



## ΠΡΟΧΩΡΗΜΕΝΑ ΘΕΜΑΤΑ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

Γκίκας Άγγελος el18218

Σκουρτσή Δήμητρα-Άννα el18044

### Ερώτημα 1:

Αρχικά δημιουργήσαμε τα μηχανήματα μας στην πλατφόρμα του okanos και τα κάναμε set-up σύμφωνα με τις οδηγίες που μας δόθηκαν.

okanos KNOSSOS

machines

New Machine +

icon list single

**master**  
[advancedDB.dblab.ntu...]   
snf-34535.ok-kno.grnetcloud.net  
info disks IPs

**slave**  
[advancedDB.dblab.ntu...]   
snf-34536.ok-kno.grnetcloud.net  
info disks IPs

Running

Running

okanos KNOSSOS

IP addresses

New IP Address +

**IP**

**83.212.80.104**  
[advancedDB.dblab....]   
master  
MAC: aa:00:02:c2:98:b5

In use - Running

localnet 192.168.0.0/24

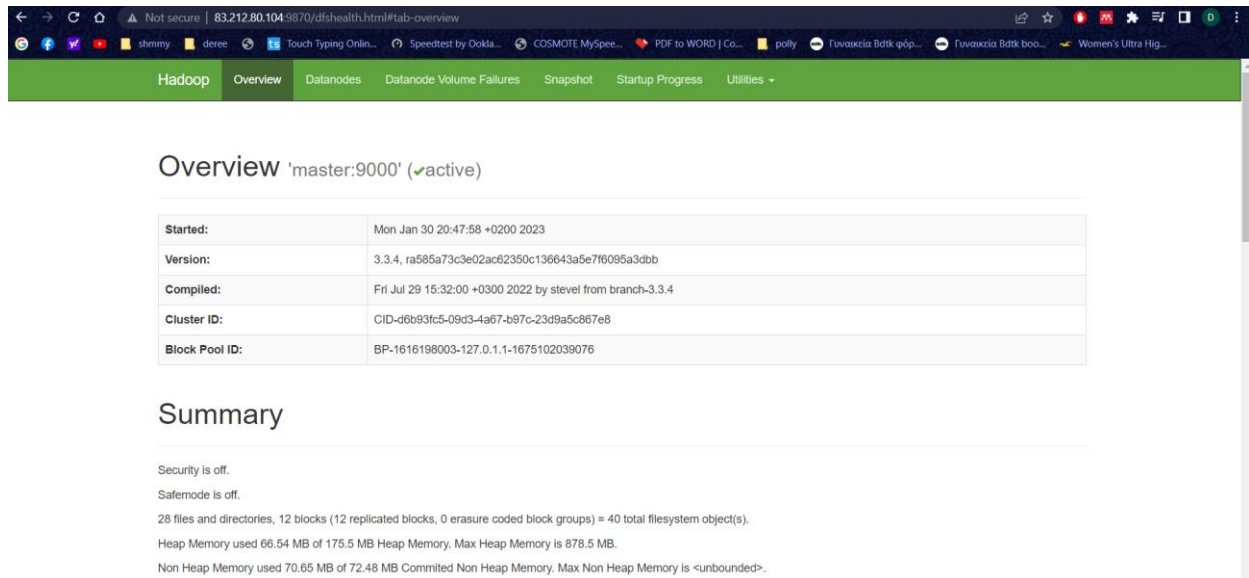
[advancedDB.dblab....]

Connections (2)

**slave**  
IPv4 192.168.0.1

**master**  
IPv4 192.168.0.2

Για την εγκατάσταση της πλατφόρμας εκτέλεσης Spark&HDFS ακολουθήσαμε αρχικά τον οδηγό εγκατάστασης του Hadoop <https://sparkbyexamples.com/hadoop/apache-hadoop-installation/> και στα δυο μηχανήματα, κάναμε format του hdfs δίσκου και εκκινήσαμε το NameNode στο master και το DataNode στο slave. Το Web UI του Hadoop μπορούμε να το δούμε στον σύνδεσμο με το Public IP μας και port 9870 <http://83.212.80.104:9870/>



**Hadoop Overview 'master:9000' (active)**

Started:	Mon Jan 30 20:47:58 +0200 2023
Version:	3.3.4, ra585a73c3e02ac62350c136643a5e7f6095a3dbb
Compiled:	Fri Jul 29 15:32:00 +0300 2022 by stevel from branch-3.3.4
Cluster ID:	CID-d6b93fc5-09d3-4a67-b97c-23d9a5c867e8
Block Pool ID:	BP-1616198003-127.0.1.1-1675102039076

**Summary**

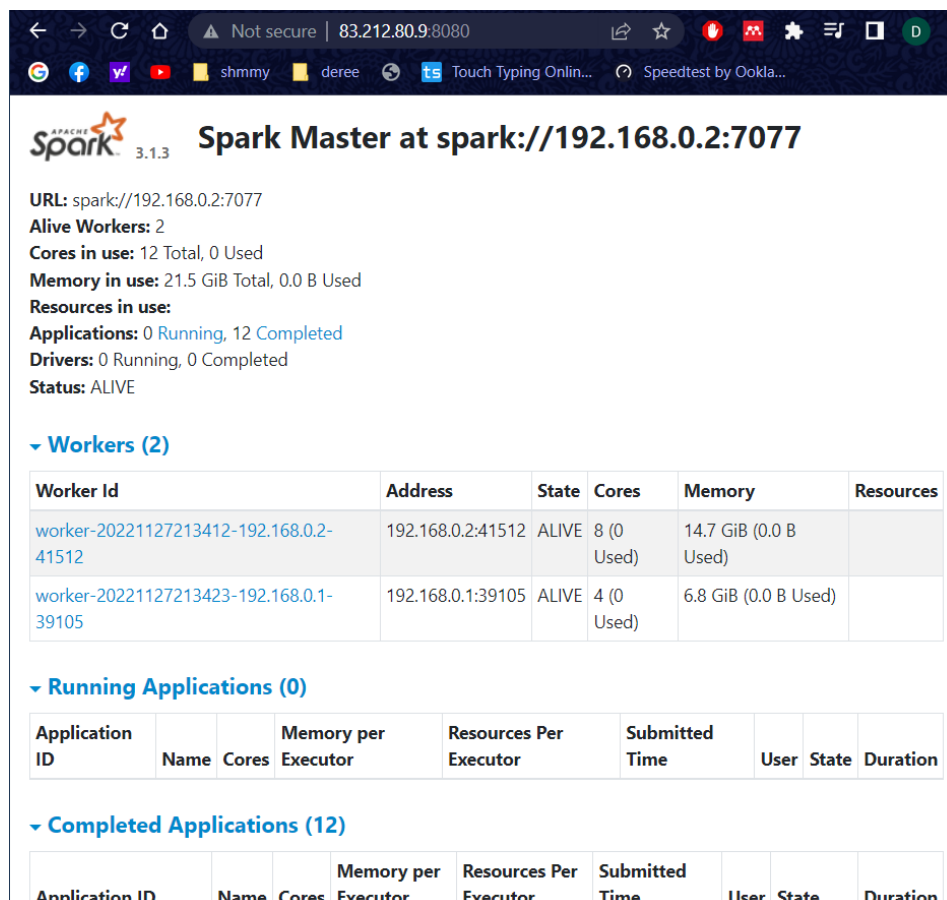
Security is off.  
Safemode is off.

28 files and directories, 12 blocks (12 replicated blocks, 0 erasure coded block groups) = 40 total filesystem object(s).

Heap Memory used 66.54 MB of 175.5 MB Heap Memory. Max Heap Memory is 878.5 MB.

Non Heap Memory used 70.65 MB of 72.48 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Στη συνέχεια το pdf Spark\_Install\_instructions.pdf που δόθηκε στα πλαίσια του μαθήματος και κάναμε deploy ένα worker στον master και έναν στον slave που αργότερα θα τον σταματήσουμε ανάλογα τα ζητούμενα του κάθε ερωτήματος. Το Web UI του Spark μπορούμε να το δούμε στον σύνδεσμο με το Public IP μας και port 8080 <http://83.212.80.104:8080/>



**Spark Master at spark://192.168.0.2:7077**

URL: spark://192.168.0.2:7077

**Alive Workers: 2**

**Cores in use:** 12 Total, 0 Used

**Memory in use:** 21.5 GiB Total, 0.0 B Used

**Resources in use:**

**Applications:** 0 Running, 12 Completed

**Drivers:** 0 Running, 0 Completed

**Status:** ALIVE

**Workers (2)**

Worker Id	Address	State	Cores	Memory	Resources
worker-20221127213412-192.168.0.2-41512	192.168.0.2:41512	ALIVE	8 (0 Used)	14.7 GiB (0.0 B Used)	
worker-20221127213423-192.168.0.1-39105	192.168.0.1:39105	ALIVE	4 (0 Used)	6.8 GiB (0.0 B Used)	

**Running Applications (0)**

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	------------------------	----------------	------	-------	----------

**Completed Applications (12)**

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	------------------------	----------------	------	-------	----------

Για την εκπόνηση της εργασίας δημιουργήσαμε ένα script pyhton όπου αρχικά φορτώσαμε τα αρχεία δημιουργώντας ένα dataframe και ένα rdd για κάθε αρχείο. Για την περίπτωση του tripdata παρατηρήσαμε ότι περιείχε και ορισμένες ημερομηνίες εκτός του χρονικού διαστήματος που μας απασχολεί οπότε και τις αφαιρέσαμε.

```
#Create tripdata dfs
jan = spark.read.parquet("hdfs://master:9000/user/user/project/yellow_tripdata_2022-01.parquet")
feb = spark.read.parquet("hdfs://master:9000/user/user/project/yellow_tripdata_2022-02.parquet")
mar = spark.read.parquet("hdfs://master:9000/user/user/project/yellow_tripdata_2022-03.parquet")
apr = spark.read.parquet("hdfs://master:9000/user/user/project/yellow_tripdata_2022-04.parquet")
may = spark.read.parquet("hdfs://master:9000/user/user/project/yellow_tripdata_2022-05.parquet")
jun = spark.read.parquet("hdfs://master:9000/user/user/project/yellow_tripdata_2022-06.parquet")
dfs = [jan,feb,mar,apr,may,jun]
tripdata = reduce(DataFrame.unionAll, dfs)
#notices some false data so we cleaned the dataframe
tripdata = tripdata.where(tripdata.tpep_pickup_datetime >= "2022-01-01").where(tripdata.tpep_pickup_datetime < "2022-07-01")
tripdata.printSchema()
tripdata.show()
tripdata_rdd = tripdata.rdd

#create location lookup dfs
locations = spark.read.load("hdfs://master:9000/user/user/project/taxi+_zone_lookup.csv", format="csv", inferSchema="true", header="true")
locations.printSchema()
locations.show()
locations_rdd = locations.rdd
```

Οι χρόνοι για όλα τα queries παρουσιάζονται στο τέλος της αναφοράς.

Ερώτημα 2:

Το αποτέλεσμα του Q1 είναι η ακόλουθη γραμμή του πίνακα tripdata.

VendorID	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance	RatecodeID	store_and_fwd_flag	PULocationID	DOLocationID	payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount	improvement_surcharge	total_amount	congestion_surcharge	airport_fee
2	17/3/2022 12:27	17/3/2022 12:27	10	0	10	N	12	12	1	25	0	5	400	0	3	458	25	0

Τα αποτελέσματα του Q2 είναι οι ακόλουθες γραμμές του πίνακα tripdata:

month	VendorID	tpep_pickup_datetime	tpep_dropoff_datetime	Passenger_count	trip_distance	RatecodeID	store_and_fwd_flag	PULocationID	DOLocationID	Payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount	Improvement_surcharge	total_amount	Congestion_surcharge	Airport_fee
1	1	22/1/2022 11:39	22/1/2022 12:31	10	334	10	Y	70	265	4	880	0	5	0	1933	3	2821	0	0
2	1	18/2/2022 2:33	18/2/2022 2:35	10	13	10	N	265	265	1	30	5	5	1985	950	3	11915	0	0
3	1	11/3/2022 20:08	11/3/2022 20:09	10	0	10	N	265	265	1	25	10	5	480	2357	3	2880	0	0
4	1	29/4/2022 4:31	29/4/2022 4:32	20	0	10	N	249	249	3	30	30	5	0	91187	3	91867	25	0
5	1	21/5/2022 16:47	21/5/2022 17:05	10	24	30	N	239	246	3	315	0	0	0	81375	3	84555	0	0
6	1	12/6/2022 16:51	12/6/2022 17:56	90	220	10	N	142	132	2	675	25	5	0	80009	3	87089	25	0

### Ερώτημα 3:

Τα αποτελέσματα του Q3 είναι τα ακόλουθα:

start	end	average amount	average distance
1/1/2022 0:00	16/1/2022 0:00	19,90370264	5,576410378
16/1/2022 0:00	31/1/2022 0:00	19,03660791	4,804840472
31/1/2022 0:00	15/2/2022 0:00	19,55389133	5,950485845
15/2/2022 0:00	2/3/2022 0:00	20,17207809	6,185767213
2/3/2022 0:00	17/3/2022 0:00	20,69235771	6,60698632
17/3/2022 0:00	1/4/2022 1:00	21,11828731	5,524788048
1/4/2022 1:00	16/4/2022 1:00	21,51324609	5,679221476
16/4/2022 1:00	1/5/2022 1:00	21,43101017	5,800096624
1/5/2022 1:00	16/5/2022 1:00	21,929327	6,25531699
16/5/2022 1:00	31/5/2022 1:00	22,80847294	8,000620246
31/5/2022 1:00	15/6/2022 1:00	22,44434698	6,372734052
15/6/2022 1:00	30/6/2022 1:00	22,35241113	6,15420819
30/6/2022 1:00	15/7/2022 1:00	22,24261084	5,946051674

### Ερώτημα 4:

Τα αποτελέσματα του Q4 είναι τα ακόλουθα:

day	hour	average_passengers	rank
1	0	1,529945651	1
1	1	1,527838567	2
1	2	1,508072619	3
2	0	1,467988771	1
2	1	1,444286792	2
2	2	1,423199399	3
3	0	1,420031388	1
3	1	1,417512474	2
3	2	1,410452081	3
4	1	1,408848021	1
4	0	1,401229186	2
4	2	1,401148965	3
5	23	1,405382315	1
5	1	1,402590729	2
5	0	1,401038253	3
6	23	1,475576918	1
6	22	1,444813976	2
6	2	1,423058114	3
7	23	1,522606766	1
7	22	1,506817619	2
7	0	1,499315428	3

Τα αποτελέσματα του Q5 είναι τα ακόλουθα:

month	day	tip_percentange	rank
1	9/1/2022	45,78674775	1
1	31/1/2022	43,93563581	2
1	1/1/2022	29,07803686	3
1	29/1/2022	24,05951845	4
1	16/1/2022	23,37729992	5
2	21/2/2022	25,98165745	1
2	13/2/2022	24,57206839	2
2	9/2/2022	23,90453564	3
2	10/2/2022	23,3396159	4
2	27/2/2022	23,30067995	5
3	18/3/2022	29,67134161	1
3	21/3/2022	27,57992602	2
3	26/3/2022	22,70884595	3
3	5/3/2022	22,55546137	4
3	12/3/2022	22,10085911	5
4	12/4/2022	48,3688441	1
4	2/4/2022	31,17509288	2
4	21/4/2022	30,4486125	3
4	3/4/2022	24,4637277	4
4	30/4/2022	21,99676966	5
5	12/5/2022	32,40265897	1
5	20/5/2022	26,03403609	2
5	16/5/2022	23,65911079	3
5	15/5/2022	22,05244525	4
5	6/5/2022	21,83200616	5
6	13/6/2022	38,45136994	1
6	25/6/2022	32,91307329	2
6	10/6/2022	27,39763781	3
6	16/6/2022	25,53497576	4
6	20/6/2022	24,24291459	5

### Χρόνοι Εκτέλεσης:

Για κάθε query υπολογίστηκε ο χρόνος εκτέλεσης του κάνοντας collect το αποτέλεσμα του. Αρχικά εκτελέστηκαν όλα με έναν worker 10 φορές και προέκυψε ο μέσος χρόνος εκτέλεσης του κάθε query και έπειτα με δύο workers ώστε να είναι συγκρίσιμα τα αποτελέσματα.

1 worker		2 workers	
Query	Time	Query	Time
1	6,126370049	1	6,214474
2	64,57377508	2	63,54046
3	1,17319293	3	1,10289
4	8,586959314	4	8,349702
5	0,741154909	5	0,75827