

## **ASSIGNMENT - 10**

### **BIG DATA (CSP 554)**

```

SSH-in-browsers
[+] UPLOAD FILE [+] DOWNLOAD FILE [+] [+] [+]
cynnavarapu@a20561894-n2-m:~$ wget https://downloads.apache.org/hbase/2.4.18/hbase-2.4.18-bin.tar.gz
--2024-11-14 21:02:46-- https://downloads.apache.org/hbase/2.4.18/hbase-2.4.18-bin.tar.gz
Resolving downloads.apache.org (downloads.apache.org)... 135.181.214.104, 88.99.208.237, 2a01:f80:10a:39da::2, ...
Connecting to downloads.apache.org (downloads.apache.org) [135.181.214.104]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 295219127 (282M) [application/x-gzip]
Saving to: 'hbase-2.4.18-bin.tar.gz'

hbase-2.4.18-bin.tar.g 100%[=====>] 281.54M 23.6MB/s in 13s

2024-11-14 21:02:59 (21.9 MB/s) - 'hbase-2.4.18-bin.tar.gz' saved [295219127/295219127]

cynnavarapu@a20561894-n2-m:~$ ssh-keygen -t rsa -b 2048
Generating public/private rsa key pair.
Enter file in which to save the key (/home/cynnavarapu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/cynnavarapu/.ssh/id_rsa.
Your public key has been saved in /home/cynnavarapu/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:6mx48+PGR/LkwhfR9C9FC0AfaAwUGC7hwU1uTqr4 cynnavarapu@a20561894-n2-m
The key's randomart image is:
+--[RSA 2048]-----+
| ..oo=+o.to. |
|o oo ++t +o. |
|.O .==O O.O . |
|. ..+.o = |
| . .+ S. +. |
| . .o.o+ + |
| . .o* . . |
| . .o.o+ . |
| E. .ooo |
+-----[SHA256]-----+

cynnavarapu@a20561894-n2-m:~$ ls -l ~/.ssh/
total 12
-rw-r----- 1 cynnavarapu cynnavarapu 1182 Nov 14 21:02 authorized_keys
-rw-r----- 1 cynnavarapu cynnavarapu 1831 Nov 14 21:03 id_rsa
-rw-r--r-- 1 cynnavarapu cynnavarapu 408 Nov 14 21:03 id_rsa.pub
cynnavarapu@a20561894-n2-m:~$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCKkgE+srIkQCRtMEITD2HSOQG+q1H1jENs1Qfc8ddqUpwYb8A5K2O4BoRzU5gR95BtOwGMxbv9cagKklDbgQUNhA/ViLQbU0X3P9ht0FEp7ajtK2gyqgh
2BzQp0w041lJ/AwXwNixG4/EyPTRZWhQZr3inQs9pfkzGectX/mssPiaD69vv+MrG3HCmMfxGXH1ZJj681v09g3+Q9W/Pu/hAtFPKbFPX9QpAnEqLg7pMc9KxyJQJEqY091rolCBYz0U8MMp0X5FaD
V136cjRuvbyochA9By9c03Ukve7yfi3i+DJfIOGV+KnEXPDy6EuBd83JA6QvJ2GNb cynnavarapu@a20561894-n2-m
cynnavarapu@a20561894-n2-m:~$ ssh-copy-id cynnavarapu@a20561894-n2-w0
The authenticity of host 'a20561894-n2-w0 (10.128.0.51)' can't be established.
ECDSA key fingerprint is SHA256:dCa+x/SS9mXGDvej16JHX/5b6Boc8Th9VUNlGfmlRSA.
Are you sure you want to continue connecting (yes/no)? yes

```

The image shows a terminal window with a dark background and light-colored text. At the top, there is a title bar with a logo and the text "SSH-in-browser". Below the title bar, the terminal displays the output of an SSH command. The first line is the SSH banner for the local host: "Linux a20561894-n2-m 5.10.0-0.deb10.16-cloud-amd64 #1 SMP Debian 5.10.127-2-bpo10+1 (2022-07-28) x86\_64". The next two lines are: "The programs included with the Debian GNU/Linux system are free software;" and "the exact distribution terms for each program are described in the individual files in /usr/share/doc/\*/copyright." This is followed by a blank line and then the Debian GNU/Linux warranty statement: "Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law." The next line is "Last login: Thu Nov 14 21:02:41 2024 from 35.235.244.33". Then, the user "cnynavarapu@a20561894-n2-m" enters the command "\$ ssh-copy-id cnynavarapu@a20561894-n2-w-1". The terminal shows the output of this command: "The authenticity of host 'a20561894-n2-w-1 (10.128.0.50)' can't be established. ECDSA key fingerprint is SHA256:Kg3HrS+5AQoGBZh02fC4N3bxVLLUUPKJKIUDc7tjKVs." The user then enters "yes" to continue connecting. The terminal shows the output of the ssh-copy-id command: "/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed" and "/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys". The next line is "Number of key(s) added: 1". Then, the user enters the command "\$ ssh 'cnynavarapu@a20561894-n2-w-1'", and the terminal shows the output: "Now try logging into the machine, with: 'ssh 'cnynavarapu@a20561894-n2-w-1'' and check to make sure that only the key(s) you wanted were added." The user then enters the command "\$ ssh cnynavarapu@a20561894-n2-w-1". The terminal shows the output of this command: "Linux a20561894-n2-w-1 5.10.0-0.deb10.16-cloud-amd64 #1 SMP Debian 5.10.127-2-bpo10+1 (2022-07-28) x86\_64". The next two lines are: "The programs included with the Debian GNU/Linux system are free software;" and "the exact distribution terms for each program are described in the individual files in /usr/share/doc/\*/copyright." This is followed by a blank line and then the Debian GNU/Linux warranty statement: "Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law." The final line is the prompt "cnynavarapu@a20561894-n2-w-1:~\$".

[illegible]

 UPLOAD FILE
  DOWNLOAD FILE
 



29.17 All

```

Took 0.0018 seconds
hbase:001:0> create 'csp554Tbl', 'cf1', 'cf2'
Created table csp554Tbl
Took 2.2954 seconds
=> Hbase::Table - csp554Tbl
hbase:002:0> describe 'csp554Tbl'
Table csp554Tbl is ENABLED
csp554Tbl
COLUMN FAMILIES DESCRIPTION
(NAME => 'cf1', BLOOMFILTER => 'ROW', IN_MEMORY => 'false', VERSIONS => '1', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', COMPRESSION => 'NONE',
TTL => 'FOREVER', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0')
(NAME => 'cf2', BLOOMFILTER => 'ROW', IN_MEMORY => 'false', VERSIONS => '1', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', COMPRESSION => 'NONE',
TTL => 'FOREVER', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0')

2 row(s)
Quota is disabled

```

## Exercise 2:

```
hbase:003:0> put 'csp554Tbl', 'Row1', 'cf1:name', 'Sam'
Took 0.0902 seconds
hbase:004:0> put 'csp554Tbl', 'Row2', 'cf1:name', 'Ahmed'
Took 0.0072 seconds
hbase:005:0> put 'csp554Tbl', 'Row1', 'cf2:job', 'Pilot'
Took 0.0090 seconds
hbase:006:0> put 'csp554Tbl', 'Row2', 'cf2:job', 'Doctor'
Took 0.0053 seconds
hbase:007:0> put 'csp554Tbl', 'Row1', 'cf2:level', 'LZ3'
Took 0.0060 seconds
hbase:008:0> put 'csp554Tbl', 'Row2', 'cf2:level', 'AR7'
Took 0.0053 seconds

hbase:009:0> scan 'csp554Tbl'
ROW
Row1      column=cf1:name, timestamp=2024-11-14T18:30:19.364, value=Sam
Row1      column=cf2:job, timestamp=2024-11-14T18:30:19.421, value=Pilot
Row1      column=cf2:level, timestamp=2024-11-14T18:30:19.490, value=LZ3
Row2      column=cf1:name, timestamp=2024-11-14T18:30:19.394, value=Ahmed
Row2      column=cf2:job, timestamp=2024-11-14T18:30:19.463, value=Doctor
Row2      column=cf2:level, timestamp=2024-11-14T18:30:28.328, value=AR7
2 row(s)
Took 0.0356 seconds
```

## Exercise 3:

```
hbase:010:0> get 'csp554Tbl', 'Row1', 'cf2:level'
COLUMN    CELL
cf2:level  timestamp=2024-11-14T18:30:19.490, value=LZ3
1 row(s)
Took 0.0134 seconds
```

## Exercise 4:

```
hbase:011:0> get 'csp554Tbl', 'Row2', 'cf1:name'
COLUMN    CELL
cf1:name   timestamp=2024-11-14T18:30:19.394, value=Ahmed
1 row(s)
Took 0.0076 seconds
```

## Exercise 5:

```
hbase:012:0> scan 'csp554Tbl', (LIMIT => 2)
ROW
Row1      column=cf1:name, timestamp=2024-11-14T18:30:19.364, value=Sam
Row1      column=cf2:job, timestamp=2024-11-14T18:30:19.421, value=Pilot
Row1      column=cf2:level, timestamp=2024-11-14T18:30:19.490, value=LZ3
Row2      column=cf1:name, timestamp=2024-11-14T18:30:19.394, value=Ahmed
Row2      column=cf2:job, timestamp=2024-11-14T18:30:19.463, value=Doctor
Row2      column=cf2:level, timestamp=2024-11-14T18:30:28.328, value=AR7
2 row(s)
Took 0.0158 seconds
```

## Exercise 6:

### A Novel HBase Data Storage in Wireless Sensor Networks

#### Introduction to the Problem

Wireless sensor networks (WSNs) generate enormous amounts of data that traditional storage systems often struggle to manage. These networks rely on distributed sensors, which gather data continuously across various regions. Managing this high-speed, high-volume data efficiently is critical, especially in real-time applications like environmental monitoring and smart cities. This study addresses the need for a storage solution that can handle WSN data's unique demands.

## Purpose and Hypothesis

The article explores the potential of using an optimized HBase-based storage system to improve data management in WSNs. The researchers hypothesize that a strategically configured HBase cluster could meet WSNs' real-time processing requirements, balancing quick data access, efficient storage usage, and seamless retrieval.

## Methodology

- **Two-layer Distributed Storage Model**
  - To structure the data effectively, the researchers developed a two-layer distributed model in HBase. They used Extencis primitives—a formalization approach to organize heterogeneous datasets—allowing seamless storage and retrieval of data from multiple sources across different regions.
- **Multi-threaded Data Buffering and Partitioning**
  - A multi-threaded approach was adopted for data buffering and partitioning, enabling faster data processing. This system efficiently organizes data before storage, reducing bottlenecks caused by high-speed data flow.
- **Filters and Coprocessors**
  - To enhance data retrieval, HBase's filtering and coprocessing features were utilized. These tools enable the system to handle large data sets and distribute complex processing tasks across HBase servers, which improves retrieval speed and efficiency.
- **Dynamic Node Updating for Scalability**
  - A dynamic updating mechanism for HBase nodes allows the system to expand as data demands grow. This feature helps ensure the storage solution remains scalable and adaptable to increasing data volumes.

## Results and Findings

- **Performance Gains in Speed and Efficiency**
  - The optimized HBase storage system demonstrated notable improvements in both speed and efficiency over single-node storage configurations. Multi-threaded data processing significantly reduced delays and boosted throughput, while the cluster-based model achieved faster write and read times, proving that the HBase architecture is effective for real-time data processing.
- **Specific Performance Metrics**
  - The article details a comprehensive performance analysis, including tests that compare write and read times, delays at different buffer thresholds, and a comparison to Oracle's streaming data storage. The system's performance was shown to surpass traditional storage setups, particularly in high-speed data scenarios typical of WSNs.

## Discussion and Future Directions

### System Suitability and Potential Improvements

The findings suggest that an optimized HBase cluster is highly suitable for WSN data management, especially in applications demanding real-time data access and large-scale data integration. The authors also discuss potential future improvements, such as enhancing fault tolerance and optimizing storage space management to further boost reliability.

## Conclusion

The study provides a compelling case for using HBase as a storage solution in WSNs. With the right optimizations, HBase can handle the specific demands of WSNs, including real-time data processing and scalability. This research contributes to the field of WSN data management, offering practical insights and solutions for data-intensive applications.