daemon/User/Config/Logging.html#common-logfile-format)

* remotehost  - Remote hostname (or IP number if DNS hostname is not available, or if DNSLookup is Off.
* rfc931 - The remote logname of the user.
* authuser - The username as which the user has authenticated himself.
* [date] - Date and time of the request.
* "request" - The request line exactly as it came from the client.
* status - The HTTP status code returned to the client.
* bytes - The content-length of the document transferred.
```

The logs follow Common Log Format (https://www.w3.org/Daemon/User/Config/Logging.html#common-logfile-format)

- remotehost - Remote hostname (or IP number if DNS hostname is not available, or if DNSLookup is Off.
- rfc931 - The remote logname of the user.
- authuser - The username as which the user has authenticated himself.
- [date] - Date and time of the request.
- “request” - The request line exactly as it came from the client.
- status - The HTTP status code returned to the client.
- bytes - The content-length of the document transferred.

Took 0 seconds

FINISHED

```
// import all spark sql functions
import org.apache.spark.sql.functions._
```

```
import org.apache.spark.sql.functions._
```

Took 1 seconds

FINISHED

```
// Let's see if we can extract the host name or ip address from the logs into a dataframe
val split_df = base_df.select( regexp_extract(base_df("value"), """^([^\s]+\s)""", 1).alias("host") )
split_df.show(10, truncate=false)
```

```
split_df: org.apache.spark.sql.DataFrame = [host: string]
+---------------------------+
|host                       |
+---------------------------+
|in24.inetnebr.com          |
```

```
|uplherc.upl.com             |
|uplherc.upl.com             |
|uplherc.upl.com             |
|uplherc.upl.com             |
|ix-esc-ca2-07.ix.netcom.com |
|uplherc.upl.com             |
|slppp6.intermind.net        |
|piweba4y.prodigy.com        |
|slppp6.intermind.net        |
+----------------------------+
only showing top 10 rows
```

Took 1 seconds

FINISHED

```
// Now we can extract all elements required from the logs into a dataframe
var split_df = base_df.select(regexp_extract(base_df("value"), """^([^\s]+\s)""", 1).alias("host"),
                          regexp_extract(base_df("value"), """^.*\[(\d\d/\w{3}/\d{4}:\d{2}:\d{2}:\d{2} -\d{4})]""", 1).alias("timestamp"),
                          regexp_extract(base_df("value"), """^.*"\w+\s+([^\s]+)\s+HTTP.*"""", 1).alias("path"),
                          regexp_extract(base_df("value"), """^.*"\s+([^\s]+)""", 1).cast("integer").alias("status"),
                          regexp_extract(base_df("value"), """^.*\s+(\d+)$""", 1).cast("integer").alias("content_size"))
split_df.show( 10, truncate = false )
```

```
split_df: org.apache.spark.sql.DataFrame = [host: string, timestamp: string, path: string, status: int, content_size: int]
+----------------------------+-------------------------+----------------------------------------------+------+------------+
|host                        |timestamp                |path                                          |status|content_size|
+----------------------------+-------------------------+----------------------------------------------+------+------------+
|in24.inetnebr.com           |01/Aug/1995:00:00:01 -0400|/shuttle/missions/sts-68/news/sts-68-mcc-05.txt|200   |1839        |
|uplherc.upl.com             |01/Aug/1995:00:00:07 -0400|/                                             |304   |0           |
|uplherc.upl.com             |01/Aug/1995:00:00:08 -0400|/images/ksclogo-medium.gif                    |304   |0           |
|uplherc.upl.com             |01/Aug/1995:00:00:08 -0400|/images/MOSAIC-logosmall.gif                  |304   |0           |
|uplherc.upl.com             |01/Aug/1995:00:00:08 -0400|/images/USA-logosmall.gif                     |304   |0           |
|ix-esc-ca2-07.ix.netcom.com |01/Aug/1995:00:00:09 -0400|/images/launch-logo.gif                       |200   |1713        |
|uplherc.upl.com             |01/Aug/1995:00:00:10 -0400|/images/WORLD-logosmall.gif                   |304   |0           |
|slppp6.intermind.net        |01/Aug/1995:00:00:10 -0400|/history/skylab/skylab.html                   |200   |1687        |
|piweba4y.prodigy.com        |01/Aug/1995:00:00:10 -0400|/images/launchmedium.gif                      |200   |11853       |
|slppp6.intermind.net        |01/Aug/1995:00:00:11 -0400|/history/skylab/skylab-small.gif              |200   |9202        |
+----------------------------+-------------------------+----------------------------------------------+------+------------+
only showing top 10 rows
```

Took 1 seconds

```
split_df.cache()
```

FINISHED

```
res41: org.apache.spark.sql.DataFrame = [host: string, timestamp: string, path: string, status: int, content_size: int]
```

Took 0 seconds

```
// What are the columns available in the dataframe?
split_df.columns
```

FINISHED

```
res99: Array[String] = Array(host, timestamp, path, status, content_size)
```

Took 0 seconds

```
// How many null strings in the original apache logs dataframe
base_df.where( base_df("value").isNull ).count()
```

FINISHED

```
res65: Long = 0
```

Took 2 seconds (outdated)

```
// How many null values in the dataframe after parsing
split_df.where( $"host".isNull || $"timestamp".isNull || $"path".isNull || $"status".isNull || $"content_size".isNull ).count()
```

FINISHED

```
res77: Long = 14178
```

Took 1 seconds (outdated)

```
// Let's write a small function to find out if there are any null values in any of the columns
import org.apache.spark.sql.Column

def count_null( col_name:String ): Long = {
  return split_df.filter( split_df(col_name).isNull ).count()
}
```

FINISHED

```
import org.apache.spark.sql.Column
count_null: (col_name: String)Long
```

Took 1 seconds (outdated)

```
// Iterate through the columns to find out
for ( col_name <- split_df.columns ) {
  println( col_name + " : " + count_null( col_name ) )
}
```

FINISHED

```
host : 0
```

```
timestamp : 0
path : 0
status : 0
content_size : 14178
```

Took 2 seconds (outdated)

FINISHED

```
// Looks like all the null values are originating from content_size

split_df.filter( split_df("content_size").isNull ).show( 10 )
```

```
+--------------------+--------------------+--------------------+------+------------+
|                host|           timestamp|                path|status|content_size|
+--------------------+--------------------+--------------------+------+------------+
|        gw1.att.com |01/Aug/1995:00:03...|/shuttle/missions...|   302|        null|
|js002.cc.utsunomi...|01/Aug/1995:00:07...|/shuttle/resource...|   404|        null|
|    tia1.eskimo.com |01/Aug/1995:00:28...|/pub/winvn/releas...|   404|        null|
|itws.info.eng.nii...|01/Aug/1995:00:38...|/ksc.html/facts/a...|   403|        null|
|grimnet23.idirect...|01/Aug/1995:00:50...|/www/software/win...|   404|        null|
|miriworld.its.uni...|01/Aug/1995:01:04...|/history/history.htm|   404|        null|
|      ras38.srv.net |01/Aug/1995:01:05...|/elv/DELTA/uncons...|   404|        null|
| cs1-06.leh.ptd.net |01/Aug/1995:01:17...|                    |   404|        null|
|www-b2.proxy.aol....|01/Aug/1995:01:22...|  /shuttle/countdown|   302|        null|
|    maui56.maui.net |01/Aug/1995:01:31...|            /shuttle|   302|        null|
+--------------------+--------------------+--------------------+------+------------+
only showing top 10 rows
```

Took 0 seconds

FINISHED

```
base_df.filter(!base_df("value").rlike("""\d+$""")).show( 10, truncate = false )
```

```
+-----------------------------------------------------------------------------------------------------------------------------------------------+
|value                                                                                                                                          |
+-----------------------------------------------------------------------------------------------------------------------------------------------+
|gw1.att.com - - [01/Aug/1995:00:03:53 -0400] "GET /shuttle/missions/sts-73/news HTTP/1.0" 302 -                                                |
|js002.cc.utsunomiya-u.ac.jp - - [01/Aug/1995:00:07:33 -0400] "GET /shuttle/resources/orbiters/discovery.gif HTTP/1.0" 404 -|
|tia1.eskimo.com - - [01/Aug/1995:00:28:41 -0400] "GET /pub/winvn/release.txt HTTP/1.0" 404 -                                                   |
|itws.info.eng.niigata-u.ac.jp - - [01/Aug/1995:00:38:01 -0400] "GET /ksc.html/facts/about_ksc.html HTTP/1.0" 403 -                            |
|grimnet23.idirect.com - - [01/Aug/1995:00:50:12 -0400] "GET /www/software/winvn/winvn.html HTTP/1.0" 404 -                                     |
|miriworld.its.unimelb.edu.au - - [01/Aug/1995:01:04:54 -0400] "GET /history/history.htm HTTP/1.0" 404 -                                       |
|ras38.srv.net - - [01/Aug/1995:01:05:14 -0400] "GET /elv/DELTA/uncons.htm HTTP/1.0" 404 -                                                      |
|cs1-06.leh.ptd.net - - [01/Aug/1995:01:17:38 -0400] "GET /sts-71/launch/" 404 -                                                                |
```

```
|www-b2.proxy.aol.com - - [01/Aug/1995:01:22:07 -0400] "GET /shuttle/countdown HTTP/1.0" 302 -                                |
|maui56.maui.net - - [01/Aug/1995:01:31:56 -0400] "GET /shuttle HTTP/1.0" 302 -                                               |
+---------------------------------------------------------------------------------------------------------------------------+
only showing top 10 rows
```

Took 1 seconds

FINISHED

```
// Look like the content_size has a character ( - ). This is when the status code is not 200 that means
// the http request was not successful
var cleaned_df = split_df.na.fill( 0.0 )
cleaned_df.cache()
```

```
cleaned_df: org.apache.spark.sql.DataFrame = [host: string, timestamp: string, path: string, status: int, content_size: int]
```

Took 0 seconds (outdated)

FINISHED

```
// Let's find out
cleaned_df.filter( cleaned_df("content_size").isNull ).count()
```

```
res227: Long = 0
```

Took 1 seconds

FINISHED

```
// Write a functiont return day
// Write an UDF ( User defined function ) to extract week day name from the date field

import java.util.Calendar
import java.util.Locale
import java.text.SimpleDateFormat

def get_day_of_week(tDate: String ): String = {

    val dateFormat = new SimpleDateFormat("dd/MMM/yyyy")
    val c = Calendar.getInstance()
    c.setTime(dateFormat.parse( tDate ) )
    val dayOfWeek = c.getDisplayName(Calendar.DAY_OF_WEEK, Calendar.SHORT, Locale.US)

    return dayOfWeek
}
```

```
import java.util.Calendar
import java.util.Locale
import java.text.SimpleDateFormat
get_day_of_week: (tDate: String)String
```

Took 2 seconds

FINISHED

```
// Test the function
getDayOfWeek( "13/July/2016" )
```

```
res278: String = Wed
```

Took 0 seconds

FINISHED

```
// Register the function as a udf function
import org.apache.spark.sql.functions.udf

val get_day_name = udf( get_day_of_week(_:String) )
```

```
import org.apache.spark.sql.functions.udf
get_day_name: org.apache.spark.sql.UserDefinedFunction = UserDefinedFunction(<function1>,StringType,List(StringType))
```

Took 1 seconds (outdated)

FINISHED

```
// Add a new column called weekday to the dataframe

val nasa_logs_df = cleaned_df.withColumn( "weekday", get_day_name( cleaned_df("timestamp") ) )
```

```
nasa_logs_df: org.apache.spark.sql.DataFrame = [host: string, timestamp: string, path: string, status: int, content_size: int, weekday: string]
```

Took 0 seconds

FINISHED

```
nasa_logs_df.cache()
nasa_logs_df.show( 5, truncate = false )
```

```
res367: nasa_logs_df.type = [host: string, timestamp: string, path: string, status: int, content_size: int, weekday: string]
+------------------+-------------------------+----------------------------------------------+------+------------+-------+
|host              |timestamp                |path                                          |status|content_size|weekday|
+------------------+-------------------------+----------------------------------------------+------+------------+-------+
|in24.inetnebr.com |01/Aug/1995:00:00:01 -0400|/shuttle/missions/sts-68/news/sts-68-mcc-05.txt|200   |1839        |Tue    |
|uplherc.upl.com   |01/Aug/1995:00:00:07 -0400|/                                             |304   |0           |Tue    |
|uplherc.upl.com   |01/Aug/1995:00:00:08 -0400|/images/ksclogo-medium.gif                    |304   |0           |Tue    |
|uplherc.upl.com   |01/Aug/1995:00:00:08 -0400|/images/MOSAIC-logosmall.gif                  |304   |0           |Tue    |
|uplherc.upl.com   |01/Aug/1995:00:00:08 -0400|/images/USA-logosmall.gif                     |304   |0           |Tue    |
+------------------+-------------------------+----------------------------------------------+------+------------+-------+
only showing top 5 rows
```

Took 2 seconds

FINISHED

```
nasa_logs_df.printSchema()
```

```
root
 |-- host: string (nullable = true)
 |-- timestamp: string (nullable = true)
 |-- path: string (nullable = true)
 |-- status: integer (nullable = false)
 |-- content_size: integer (nullable = false)
 |-- weekday: string (nullable = true)
```

Took 1 seconds

FINISHED

```
var content_size_summary_df = nasa_logs_df.describe(("content_size"))
content_size_summary_df.show()
```

```
content_size_summary_df: org.apache.spark.sql.DataFrame = [summary: string, content_size: string]
+-------+------------------+
|summary|      content_size|
+-------+------------------+
|  count|           1569898|
|   mean|17089.225812122826|
| stddev| 67954.76392157064|
|    min|                 0|
|    max|           3421948|
+-------+------------------+
```

Took 9 seconds

FINISHED

```
// Hits per day.. Volume of incoming requests by day
var hit_by_day = nasa_logs_df.groupBy( ("weekday") ).count()
```

```
hit_by_day: org.apache.spark.sql.DataFrame = [weekday: string, count: bigint]
```

Took 1 seconds (outdated)

FINISHED

```
hit_by_day.show( 10 )
```

```
+-------+------+
```

```
|weekday| count|
+-------+------+
|    Tue|278750|
|    Thu|304301|
|    Sat|133666|
|    Sun|134686|
|    Fri|234370|
|    Mon|228276|
|    Wed|255849|
+-------+------+
```

Took 5 seconds

FINISHED

```
// Total bytes served per weekday and sort in descending order
var average_bytes_served_by_day = nasa_logs_df.groupBy( "weekday" ).agg( sum("content_size").alias("totalbytes") ).sort( desc( "totalbytes" ) )
```

```
average_bytes_served_by_day: org.apache.spark.sql.DataFrame = [weekday: string, totalbytes: bigint]
```

Took 1 seconds (outdated)

FINISHED

```
average_bytes_served_by_day.show()
```

```
+-------+----------+
|weekday|totalbytes|
+-------+----------+
|    Thu|5045134840|
|    Tue|4679266066|
|    Wed|4217550565|
|    Fri|4087947003|
|    Mon|3803970385|
|    Sun|2516940002|
|    Sat|2477532563|
+-------+----------+
```

Took 4 seconds

FINISHED

```
%md

## Participants Exercises
* Define an udf function to find out if the request was successful. If status = 200, it is OK. Otherwise ERROR.
* Add one more column to dataframe showing if the request was OK or ERROR
```

```
* Find out which page has maximum error rate
* Find top 10 pages which encountered errors
* Which are top 10 hosts by count of page access sorted in descending order
* Find out the frequency of different error codes
```

Participants Exercises

- Define an udf function to find out if the request was successful. If status = 200, it is OK. Otherwise ERROR.
- Add one more column to dataframe showing if the request was OK or ERROR
- Find out which page has maximum error rate
- Find top 10 pages which encountered errors
- Which are top 10 hosts by count of page access sorted in descending order
- Find out the frequency of different error codes

Took 0 seconds

READY