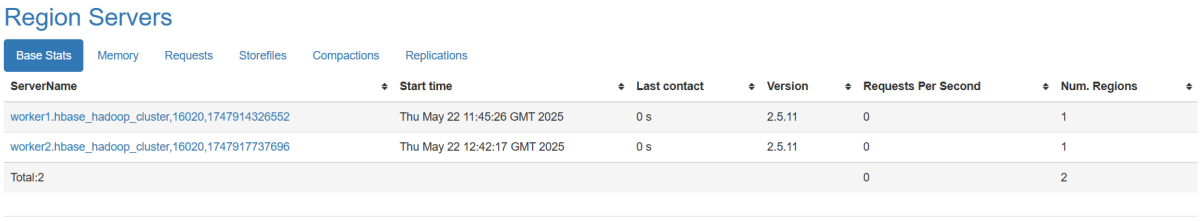
Project Doc

First HBASE Image:

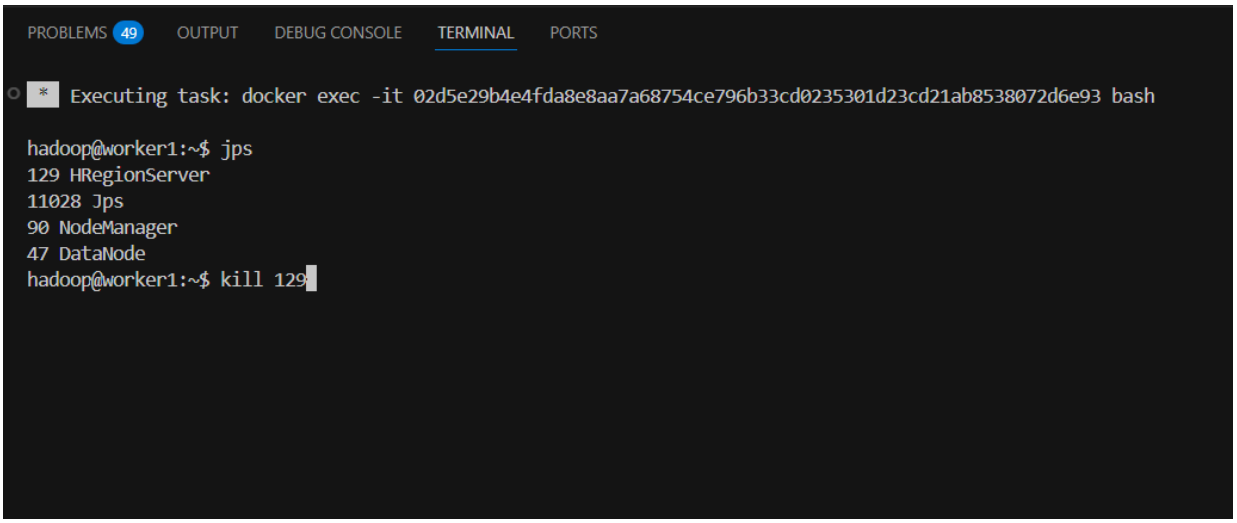
- First Download HBASE 2.5: https://dlcdn.apache.org/hbase/2.5.11/hbase-2.5.11-bin.tar.gz
- **Export Env Vars and Create HBASE Root Dir on HDFS**
  - `export HBASE_HOME=/data/hbase && export PATH=$HBASE_HOME/bin:$PATH`
  - `hdfs dfs -mkdir -p /hbase && hdfs dfs -chown hadoop:hadoop /hbase`
- copy hdfs-site.xml to the container
- doing this as multi-stage on hadoop image

Failover Tests :

- this is my active region servers



- now lets turn off one of them

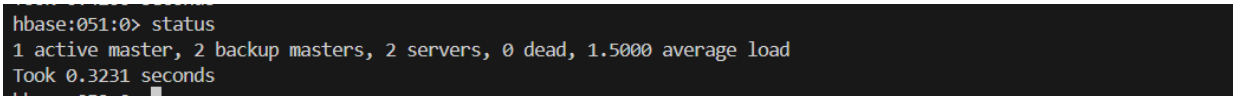


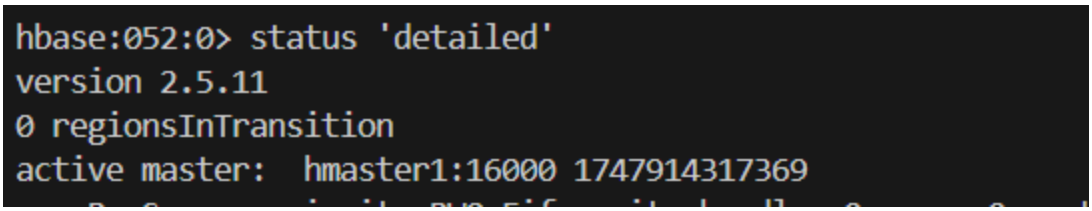
- now there is one region server , and has all regions bcs load balancer redirected the region to another region server

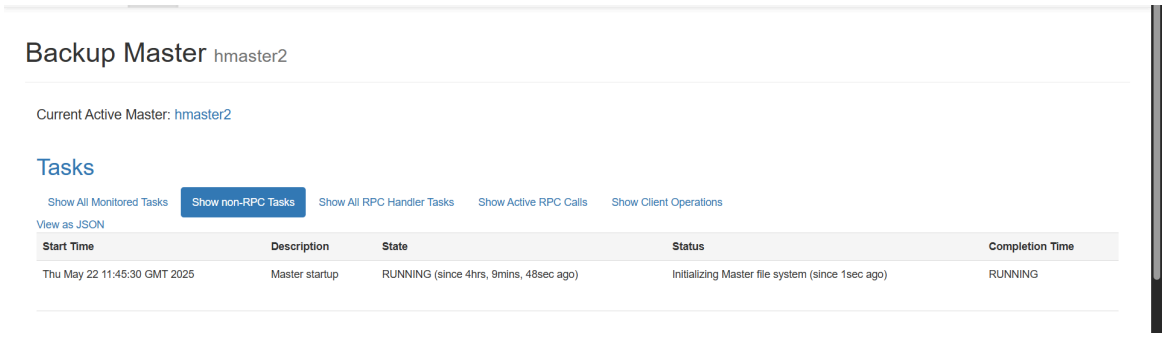
- now let's active it using this command `hbase-daemon.sh start regionserver`

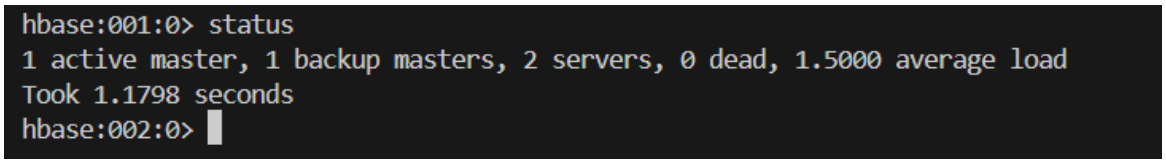
  - now its up and load balancer re-assigned the region to it

HMASTER Failover Test:

- i have 1 active and 2 backups
  - 

  - 

  - HMaster1 is the Active Master
  - now lets turn off it using kill command or turning off the container
  - now my active is HMASER 2 :
    - 

    - 

      - we can notice that i have 1 active and 1 backup and 1 dead
  - now let's turn the container on:

Backup Masters

ServerName	Port	Start Time
hmaster1	16000	Thu May 22 15:59:42 GMT 2025
hmaster3	16000	Thu May 22 11:45:17 GMT 2025

Total:2

- we can find it in the backup masters

```
hbase:002:0> status
1 active master, 2 backup masters, 2 servers, 0 dead, 1.5000 average load
Took 0.0344 seconds
```

Web Table:

HBase Table Schema: `webTable`

This HBase table is named `webTable` and is designed for efficient storage and retrieval of web content, metadata, and link structure. The schema uses salting, bloom filters, and finely tuned block settings to optimize performance for web-scale workloads.

🧱 Table Design Overview

Column Families

The table defines **four column families**, each with specific settings tailored for different access patterns:

---

1. `content`

- **Purpose**: Stores the main content of a webpage (e.g., HTML, text).
- **BLOOMFILTER**: `ROW` — optimized for lookups by row key.
- **BLOCKSIZE**: `65536` bytes (64 KB) — suited for larger values like full HTML pages.
- **BLOCKCACHE**: `true` — enables in-memory caching of blocks to improve read speed.
- **IN_MEMORY**: `true` — keeps data in memory for fast access (only viable if memory allows).
- **VERSIONS**: `1` — only the latest version of each cell is retained.

2. `meta`

- **Purpose**: Stores metadata such as titles, headers, timestamps.
- **BLOOMFILTER**: `ROW` — fast row-level lookups.
- **BLOCKSIZE**: `16384` bytes (16 KB) — smaller data blocks for lightweight metadata.
- **BLOCKCACHE**: `true` — metadata is likely to be reused, so caching is beneficial.
- **VERSIONS**: `1` — only the most recent metadata is needed.

---

3. `outlinks`

- **Purpose**: Stores links from the page to others (outbound links).
- **BLOOMFILTER**: `ROWCOL` — optimized for access to specific link values by row and column.
- **BLOCKSIZE**: `32768` bytes (32 KB) — moderate block size for lists of links.
- **VERSIONS**: `1` — only the latest state of links is retained.

---

4. `inlinks`

- **Purpose**: Stores backlinks from other pages (who links to this page).
- **BLOOMFILTER**: `ROWCOL` — enables precise lookup for specific incoming links.
- **BLOCKSIZE**: `32768` bytes (32 KB).
- **VERSIONS**: `1`.

🧂 Region Pre-splitting with Salts

```
['0!', '1!', '2!', '3!', '4!', '5!', '6!', '7!', '8!', '9!', 'a!', 'b!', 'c!',
'd!', 'e!', 'f!']
```

```
create 'webTable',
  {NAME => 'content',
   BLOOMFILTER => 'ROW',
   BLOCKSIZE => 65536,
```

```
   BLOCKCACHE => true,
   IN_MEMORY => true,
   VERSIONS => 3,
   TTL => 7776000},

  {NAME => 'meta',
   BLOOMFILTER => 'ROW',
   BLOCKSIZE => 16384,
   BLOCKCACHE => true,
   VERSIONS => 1,
   TTL => 2147483647},

  {NAME => 'outlinks',
   BLOOMFILTER => 'ROWCOL',
   BLOCKSIZE => 32768,
   VERSIONS => 2,
   TTL => 15552000},

  {NAME => 'inlinks',
   BLOOMFILTER => 'ROWCOL',
   BLOCKSIZE => 32768,
   VERSIONS => 2,
   TTL => 15552000},

  ['0!', '1!', '2!', '3!', '4!', '5!', '6!', '7!', '8!', '9!', 'a!', 'b!',
'c!', 'd!', 'e!', 'f!']
```

- it's better to compress content with SNAPPY but need needs to be downloaded in HBASE lib

---

Why i used this table design:

- our table has heavy read and write but reads are more
- reading web content is much more that creating new content
- so i choose to use bloom filters , it might to delay write a bit but it will help us avoiding searching for data that doesn't exist
- we pre defined salted regions to avoid hotspotting
- we used this salting to help us designing our key

```
e!org.wikipedia.www/page-3
e!org.wikipedia.www/page-3
e!org.wikipedia.www/page-3
e!org.wikipedia.www/page-3

e!org.wikipedia.www/page-3

f!com.facebook.www/page-5

f!com.facebook.www/page-5
```

- this is keys sample , we reverse the web domain bcs all websites starts with www
- salting part , to make websites pages rows stored on the same region and same pages of the same domain stored with salt near to it's domain ,
  - for ex: google.com has salt 'e'
  - google.com/about has salt 'f'
  - but facebook for ex may get salt = 'a'

Queries:

Retrieve a page by exact row key

```
Took 0.0456 seconds
hbase:029:0> get 'webTable', 'e!org.wikipedia.www/page-3'
COLUMN                          CELL
 content:html                   timestamp=2025-05-22T17:24:47.849, value=<html><h1>www.wikipedia.org /page-3</h1></ht
                                ml>
 content:text                   timestamp=2025-05-22T17:24:47.849, value=Welcome to www.wikipedia.org /page-3
 meta:content_type              timestamp=2025-05-22T17:24:47.849, value=text/html
 meta:fetch_time                timestamp=2025-05-22T17:24:47.849, value=1747934687839
 meta:status                    timestamp=2025-05-22T17:24:47.849, value=200
 outlinks:a!org.wikipedia.www/ timestamp=2025-05-22T17:24:47.849, value=Previous Page
 page-2
1 row(s)
Took 0.0132 seconds
hbase:030:0>
```

Scan all pages for a specific domain:

```
hbase:030:0> scan 'webTable', {FILTER => "PrefixFilter('e!org.wikipedia.www')"}
ROW                           COLUMN+CELL
 e!org.wikipedia.www/page-3   column=content:html, timestamp=2025-05-22T17:24:47.849, value=<html><h1>www.wikipedia
                              .org /page-3</h1></html>
 e!org.wikipedia.www/page-3   column=content:text, timestamp=2025-05-22T17:24:47.849, value=Welcome to www.wikipedi
                              a.org /page-3
 e!org.wikipedia.www/page-3   column=meta:content_type, timestamp=2025-05-22T17:24:47.849, value=text/html
 e!org.wikipedia.www/page-3   column=meta:fetch_time, timestamp=2025-05-22T17:24:47.849, value=1747934687839
 e!org.wikipedia.www/page-3   column=meta:status, timestamp=2025-05-22T17:24:47.849, value=200
 e!org.wikipedia.www/page-3   column=outlinks:a!org.wikipedia.www/page-2, timestamp=2025-05-22T17:24:47.849, value=
                              Previous Page
1 row(s)
Took 0.0623 seconds
```

Find pages modified within a time range:

```
hbase:031:0> scan 'webTable', {  FILTER => "SingleColumnValueFilter('meta', 'fetch_time', >=, 'binary:START_TIMESTAM
P') AND " +            "SingleColumnValueFilter('meta', 'fetch_time', <=, 'binary:END_TIMESTAMP')"}
ROW                           COLUMN+CELL
 c0!net.blog.tech/123         column=content:html, timestamp=2025-05-22T17:24:47.493, value=<html><p>Check out <a h
                              ref="https://www.example.com">Example Inc</a></p></html>
 c0!net.blog.tech/123         column=outlinks:com.example.www/, timestamp=2025-05-22T17:24:47.493, value=Example In
                              c
1 row(s)
Took 0.0789 seconds
```

Find inbound links to a specific page:

```
hbase:032:0> scan 'webTable', {FILTER => "QualifierFilter(=, 'substring:e!org.wikipedia.www/page-3')"}
ROW                           COLUMN+CELL
 a!org.wikipedia.www/page-2   column=outlinks:e!org.wikipedia.www/page-3, timestamp=2025-05-22T17:24:47.842, value=
                              Next Page
1 row(s)
Took 0.0456 seconds
```

Find pages containing specific text:

```
hbase:033:0> scan 'webTable', {FILTER => "ValueFilter(=, 'substring:Example')"}
ROW                           COLUMN+CELL
 a0!com.example.www/          column=inlinks:net.blog.tech/123, timestamp=2025-05-22T17:24:47.500, value=Example In
                              c
 c0!net.blog.tech/123         column=content:html, timestamp=2025-05-22T17:24:47.493, value=<html><p>Check out <a h
                              ref="https://www.example.com">Example Inc</a></p></html>
 c0!net.blog.tech/123         column=outlinks:com.example.www/, timestamp=2025-05-22T17:24:47.493, value=Example In
                              c
2 row(s)
Took 0.0249 seconds
```

Insert complete web page data (content, metadata, links):

```
put 'webTable', 'a0!com.example.www/', 'content:html', '<html>...</html>'
put 'webTable', 'a0!com.example.www/', 'meta:fetch_time', '1717020000000'
put 'webTable', 'a0!com.example.www/', 'meta:status', '200'
put 'webTable', 'a0!com.example.www/', 'outlinks:com.example.www/about',
'About Us'
```

```
put 'webTable', 'a0!com.example.www/', 'inlinks:net.blog.tech/123', 'Example
Inc'
```

Retrieve a page by exact URL (row key):

```
hbase:039:0> get 'webTable', 'a0!com.example.www/'
COLUMN                          CELL
 content:html                   timestamp=2025-05-22T17:34:18.587, value=<html>...</html>
 content:text                   timestamp=2025-05-22T17:24:47.477, value=Welcome to our site. About Us.
 inlinks:net.blog.tech/123      timestamp=2025-05-22T17:34:39.154, value=Example Inc
 meta:content_type              timestamp=2025-05-22T17:24:47.477, value=text/html
 meta:fetch_time                timestamp=2025-05-22T17:34:23.326, value=1717020000000
 meta:status                    timestamp=2025-05-22T17:34:31.395, value=200
 outlinks:com.example.www/abou timestamp=2025-05-22T17:34:35.923, value=About Us
 t
1 row(s)
Took 0.0185 seconds
```

Update a page's content and metadata:

```
put 'webTable', 'a0!com.example.www/', 'content:html', '<html><h1>Updated
Content</h1></html>'
put 'webTable', 'a0!com.example.www/', 'meta:fetch_time', '1717030000000'
```

Delete a page and all its information:

```
hbase:042:0> deleteall 'webTable', 'a0!com.example.www/'
Took 0.1167 seconds
hbase:043:0>
```

Find pages with titles containing specific keywords:

```
hbase:043:0> scan 'webTable', {FILTER => "ValueFilter(=, 'substring:Keyword')"}
ROW                             COLUMN+CELL
0 row(s)
Took 0.0129 seconds
```

List pages with specific HTTP status codes:

```
hbase:044:0> scan 'webTable', {FILTER => "SingleColumnValueFilter('meta', 'status', =, 'binary:404')"}
ROW                           COLUMN+CELL
 b0!com.example.www/about     column=content:html, timestamp=2025-05-22T17:24:47.485, value=<html><h2>About</h2><p>
                              Our company info</p></html>
 b0!com.example.www/about     column=inlinks:com.example.www/, timestamp=2025-05-22T17:24:47.485, value=Home
 b0!com.example.www/about     column=meta:fetch_time, timestamp=2025-05-22T17:24:47.485, value=1717020001000
 c0!net.blog.tech/123         column=content:html, timestamp=2025-05-22T17:24:47.493, value=<html><p>Check out <a h
                              ref="https://www.example.com">Example Inc</a></p></html>
 c0!net.blog.tech/123         column=outlinks:com.example.www/, timestamp=2025-05-22T17:24:47.493, value=Example In
                              c
2 row(s)
Took 0.0181 seconds
```

```
hbase:045:0> scan 'webTable', {FILTER => "SingleColumnValueFilter('meta', 'status', >=, 'binary:400')"}
ROW                           COLUMN+CELL
 b0!com.example.www/about     column=content:html, timestamp=2025-05-22T17:24:47.485, value=<html><h2>About</h2><p>
                              Our company info</p></html>
 b0!com.example.www/about     column=inlinks:com.example.www/, timestamp=2025-05-22T17:24:47.485, value=Home
 b0!com.example.www/about     column=meta:fetch_time, timestamp=2025-05-22T17:24:47.485, value=1717020001000
 c0!net.blog.tech/123         column=content:html, timestamp=2025-05-22T17:24:47.493, value=<html><p>Check out <a h
                              ref="https://www.example.com">Example Inc</a></p></html>
 c0!net.blog.tech/123         column=outlinks:com.example.www/, timestamp=2025-05-22T17:24:47.493, value=Example In
                              c
2 row(s)
Took 0.0103 seconds
```

Find pages modified after a specific date:

```
hbase:046:0> scan 'webTable', {FILTER => "SingleColumnValueFilter('meta', 'fetch_time', >, 'binary:TIMESTAMP')"}
ROW                           COLUMN+CELL
 c0!net.blog.tech/123         column=content:html, timestamp=2025-05-22T17:24:47.493, value=<html><p>Check out <a h
                              ref="https://www.example.com">Example Inc</a></p></html>
 c0!net.blog.tech/123         column=outlinks:com.example.www/, timestamp=2025-05-22T17:24:47.493, value=Example In
                              c
1 row(s)
Took 0.0218 seconds
```

Manual purge for outdated content:

```
Took 0.0255 seconds
hbase:050:0> delete 'webTable', 'a0!com.example.www/', 'content:html', 1680000000000  # delete specific timestamp ve
rsion
Took 0.0148 seconds
hbase:051:0> deleteall 'webTable', 'a0!com.example.www/'  # delete entire row
Took 0.0073 seconds
```

Retrieve latest N versions of content:

```
hbase:065:0> get 'webTable', 'a0!com.example.www/', {COLUMN => 'content:html', VERSIONS => 3}
COLUMN                        CELL
0 row(s)
Took 0.0042 seconds
```