

CARA BUAT CLUSTER UNTUK HADOOP

Masuk ke EMR

Create user lalu pilih hadoop dan hive

The screenshot shows the 'Create cluster' wizard in the AWS EMR console. On the left, under 'Application bundle', there are several options: Spark Interactive, Core Hadoop, Flink, HBase, Presto, Trino, and Custom. Under 'Custom', there are checkboxes for various services like AmazonCloudWatchAgent, Hue, Livy, Pig, TensorFlow, Zeppelin, Flink, Hadoop, Oozie, Presto, Tez, ZooKeeper, Hive, JupyterEnterpriseGateway, Phoenix, Spark, and Trino. Most of these checkboxes are unchecked except for Hadoop 3.4.1 and Hive 3.1.3. Below this is a section for 'AWS Glue Data Catalog settings' and 'Operating system options'. On the right, the 'Summary' pane shows the selected 'Name and applications - required' (Amazon EMR release emr-7.12.0, Application bundle Custom (Hadoop 3.4.1, Hive 3.1.3)), 'Cluster configuration - required' (Uniform instance groups: Primary (r8g.xlarge), Core (r8g.xlarge), Task (r8g.xlarge)), and 'Cluster scaling and provisioning - required' (Provisioning configuration: Core size: 1 instance, Task size: 1 instance). A callout box highlights the 'Configure IAM roles' section, which requires choosing a service role and instance profile before creating the cluster. At the bottom right are 'Cancel' and 'Create cluster' buttons.

Turunkan instance type (biaya)

Di Cluster configuration (bagian bawah):

Primary / otak

✗ r8g.xlarge (INI MAHAL)

Ganti jadi

M5.large

Lalu di core juga samakan

Untuk tasknya itu ibarat pekerja tambahan remove aja (pakai kalau kelas production)

Task 1 of 1

Name
Task - 1 Remove instance group

Choose EC2 instance type
m5.xlarge
4 vCore 16 GiB memory
EBS only storage
On-Demand price: \$0.192 per instance/ ...
Lowest Spot price: \$0.053 (us-east-1f)

Node configuration - optional

Add task instance group
You can add up to 47 more task instance groups.

Ebs root

Untuk apa? OS, Hadoop binaries, Log, Temporary file

Amazon EMR > EMR on EC2: Clusters > Create cluster

Add task instance group
You can add up to 48 more task instance groups.

EBS root volume
EBS root volume applies to the operating systems and applications that you install on the cluster. [EBS root volume ratio constraints](#) ▾

Size (GiB) 15	IOPS 3000	Throughput (MiB/s) 125
------------------	--------------	---------------------------

15 - 100 GiB per volume
General Purpose SSD (gp3)

3000 - 16000 IOPS per volume. Choose a maximum ratio of 500:1 between IOPS and volume size.

125 - 1000 MiB/s per volume. Choose a maximum ratio of 0.25:1 between throughput and IOPS.

Cluster scaling and provisioning - required Info
Choose how Amazon EMR should size your cluster.

Choose an option

- Set cluster size manually
Use this option if you know your workload patterns in advance.
- Use EMR-managed scaling
Monitor key workload metrics so that EMR can optimize the cluster size and resource utilization.
- Use custom automatic scaling
To programmatically scale core and task nodes, create custom automatic scaling policies.

Provisioning configuration
Set the size of your core instance group. Amazon EMR attempts to provision this capacity when you launch your cluster.

Name	Instance type	Instance(s) size	Use Spot purchasing option
Core	m5.xlarge	1	<input type="checkbox"/>

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Setting yang benar

Size (GiB):

👉 15 GB ✓

IOPS:

👉 Biarkan default (3000)

Throughput:

👉 Biarkan default (125)

Cluster Scalling and provisioning

The screenshot shows the 'Cluster scaling and provisioning - required' section. It asks to choose how Amazon EMR should size your cluster. Three options are available: 'Set cluster size manually' (selected), 'Use EMR-managed scaling', and 'Use custom automatic scaling'. Below this, the 'Provisioning configuration' section allows setting the size of the core instance group. A table shows 'Core' with 'm5.xlarge' and '1' instances. The 'Networking - required' section shows a VPC and subnet selection. The 'Steps (0)' section indicates no steps have been defined.

- Kamu mengunci jumlah node
- EMR tidak auto nambah node
- Biaya terkontrol
- Cocok buat belajar

Networking

Vpc sudah terisi

Subnet sudah terisi

SSH / Amazon EC2 key pair for SSH to the cluster

Kunci untuk masuk (SSH) ke master node tanpa ini tidak bisa ssh, tidak bisa praktik hdfs, dfs, hive

Klik create key pair

Beri nama misal emr-belajar-key

Type RSA

Format .pem

Amazon EMR service role

Pilih Create a service role

Ec2 instance profile for amazon emr

Itu isi create an instance profile

Kalo sudah klik create, tunggu sampain status ke waiting

Setelah waiting selesai klik Connect to the Primary node using SSM

Lalu connect

PRAKTEK HDFS

HDFS BASIC

Step 1

Pastikan login sebagai user hadoop ketik whoami target outputnya hadoop

Kalau mau ganti / masih root pindah aja sudo su – hadoop

Step 2 - Lihat “rumah” lo di HDFS

Di Hadoop, tiap user punya direktori sendiri di HDFS:

hdfs dfs -ls /user

```
hadoop@ip-172-31-10-158 ~]$ hdfs dfs -ls /user
Found 4 items
rwxrwxrwx - hadoop hdfsadmingroup          0 2026-02-09 06:46 /user/hadoop
rwxr-xr-x - mapred mapred                 0 2026-02-09 06:46 /user/history
rwxrwxrwx - hdfs  hdfsadmingroup          0 2026-02-09 06:46 /user/hive
rwxrwxrwx - root   hdfsadmingroup          0 2026-02-09 06:46 /user/root
hadoop@ip-172-31-10-158 ~]$
```

Penjelasan

1. user/hadoop : owner : hadoop, ini direktori kerja utama saya di HDFS, Semua latihan HDFS & Hive nanti akan dilakukan di sini
2. /user/history : Dipakai Hadoop internal, buat job history, dipakai resource manager dan mapreduce
3. /user/hive : Dipakai Hive, Metadata & warehouse, Table Hive sering nyimpan data di bawah sini
4. user/root : Direktori untuk user root di HDFS

Praktik 1 bikin direktori sendiri di HDFS

Buat : hdfs dfs -mkdir /user/hadoop/input

Cek : hdfs dfs -ls /user/hadoop maka akan muncul

Knapa?

Hadoop SELALU kerja pakai folder

Hive & MapReduce hampir gak pernah baca file langsung di root

Pola standar:

/user/hadoop/input → data mentah

/user/hadoop/output → hasil proses

Praktik 2 – bikin file lokasl (linux)

Bikin file sementara

```
echo -e "1,andi,90\n2,budi,85\n3,citra,95" > nilai.csv
```

```
ls -l nilai.csv
```

```
cat nilai.csv
```

lalu enter maka file akan muncul

```
[hadoop@ip-172-31-10-158 ~]$ echo -e "1,andi,90\n2,budi,85\n3,citra,95" > nilai.csv
ls -l nilai.csv
cat nilai.csv
-rw-r--r--. 1 hadoop hadoop 31 Feb  9 07:28 nilai.csv
1,andi,90
2,budi,85
3,citra,95
[hadoop@ip-172-31-10-158 ~]$ █
```

Praktik 3 – Upload ke HDFS

Pindahkan file itu ke HDFS

```
hdfs dfs -put nilai.csv /user/hadoop/input/
```

lalu cek apakah ada di hdfs

```
hdfs dfs -ls /user/hadoop/input
```

```
hdfs dfs -cat /user/hadoop/input/nilai.csv
```

```
3,citra,95
[hadoop@ip-172-31-10-158 ~]$ hdfs dfs -put nilai.csv /user/hadoop/input/
[hadoop@ip-172-31-10-158 ~]$ hdfs dfs -ls /user/hadoop/input
Found 1 items
-rw-r--r-- 1 hadoop hdfsadmin:group          31 2026-02-09 07:30 /user/hadoop/input/nilai.csv
[hadoop@ip-172-31-10-158 ~]$ hdfs dfs -cat /user/hadoop/input/nilai.csv
1,andi,90
2,budi,85
3,citra,95
[hadoop@ip-172-31-10-158 ~]$ █
```

Penjelasan singkat:

- `rw-r--r--` → permission (kayak Linux)

- 1 → replication factor
- 31 → ukuran file (byte)

👉 Karena cluster lo cuma **1 DataNode**, replikasinya = **1**
Di cluster besar, biasanya = **3**

Praktik 4 - Hapus File Lokal

rm nilai.csv

ls nilai.csv

yang terjadi itu hapus file lokal (linux) bukan hdfs

Praktik 5 Cara Hapus file HDFS

hdfs dfs -rm /user/hadoop/input/nilai.csv

kalau sukses otuptunya

Deleted /user/hadoop/input/nilai.csv

Praktik 6 Melihat Metadata

hdfs dfs -stat "%n | size=%b | repl=%r | block=%o" /user/hadoop/input/nilai.csv

output

nilai.csv | size=31 | repl=1 | block=134217728

Artinya:

- size → ukuran file asli (31 byte)
- repl → replication factor (1)
- block → block size HDFS (128 MB)

👉 Walaupun file cuma 31 byte, HDFS tetap alokasikan 1 block.

Lihat block ID & lokasi DataNode

hdfs fsck /user/hadoop/input/nilai.csv -files -blocks -locations

```
/user/hadoop/input/nilai.csv 31 bytes, replicated: replication=1, 1 block(s): OK
0. BP-1260015907-172.31.10.158-1770619602913:blk_1073741826_1002 len=31 Live_repl=1
bd4-43a2-93e2-5deed20e222d,DISK]

Status: HEALTHY
Number of data-nodes: 1
Number of racks: 1
Total dirs: 0
Total symlinks: 0

Replicated Blocks:
Total size: 31 B
Total files: 1
Total blocks (validated): 1 (avg. block size 31 B)
Minimally replicated blocks: 1 (100.0 %)
Over-replicated blocks: 0 (0.0 %)
Under-replicated blocks: 0 (0.0 %)
Mis-replicated blocks: 0 (0.0 %)
Default replication factor: 1
Average block replication: 1.0
Missing blocks: 0
Corrupt blocks: 0
Missing replicas: 0 (0.0 %)
Blocks queued for replication: 0

Erasure Coded Block Groups:
Total size: 0 B
Total files: 0
Total block groups (validated): 0
Minimally erasure-coded block groups: 0
Over-erasure-coded block groups: 0
Under-erasure-coded block groups: 0
Unsatisfactory placement block groups: 0
Average block group size: 0.0
Missing block groups: 0
Corrupt block groups: 0
Missing internal blocks: 0
Blocks queued for replication: 0
FSCK ended at Mon Feb 09 07:43:36 UTC 2026 in 13 milliseconds
```

Artinya:

- File = 31 byte
- Replikasi = 1
- Jumlah block = 1
- Status = SEHAT

👉 Walaupun kecil, HDFS tetap treat ini sebagai 1 block.

Block

blk_1073741826_1002 len=31

blk itu block ID

len 31 itu isi blok = 31byte

Datanode

DatanodeInfoWithStorage[172.31.9.11:9866, ... , DISK]

Block ini disimpan di Core node

IP: 172.31.9.11

Storage type: DISK

Itu mesin EC2 lain, bukan primary node

HIVE BASIC

Hive adalah SQL diatas HDFS

Data sudah ada tadi diatas disini

/user/hadoop/input/nilai.csv

Rencananya mau kueri pakai sql

Beeline No connection

Kalau no connection berarti dia belum nyambung ke hive service, ibarat buka mysql client tapi blm connect ke database

Cara memperbaiki

!connect jdbc:hive2://localhost:10000/default

Kalau diminta username masukan aja hadoop lalu pasword langsung enter aja

Step 1 masuk ke hive (beeline)

Diterminal yang sama jalankan beeline

Buat database

CREATE DATABASE IF NOT EXISTS belajar;

Lalu use database belajar

Lalu show databases buat cek apakah udah dibuat atau blm

```
INFO : OK
INFO : Concurrency mode is disabled, not creating locks
+-----+
| database_name |
+-----+
| belajar      |
| default       |
+-----+
2 rows selected (0.431 seconds)
0: jdbc:hive2://localhost:10000/default>
```

Step 2 buat tabel

```
CREATE TABLE nilai (
    id INT,
    nama STRING,
    skor INT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

Lalu load datanya

```
LOAD DATA INPATH '/user/hadoop/input/nilai.csv'
INTO TABLE nilai;
```

Query

```
SELECT * FROM nilai;
```

```
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Executing command(queryId=hive_20260209081212_07111e29-a25c-45b1-9ca5-43
INFO : Completed executing command(queryId=hive_20260209081212_07111e29-a25c-45
INFO : OK
INFO : Concurrency mode is disabled, not creating a lock manager
+-----+-----+
| nilai.id | nilai.nama | nilai.skor |
+-----+-----+
| 1        | andi     | 90      |
| 2        | budi     | 85      |
| 3        | citra    | 95      |
+-----+-----+
3 rows selected (0.978 seconds)
0: jdbc:hive2://localhost:10000/default>
```

```
SELECT *
FROM nilai
WHERE skor >= 90;
```

```
INFO : Concurrency mode is disabled, not creating a lock
+-----+-----+-----+
| nilai.id | nilai.nama | nilai.skor |
+-----+-----+-----+
| 1         | andi      | 90       |
| 3         | citra     | 95       |
+-----+-----+-----+
2 rows selected (0.652 seconds)
0: jdbc:hive2://localhost:10000/default> |
```