

به نام خدا

شماره دانشجویی: ۹۹۳۱۰۹۸

تهیه کننده: ابراهیم صدیقی

فاز صفر:

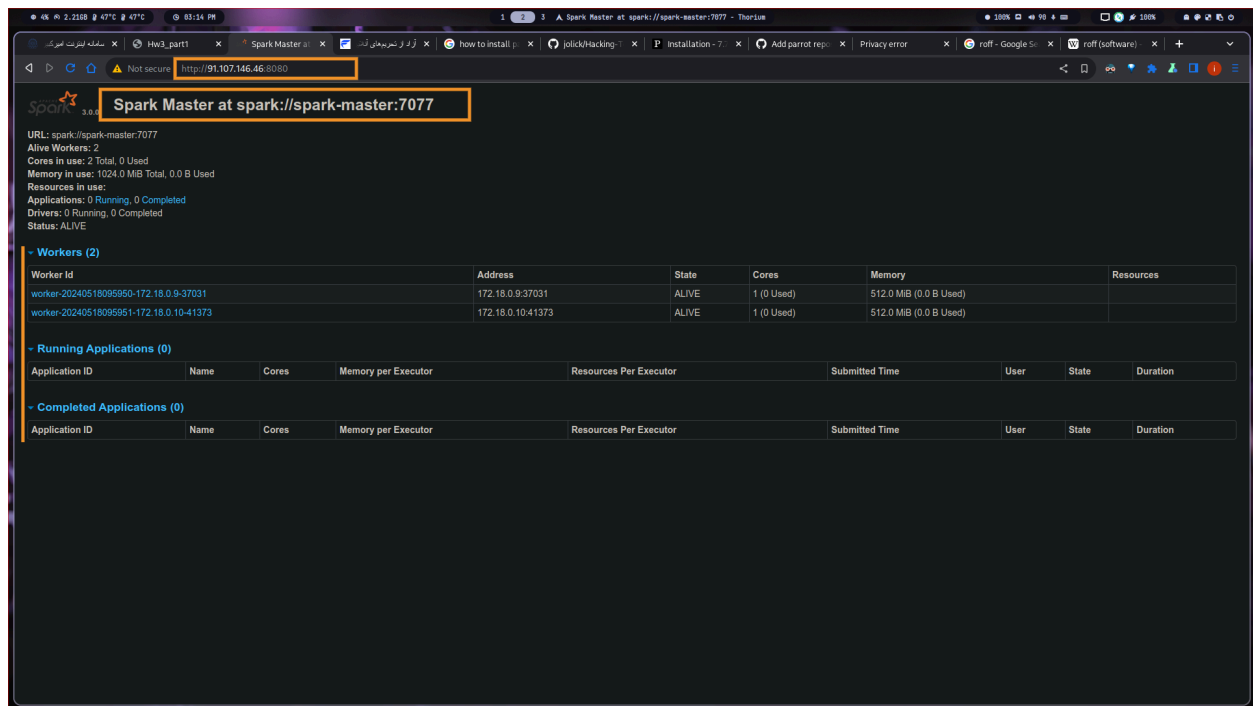
نمایش کانتینر های ایجاد شده

```
root@ubuntu-16g-nbg1-4:~# cd cloud/
root@ubuntu-16g-nbg1-4:~/cloud# ll
total 12
drwxr-xr-x 3 root root 4096 May 18 09:34 ./
drwx----- 9 root root 4096 May 18 09:49 ../
root@ubuntu-16g-nbg1-4:~/cloud# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
hind-jupyter-notebook   latest             9a245fa9a8c6       2 hours ago        2.5GB
hind-hadoop-namenode    latest             891f8b4b42f3       2 hours ago        3.69GB
hind-hadoop-historyserver latest             a88a3790f0ef       2 hours ago        3.69GB
hind-hadoop-datanode    latest             44ae13b2d4f4       2 hours ago        3.69GB
hind-hadoop-nodemanager-1 latest             ee3d5c9b28e5       2 hours ago        3.69GB
hind-hadoop-resourcemanager latest             e212752d472d       2 hours ago        3.69GB
spark-base             latest             86d81a4e3211       2 hours ago        1.66GB
hind-spark-master       latest             f4ae124c1397       2 hours ago        1.66GB
hind-spark-worker-1     latest             e8df6b0b8829       2 hours ago        1.66GB
hind-spark-worker-2     latest             399d5f7b0033       2 hours ago        1.66GB
hadoop-base            latest             604a714b6bc6       2 hours ago        3.69GB
root@ubuntu-16g-nbg1-4:~/cloud#
```

```
root@ubuntu-16g-nbg1-4:~# docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED    STATUS    PORTS
79c6d02f4f38  hadoop-base  "/entrypoint.sh /bin..." 26 hours ago  Exited (0) 26 hours ago
4f18f4803b05  hadoop-base  "/entrypoint.sh"          26 hours ago  Exited (0) 26 hours ago
8ec0320611fd  hadoop-base  "/entrypoint.sh -bash"    26 hours ago  Exited (2) 26 hours ago
3bc47ed18371  hind-spark-worker-2  "/bin/sh -c 'bin/spa..." 30 hours ago  Up 5 hours
383cd1d362b8  hind-spark-worker-1  "/bin/sh -c 'bin/spa..." 30 hours ago  Up 5 hours
1923a488fcd8  hind-hadoop-resourcemanager  "/entrypoint.sh /run..." 30 hours ago  Up 26 hours (healthy)
5494031bcb8b  hind-hadoop-namenode  "/entrypoint.sh /run..." 30 hours ago  Up 26 hours (healthy)
bdc8aed17a70  hind-spark-master  "/bin/sh -c 'bin/spa..." 30 hours ago  Up 5 hours
ad3f13225577  hind-jupyter-notebook  "/bin/sh -c 'jupyter..." 30 hours ago  Up 5 hours
83426a247567  hind-hadoop-historyserver  "/entrypoint.sh /run..." 30 hours ago  Up 26 hours (healthy)
3bde3b933alc  hind-hadoop-nodemanager-1  "/entrypoint.sh /run..." 30 hours ago  Up 26 hours (healthy)
7ac292378908  hind-hadoop-datanode  "/entrypoint.sh /run..." 30 hours ago  Up 26 hours (healthy)
root@ubuntu-16g-nbg1-4:~#
```

در بخش توضیحات کانتینر های ایجاد شده، دو کانتینر ورکر اسپارک با شماره های یک و دو برای انجام فعالیت ساخته شده، یک کانتینر برای مدیریت هدوپ با نام hind-hadoop-resourcemanager و یک کانتینر برای namenode و یکی برای مستر اسپارک یکی برای ژوپیتر و یکی برای تاریخچه هدوپ، یکی برای نود منیجر و مدیریت نود های هدوپ و در آخر یک کانتینر برای دیتا نود هدوپ

نمایش UI برای Hadoop و Spark و Jupyter:



Spark Master at spark://spark-master:7077

URL: spark://spark-master:7077
Alive Workers: 2
Cores in use: 2 Total: 0 Used
Memory in use: 1024.0 MB Total: 0.0 B Used
Resources in use:
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (2)

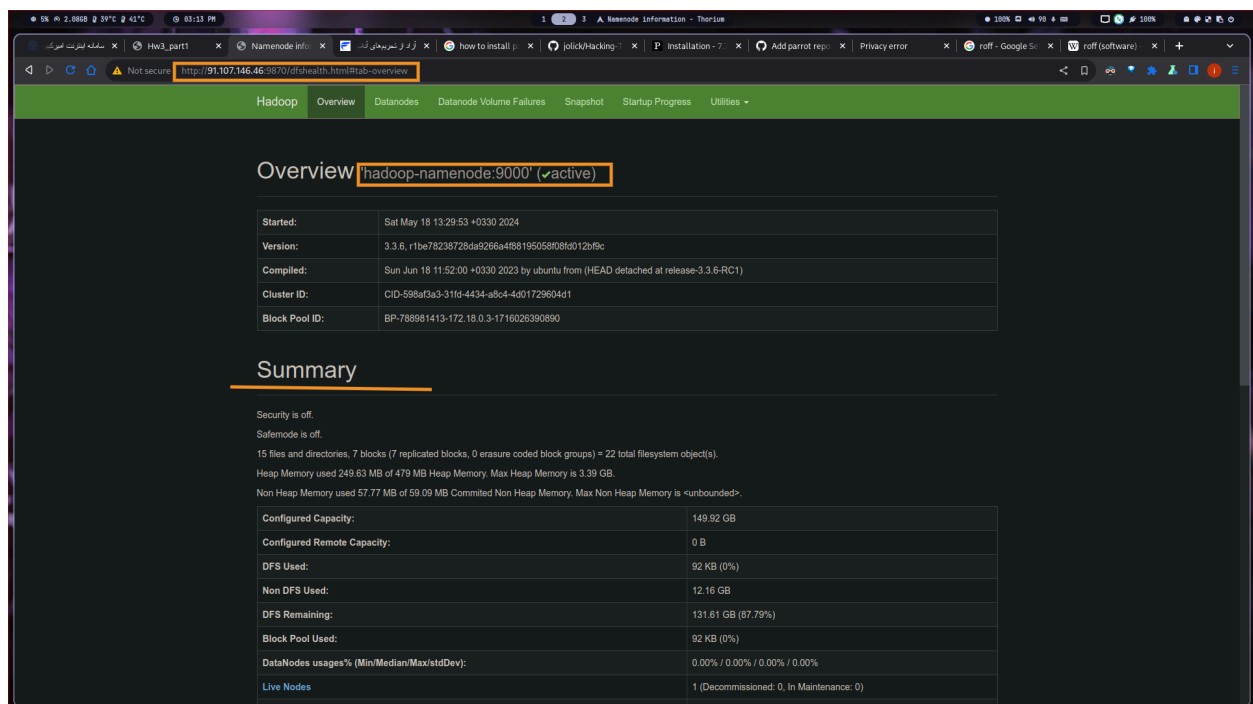
Worker Id	Address	State	Cores	Memory	Resources
worker-20240518095950-172.18.0.9-37031	172.18.0.9:37031	ALIVE	1 (0 Used)	512.0 MB (0.0 B Used)	
worker-20240518095951-172.18.0.10-41373	172.18.0.10:41373	ALIVE	1 (0 Used)	512.0 MB (0.0 B Used)	

Running Applications (0)

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

Completed Applications (0)

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



Overview hadoop-namenode:9000* (active)

Started:	Sat May 18 13:29:53 +0330 2024
Version:	3.3.6, r1be78238728da9266a48819505808d012b9c
Compiled:	Sun Jun 18 11:52:00 +0330 2023 by ubuntu from (HEAD detached at release-3.3.6-RC1)
Cluster ID:	CID-598af3a3-33f4-4434-a8c4-4d01729604d1
Block Pool ID:	BP-788981413-172.18.0.3-1716026390890

Summary

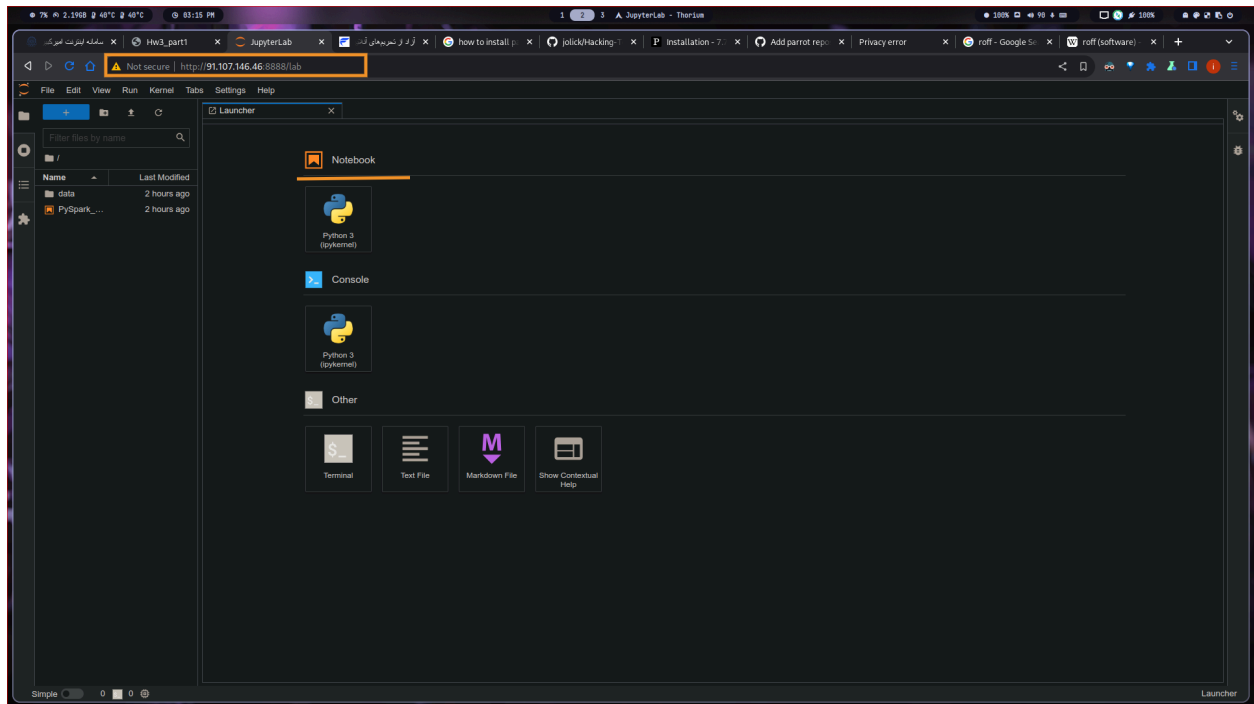
Security is off.
Safemode is off.

15 files and directories, 7 blocks (7 replicated blocks, 0 erasure coded block groups) = 22 total filesystem object(s).

Heap Memory used 249.63 MB of 479 MB Heap Memory. Max Heap Memory is 3.39 GB.

Non Heap Memory used 57.77 MB of 59.00 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	149.92 GB
Configured Remote Capacity:	0 B
DFS Used:	92 KB (0%)
Non DFS Used:	12.16 GB
DFS Remaining:	131.61 GB (87.79%)
Block Pool Used:	92 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	1 (Decommissioned: 0, In Maintenance: 0)



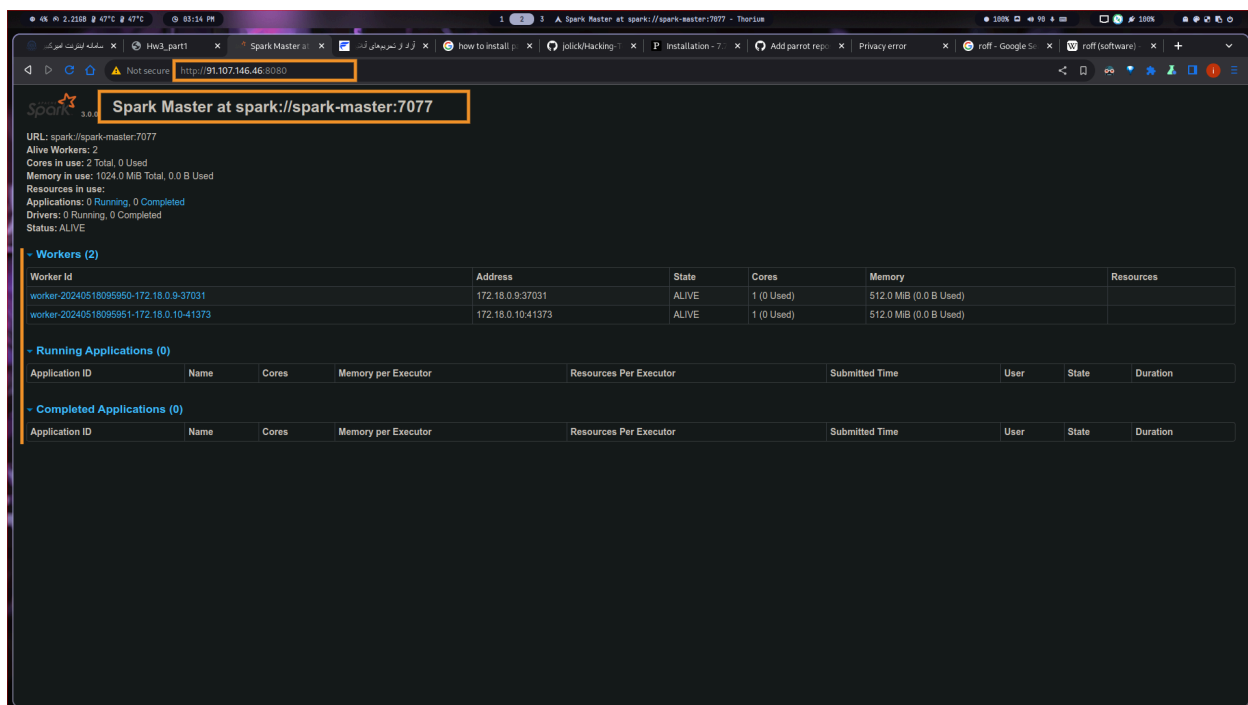
در بخش زیر اطلاعات مربوط به namenode و فایل سیستم قرار دارد.

Overview	
hadoop-namenode:9000 (active)	
Started:	Sat May 18 13:29:53 +0330 2024
Version:	3.3.6, r1be78238728da9266a45819505808d012bfc
Compiled:	Sun Jun 18 11:52:00 +0330 2023 by ubuntu from (HEAD detached at release-3.3.6-RC1)
Cluster ID:	CID-598af3a3-31fd-4434-a6c4-4d01729604d1
Block Pool ID:	BP-788981413-172.18.0.3-1716026390890

Summary	
Security is off.	
Safemode is off.	
15 files and directories, 7 blocks (7 replicated blocks, 0 erasure coded block groups) = 22 total filesystem object(s).	
Heap Memory used 249.63 MB of 479 MB Heap Memory. Max Heap Memory is 3.39 GB.	
Non Heap Memory used 57.77 MB of 59.09 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.	
Configured Capacity:	149.92 GB
Configured Remote Capacity:	0 B
DFS Used:	92 KB (0%)
Non DFS Used:	12.16 GB
DFS Remaining:	131.61 GB (87.79%)
Block Pool Used:	92 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	1 (Decommissioned: 0, In Maintenance: 0)

که طبق این تصویر ۲۲ فایل سیستم دارد و اطلاعات تکمیلی آن در تصاویر قابل مشاهده است

توضیحات مربوط به تعداد نودهای اسپارک و منابع استفاده شده:



Spark Master at spark://spark-master:7077

URL: spark://spark-master:7077
Alive Workers: 2
Cores in use: 2 Total, 0 Used
Memory in use: 1024.0 MB Total, 0.0 B Used
Resources in use:
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (2)

Worker Id	Address	State	Cores	Memory	Resources
worker-20240518095950-172.18.0.9-37031	172.18.0.9:37031	ALIVE	1 (0 Used)	512.0 MB (0.0 B Used)	
worker-20240518095951-172.18.0.10-41373	172.18.0.10:41373	ALIVE	1 (0 Used)	512.0 MB (0.0 B Used)	

Running Applications (0)

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

Completed Applications (0)

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

مشاهده می شود که دو نود دارد و مموری ها نیز قابل مشاهده هستند که چون اکنون کاری انجام نمی دهند مقدار صفر از آنها استفاده شده است.

فاز یک:

توضیح mapper:

1. ورودی: هر خط از ورودی استاندارد (sys.stdin) دریافت می‌شود.
2. جداسازی اسناد: هر خط به دو بخش doc_id (شناسه سند) و context (متن سند) بر اساس کاما تقسیم می‌شود.
3. تقسیم کلمات: متن سند (context) به کلمات جداگانه تقسیم می‌شود.
4. خروجی نگاشت: برای هر کلمه، زوج <word>\t<docid> را به خروجی استاندارد چاپ می‌کند.

توضیح reducer:

1. ورودی: هر خط از ورودی استاندارد (sys.stdin) دریافت می‌شود.
2. ساختار داده ها: هر خط به دو بخش word (کلمه) و doc_id (شناسه سند) بر اساس تب (\t) تقسیم می‌شود.
3. ساخت شاخص معکوس: با استفاده از defaultdict از ماژول collections، کلمات به لیستی از شناسه های اسناد نگاشت می‌شوند.
4. خروجی کاهش: برای هر کلمه، کلمه و لیست شناسه های اسناد حاوی آن کلمه به صورت <word>\t<id> چاپ می‌شود.

فایل ها:

```
#!/usr/bin/env python
import sys

for line in sys.stdin:
    doc_id, context = line.strip().split(',')
    words = context.split()

    for word in words:
        print(f"{word}\t{doc_id}")
```

```
#!/usr/bin/env python
from collections import defaultdict
import sys

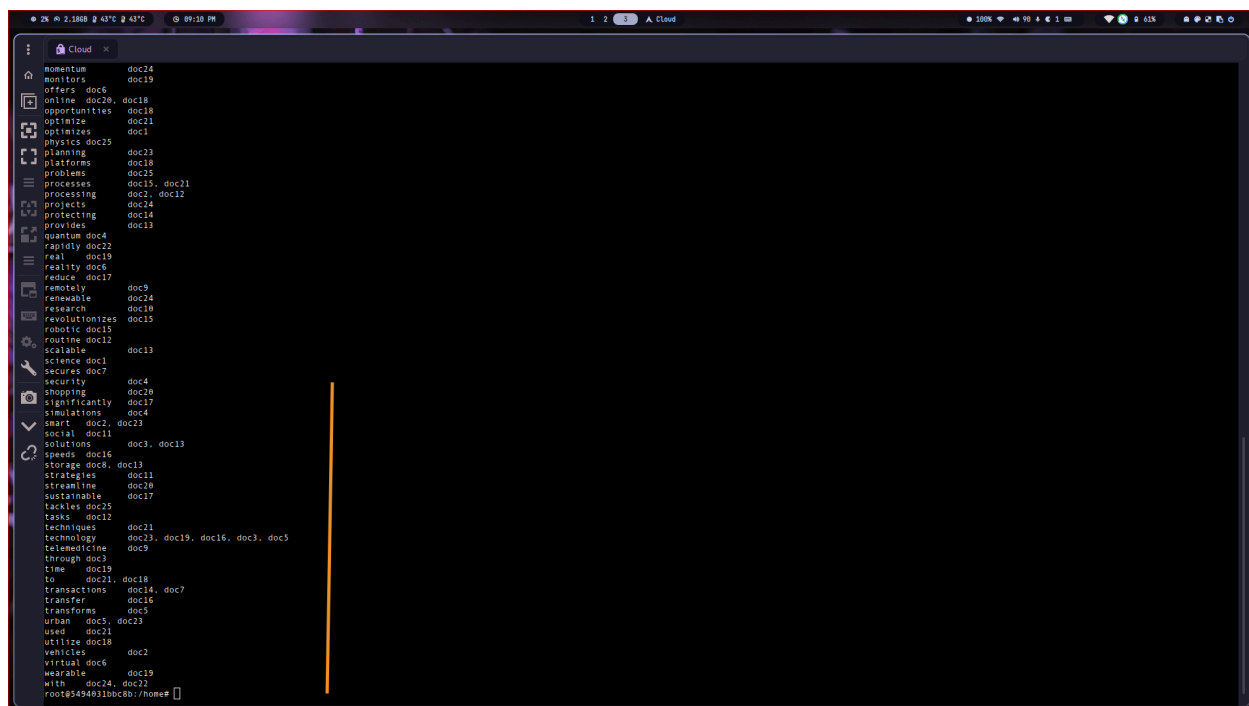
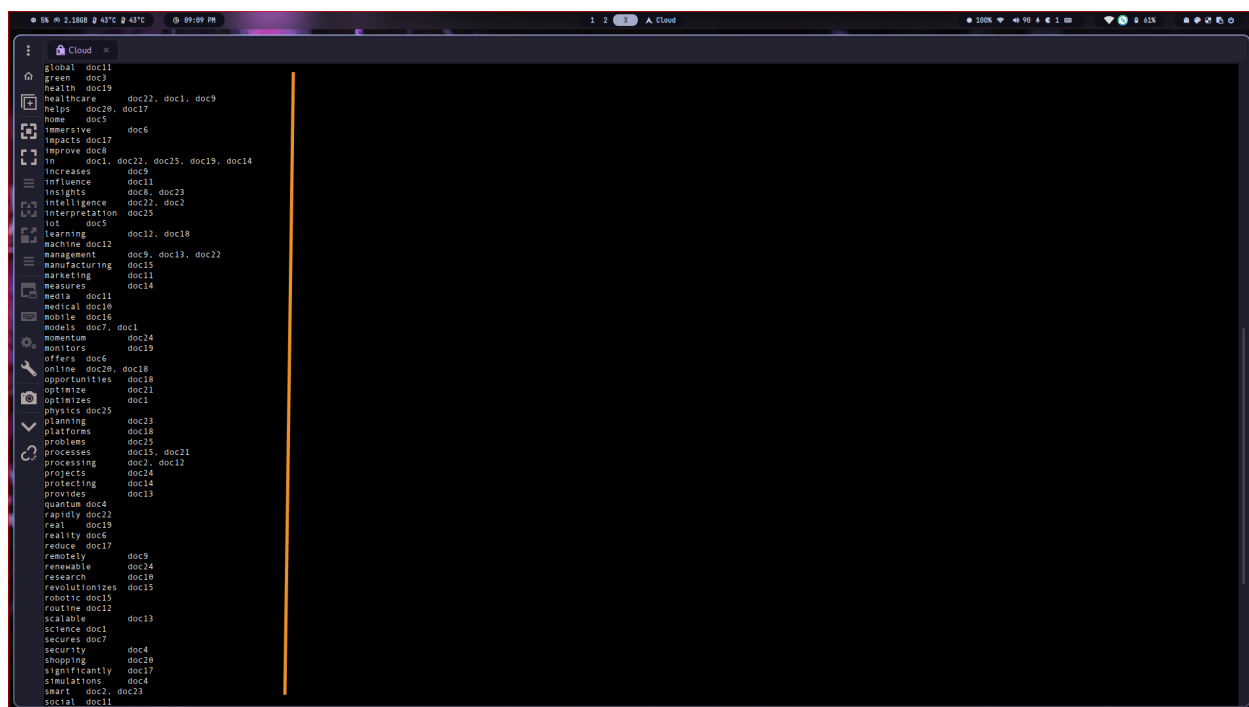
inverted_index = defaultdict(list)

for line in sys.stdin:
    word, doc_id = line.strip().split('\t')
    inverted_index[word].append(doc_id)

for word, doc_ids in inverted_index.items():
    print(f"{word}\t{' '.join(doc_ids)}")
```

خروجی ها:

```
root@549401bbc8b:/home# hdfs dfs -ls hdfs://hadoop-namenode:9000/user/root/output/
Found 2 items
-rw-r--r-- 3 root supergroup          0 2024-05-18 17:36 hdfs://hadoop-namenode:9000/user/root/output/ SUCCESS
-rw-r--r-- 3 root supergroup 2644 2024-05-18 17:36 hdfs://hadoop-namenode:9000/user/root/output/part-00000
root@549401bbc8b:/home# hdfs dfs -cat hdfs://hadoop-namenode:9000/user/root/output/part-00000
5g doc16
accessibility doc9
advanced doc25
advancements doc22
advances doc18
agricultural doc18
analysis doc3, doc20, doc21
analytics doc18, doc11, doc8, doc1
and doc5, doc1, doc13, doc7, doc12, doc8, doc6, doc9, doc4, doc10, doc2, doc19, doc16, doc25, doc23, doc20, doc21
are doc14, doc21
artificial doc22, doc2
automates doc12
automation doc15
autonomous doc2
behavior doc8
benefits doc23
big doc8
blockchain doc7
business doc21
change doc3
climate doc1
cloud doc13
collection doc12
combat doc3
complex doc4, doc25
computing doc25, doc13, doc4
connectivity doc16
consumer doc9
critical doc14
cryptocurrency doc7
cybersecurity doc14
data doc20, doc25, doc14, doc24, doc15, doc23, doc16, doc16, doc17, doc18, doc22, doc21, doc21, doc20, doc19, doc7, doc10, doc5, doc4, doc11, doc3, doc9, doc6, doc2, doc7, doc8, doc12, doc12, doc13, doc1, doc1, doc8, doc11
decisions doc24
digital doc14
driven doc24, doc15, doc6
drives doc2
e-commerce doc20
education doc18
educational doc6
energy doc24
engineering doc19
enhances doc7, doc4, doc16
environmental doc17
environments doc3
evolves doc22
expand doc18
experiences doc20, doc6
farming doc17
financial doc1
fitness doc19
flow doc21
from doc23, doc10
gain doc24
genetic doc10
global doc11
green doc3
health doc10
```



در تصاویر بالا نتایج اجرا برای بخش اول پروژه را مشاهده می کنید.

بخش امتیازی

خروجی: برای $k=3$

```
Last login: Sun May 19 13:23:17 2024 from 212.80.13.125
root@ubuntu-16gb-nbg1-4:~# docker exec -it hadoop-namenode /bin/bash
root@5494031bbc8b:/# hdfs dfs -cat hdfs://hadoop-namenode:9000/user/root/output/part-00000
Document: doc3
automation      2
intelligence    2
AI              1
Document: doc8
Architectural  1
and            1
designs         1
Document: doc4
security        2
Blockchain      1
and            1
Document: doc9
analytics       2
Data           1
and            1
Document: doc6
digital         2
Education       1
and            1
Document: doc2
is              3
sustainability  3
Environmental   1
Document: doc7
in             2
Healthcare     1
and            1
Document: doc1
in             3
technology      3
innovation      2
Document: doc10
and            2
Public         1
a              1
Document: doc5
energy         5
and            2
Renewable      1
root@5494031bbc8b:/#
```

توضیح mapper:

1. از کتابخانه‌های `sys` و `defaultdict` از `collections` استفاده می‌کند.
2. برای هر خط ورودی از `sys.stdin` (که شامل یک `doc_id` و `context` است) داده‌ها را می‌خواند.

3. با استفاده از strip() خط را تمیز کرده و سپس با استفاده از split(',') آن را به doc_id و context تقسیم می‌کند.
4. متن context را به لیستی از کلمات جدا می‌کند.
5. با استفاده از defaultdict(int), تعداد هر کلمه در متن را شمارش می‌کند.
6. برای هر کلمه و تعداد آن، خروجی را به شکل <word>\t<count>\t<doc_id> چاپ می‌کند.

توضیح reducer:

1. از کتابخانه‌های sys, defaultdict, و heapq استفاده می‌کند.
2. برای هر خط ورودی از sys.stdin داده‌ها را به شکل <word>\t<count>\t<doc_id> دریافت می‌کند.
3. این داده‌ها را با استفاده از strip() تمیز کرده و سپس با '\t' split('') به word, count و doc_id تقسیم می‌کند.
4. مقدار count را به عدد صحیح تبدیل می‌کند.
5. هر کلمه و تعداد آن را به لیستی از کلمات و تعدادهای مربوط به هر doc_id اضافه می‌کند.
6. یک متغیر K را تعیین می‌کند که تعداد پرکاربردترین کلمات را مشخص می‌کند (در اینجا ۳).
7. برای هر سند (doc_id)، لیست کلمات و تعدادها را می‌گیرد و با استفاده از heapq.nlargest, K کلمه پرکاربرد را پیدا می‌کند.

mapper.py

```
GNU nano 6.2
#!/usr/bin/env python
import sys
from collections import defaultdict

for line in sys.stdin:
    doc_id, context = line.strip().split(',')
    words = context.split()

    word_count = defaultdict(int)
    for word in words:
        word_count[word] += 1

    for word, count in word_count.items():
        print(f"{word}\t{count}\t{doc_id}")
```

reducer.py

```
GNU nano 6.2
#!/usr/bin/env python
from collections import defaultdict
import sys
import heapq

inverted_index = defaultdict(list)

for line in sys.stdin:
    word, count, doc_id = line.strip().split('\t')
    count = int(count)
    inverted_index[doc_id].append((word, count))

K = 3 # Change this value to set the desired number of top words

for doc_id, word_counts in inverted_index.items():
    top_k_words = heapq.nlargest(K, word_counts, key=lambda x: x[1])
    print(f"Document: {doc_id}")
    for word, count in top_k_words:
        print(f"{word}\t{count}")
    print()
```