

Apache Flink Tutorial

Name : Mahua Nitin Hiray (she/her/hers)

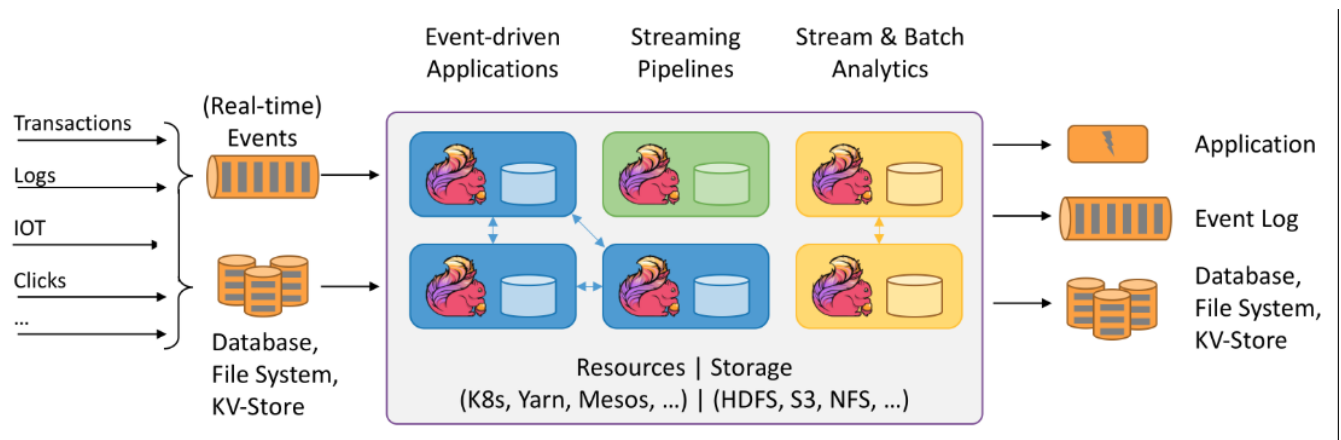
Email : mahuanitin.hiray001@umb.edu

Linkedin : <https://www.linkedin.com/in/mahuarani/> (Let's connect to innovate!)

Section 1 – Introduction

1) Flink: A Quick Overview

- Apache Flink is a powerful open-source stream processing framework for distributed, high-performing, always-available, and accurate data streaming applications. It provides efficient, fast, and reliable processing of both batch and stream data, making it a valuable tool for real-time analytics such as stock market analysis.



(**Must watch**

: <https://www.youtube.com/watch?v=3cg5dABA6mo&list=PLa7VYi0yPIH1UdmQcnUr8lvjbUV8JriK0&index=1>

)

- *Use Case:* Flink is ideal for processing continuous data streams, offering advantages in real-time analytics, where timely insights are crucial, such as monitoring stock market trends.
- *Read the article at:* <https://infofarm.be/streaming-first-with-apache-flink/> (I like how the other four big data technologies' drawbacks are established to give us a comprehensive understanding)

2) Advantages:

- High throughput and low-latency processing.
- Scalable and fault tolerant.
- Supports event time processing and state management.

3) Disadvantages:

- Complexity in setup and management.
- Steeper learning curve compared to some other streaming platforms.

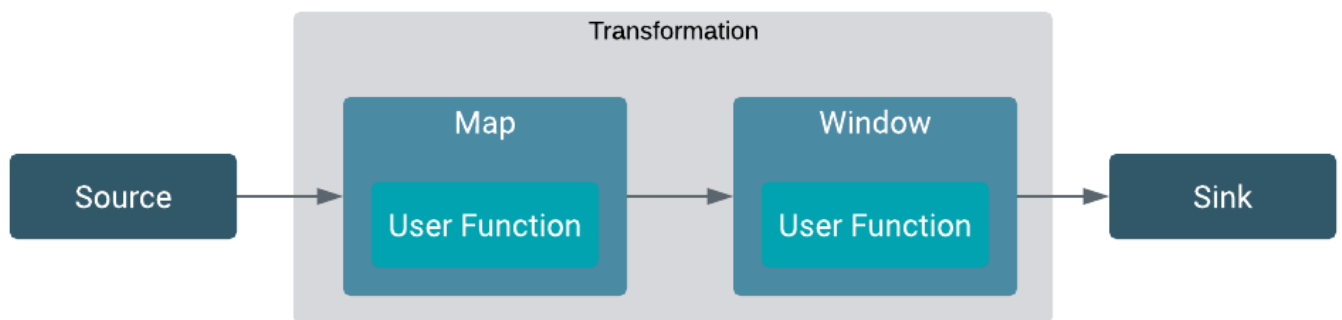
4) Alternatives Comparison:

Compared to Hadoop's MapReduce, Flink offers lower latency and can handle real-time streaming data. Unlike Spark, Flink's native support for streaming allows for more fine-grained control of state and time.

5) Flink Dataflow Model

Flink operates on the principle of dataflow, where data flows through a series of transformations. The core concepts include:

- **DataStream**: Represents a stream of data.
- **Transformation**: Operations applied to DataStreams.
- **Sink**: Defines where the processed data should be sent.



Further Reading : <https://nightlies.apache.org/flink/flink-docs-release-1.2/concepts/programming-model.html>

Section 2 – Requirements

Before you begin this tutorial, you will need an AWS (Amazon Web Services) account. If you do not already have an AWS account, you can create one by visiting the [AWS Sign-Up Page](#) . Please note that while AWS offers a free tier for new accounts, some services and resources used in this tutorial may incur costs. To avoid any errors or interruptions in service, **make sure to provide your credit card details when setting up your account.** This will enable you to access AWS services that are beyond the scope of the free tier or to continue using services after the free tier limits are exceeded.

Let's Start !

1. In the **AWS Management Console** choose **Services**, choose **Compute** and then choose **EC2**.
2. Choose the Launch instance menu and select **Launch instance**.

Step 1: Name and tags + Application and OS Images (Amazon Machine Image)

3. Configure the instance as follows:
 - **Name** : MyFlink
 - **Quick Start AMIs** : Ubuntu
 - **AMI** : Ubuntu Server 22.04 LTS (HVM), SSD Volume Type (Free tier Eligible)

Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Quick Start

Amazon Linux

aws

macOS

Mac

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

Red Hat

SUSE Linux

SUSE

Browse more AMIs
 Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type
 ami-0fc5d935ebf8bc3bc (64-bit (x86)) / ami-016485166ec7fa705 (64-bit (Arm))
 Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description
 Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-09-19

Architecture
 64-bit (x86)

AMI ID
 ami-0fc5d935ebf8bc3bc

Verified provider

Step 2 : Instance Type

4. In the Instance type panel, select **t3.large**

▼ Instance type [Info](#)

Instance type

t3.large
Family: t3 2 vCPU 8 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0832 USD per Hour
On-Demand Windows base pricing: 0.1108 USD per Hour
On-Demand RHEL base pricing: 0.1432 USD per Hour
On-Demand SUSE base pricing: 0.1395 USD per Hour

☒ All generations
[Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

Step 3 : Key pair (login)

5. For Key pair name - required, create new key pair
 - **Key pair name** : mahuakey
 - **Key pair type** : RSA
 - **Private key file format** : .pem *(For use with OpenSSH)*
 - **Click on Create key pair** , it will be in your downloads, store it safely in your root folder.

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

mahuakey

[Create new key pair](#)

Step 4 : Network settings

6. Under Firewall (security groups) , choose all the three options :
 - Allow SSH traffic from Helps you connect to your instance
 - Allow HTTPS traffic from the internetTo set up an endpoint, for example when creating a web server

- Allow HTTP traffic from the internet To set up an endpoint, for example when creating a web server

▼ **Network settings** [Info](#)

Edit

Network [Info](#)

vpc-04fe39347878aaec9

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group

☐ Select existing security group

We'll create a new security group called 'launch-wizard-5' with the following rules:

☒ Allow SSH traffic from

Helps you connect to your instance

Anywhere
0.0.0.0/0

☒ Allow HTTPS traffic from the internet

To set up an endpoint, for example when creating a web server

☒ Allow HTTP traffic from the internet

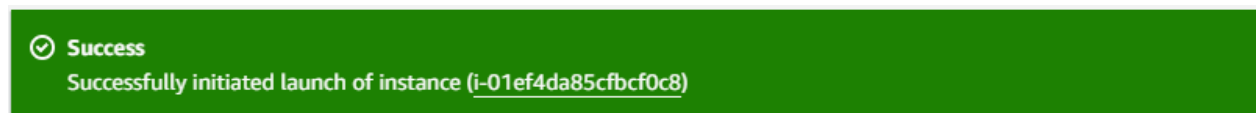
To set up an endpoint, for example when creating a web server

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

×

Step 5 : Launch the instance

7. At the bottom of the Summary panel on the right side of the screen choose Launch instance You will see a Success message



8. Choose View all instances
 - In the Instances list, select Web Server.
 - Review the information displayed in the Details tab.

- It includes information about the instance type, security settings and network settings.
- The instance is assigned a *Public IPv4 DNS* that you can use to contact the instance from the Internet.
- To view more information, drag the window divider upwards.
- At first, the instance will appear in a *Pending* state, which means it is being launched. It will then change to *Initializing*, and finally to *Running*.

EC2 > Instances > i-01ef4da85cfbcf0c8

Instance summary for i-01ef4da85cfbcf0c8 (MyFlink) [Info](#)

[Refresh](#)
[Connect](#)
[Instance state ▼](#)
[Actions ▼](#)

Updated less than a minute ago

Instance ID i-01ef4da85cfbcf0c8 (MyFlink)	Public IPv4 address 54.81.15.213 open address
Private IPv4 addresses 172.31.44.246	IPv6 address -
Instance state ✔ Running	Public IPv4 DNS ec2-54-81-15-213.compute-1.amazonaws.com open address
Hostname type IP name: ip-172-31-44-246.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-44-246.ec2.internal
	Answer private resource DNS name IPv4 (A)
Instance type t3.large	Elastic IP addresses -
Auto-assigned IP address 54.81.15.213 [Public IP]	VPC ID vpc-04fe39347878aaec9 open
AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more	IAM Role -
Subnet ID subnet-084419867c9984701 open	Auto Scaling Group name -
IMDSv2 Required	

[Details](#) | [Security](#) | [Networking](#) | [Storage](#) | [Status checks](#) | [Monitoring](#)

▼ Instance details [Info](#)

Platform	AMI ID
----------	--------

- Wait for your instance to display the following:

- **Instance State:** *Running*
- **Status Checks:** *2/2 checks passed*



10. Select **MyFlink** and click on the security dashboard and under inbound rules , scroll and check the three port range.

Instance: i-01ef4da85cfbcf0c8 (MyFlink)

Standard Time) [sg-0f5e5a4d77efdfedf \(launch-wizard-5\)](#)

▼ Inbound rules

Name	Security group rule ID	Port range
–	sgr-02326ee07c669074b	22
–	sgr-0a6b1b11b13cb18e0	80
–	sgr-075ec0ee3182df82b	443

11. On the left hand side , under network and security click on security Groups

12. Select the recent security group we made while launching the instance

EC2 Dashboard
EC2 Global View
Events

▼ Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
New

▼ Images
AMIs
AMI Catalog

▼ Elastic Block Store
Volumes
Snapshots
Lifecycle Manager

▼ Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces

▼ Load Balancing
Load Balancers
Target Groups

Security Groups (1/6) Info

Actions Export security groups to CSV Create security group

Find resources by attribute or tag

Name	Security group ID	Security group name
–	sg-088d85b0067564c32	launch-wizard-3
–	sg-095ab8342a442d16d	launch-wizard-2
–	sg-06819a71d99f7b0ef	launch-wizard-4
–	sg-090a62340c46e21d8	launch-wizard-1
–	sg-02fbf2111feb92d2c	default
<input checked="" type="checkbox"/>	sg-0f5e5a4d77efdfedf	launch-wizard-5

sg-0f5e5a4d77efdfedf - launch-wizard-5

Details **Inbound rules** Outbound rules Tags

Inbound rules (3)

Search

Name	Security group rule...	IP version	Type
–	sgr-02326ee07c66907...	IPv4	SSH

13. Under Inbound Rules , select Edit Inbound Rules

14. Scroll down , and then select add rule.

15. Configure a rule as follows :

- **Type** : Custom TCP
- **Port Range** : 8081
- **Source Type** : Anywhere – Ipv4

Inbound rule 4 Delete

Security group rule ID	Type Info	Protocol Info
-	Custom TCP ▼	TCP
Port range Info	Source type Info	Source Info
8081	Anywhere-IPv4 ▼	<input type="text" value="0.0.0.0/0"/>
Description - optional Info		
<input type="text"/>		
Add rule		

✓ Inbound security group rules successfully modified on security group (sg-0f5e5a4d77efdfedf | launch-wizard-5) ✕
▶ Details

16. Now go back to EC2 and connect Instance.

[EC2](#) > [Instances](#) > [i-01ef4da85cfbcf0c8](#) > Connect to instance

Connect to instance [Info](#)

Connect to your instance i-01ef4da85cfbcf0c8 (MyFlink) using any of these options

EC2 Instance Connect

Session Manager

SSH client

EC2 serial console

Instance ID
i-01ef4da85cfbcf0c8 (MyFlink)

Connection Type

☒ Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

☐ Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address
54.81.15.213

User name
Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ubuntu.

ubuntu

Note: In most cases, the default user name, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Cancel **Connect**

17. Ubuntu cluster will be loaded.

```
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1012-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

System information as of Sun Nov 26 19:34:37 UTC 2023

System load:  0.080078125      Processes:            101
Usage of /:   20.5% of 7.57GB   Users logged in:     0
Memory usage: 2%               IPv4 address for ens5: 172.31.44.246
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

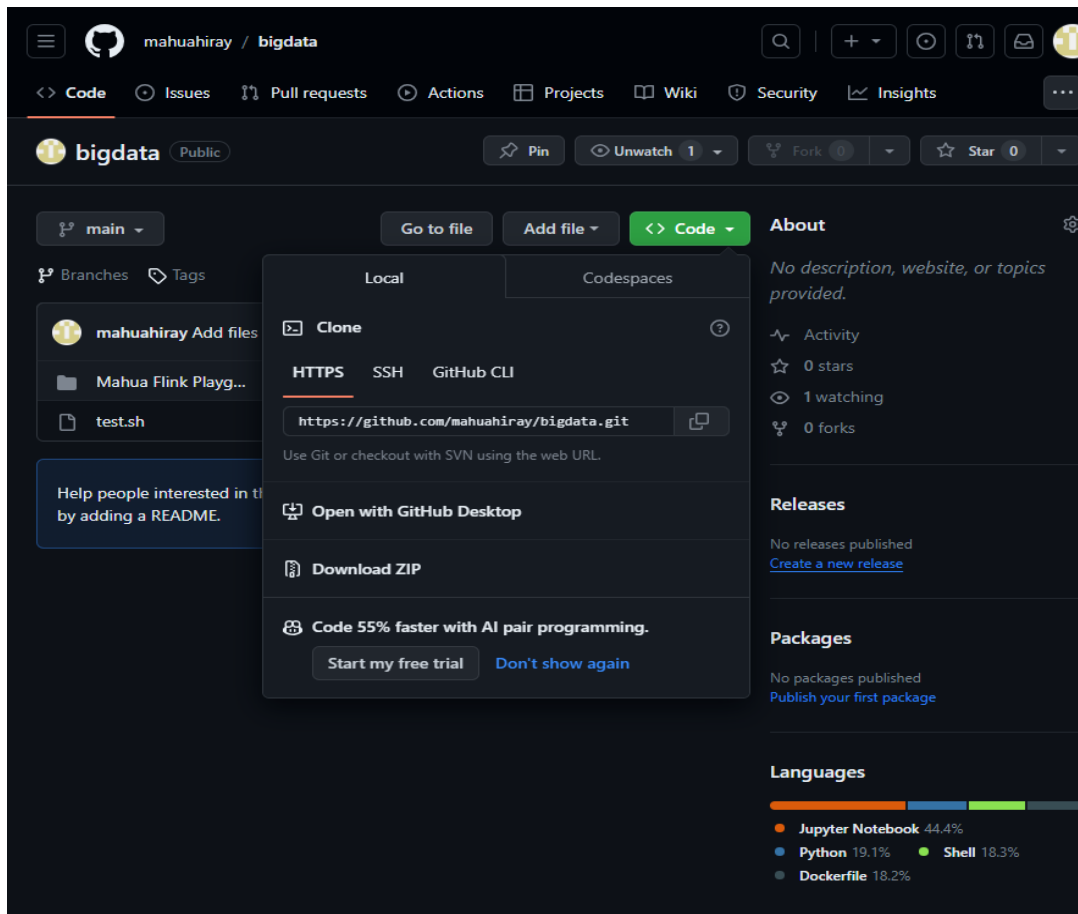
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-44-246:~$
```

18. Now open this GitHub link : <https://github.com/mahuahiray/bigdata.git>



19. Copy the code and come back to the Ubuntu cluster as displayed above.

Before copying the code take note :

This is the wrong way of copying the code :

```
ubuntu@ip-172-31-44-246:~$ git clone ^[[200~https://github.com/mahuahiray/bigdata.git~
```

This is the correct way :

```
ubuntu@ip-172-31-44-246:~$ git clone https://github.com/mahuahiray/bigdata.git
```

20. Type `git clone` and paste the link from GitHub. Press Enter.

```
ubuntu@ip-172-31-40-87:~$ git clone https://github.com/mahuahiray/bigdata.git
Cloning into 'bigdata'...
remote: Enumerating objects: 14, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 14 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (14/14), 4.76 KiB | 1.19 MiB/s, done.
```

21. Type `ls` (List the files inside the Folder)

```
ubuntu@ip-172-31-40-87:~$ ls
bigdata
```

22. Type `cd bigdata` (Changing Directory)

```
ubuntu@ip-172-31-18-182:~$ cd bigdata/
ubuntu@ip-172-31-18-182:~/bigdata$ ls
'Mahua Flink Playground'  dependencies.sh
ubuntu@ip-172-31-18-182:~/bigdata$
ubuntu@ip-172-31-18-182:~/bigdata$
```

23. Become a root user using `Sudo su`

```
ubuntu@ip-172-31-18-182:~/bigdata$ sudo su
root@ip-172-31-18-182:/home/ubuntu/bigdata#
root@ip-172-31-18-182:/home/ubuntu/bigdata#
```

24. To install dependency run `bash dependencies.sh`

25. If prompted click “enter” to complete the installation

26. Once the installation completes it should show us the “docker” and “ docker-compose version”

```
Installation complete.
Docker version:
Docker version 24.0.7, build afdd53b
Docker Compose version:
docker-compose version 1.29.2, build 5becea4c
root@ip-172-31-18-182:/home/ubuntu/bigdata#
root@ip-172-31-18-182:/home/ubuntu/bigdata#
root@ip-172-31-18-182:/home/ubuntu/bigdata#
```

27. When you type `ls` you can now see two directories: ‘Mahua Flink Playground’ and ‘dependencies.sh’

28. Now change directory into ‘Mahua Flink Playground’ using `cd Mahua Flink Playground`

```
root@ip-172-31-18-182:/home/ubuntu/bigdata# cd Mahua\ Flink\ Playground/
root@ip-172-31-18-182:/home/ubuntu/bigdata/Mahua Flink Playground#
root@ip-172-31-18-182:/home/ubuntu/bigdata/Mahua Flink Playground#
root@ip-172-31-18-182:/home/ubuntu/bigdata/Mahua Flink Playground# ls
docker-compose.yml  examples  image
root@ip-172-31-18-182:/home/ubuntu/bigdata/Mahua Flink Playground#
```

29. After using `cd Mahua Flink Playground` you will go to 'Mahua Flink Playground'
30. Type `ls` to see three directories inside.
31. Type `docker compose up -d` after this Apache Flink and its components will be installed and running on your server.

```
root@ip-172-31-18-182:/home/ubuntu/bigdata/Mahua Flink Playground# docker compose up -d
[+] Running 20/22
! taskmanager Pulling
! jobmanager 20 layers [#####] 0B/0B Pulling
  ✓ bd8f6a7501cc Pull complete
  ✓ 44718e6d535d Pull complete
  ✓ efe9738af0cb Pull complete
  ✓ c647a8e650d3 Pull complete
  ✓ 4c209a1e5186 Pull complete
  ✓ 2ea976dfda62 Pull complete
  ✓ 5d9f9110ab0f Pull complete
  ✓ aac3410d44e8 Download complete
  ✓ 00ff9be65265 Download complete
  ✓ 263b17b8a316 Download complete
  ✓ 1607bd8b25ad Download complete
  ✓ 84d043623632 Download complete
  ✓ eb7388d52077 Download complete
  ✓ da3704983bfc Download complete
  ✓ 00a25e5f57db Download complete
  ✓ 721c9f97bc85 Download complete
  ✓ 662acff7199a Download complete
  ✓ c8a63e2c6705 Download complete
  ✓ 8aa523508773 Download complete
  ✓ 12c62ec90179 Download complete
```

32. Wait for the completion. This takes about 5-7 minutes.

```
[+] Running 22/22
  ✓ taskmanager Pulled
  ✓ jobmanager 20 layers [#####] 0B/0B Pulled
    ✓ bd8f6a7501cc Pull complete
    ✓ 44718e6d535d Pull complete
    ✓ efe9738af0cb Pull complete
    ✓ c647a8e650d3 Pull complete
    ✓ 4c209a1e5186 Pull complete
    ✓ 2ea976dfda62 Pull complete
    ✓ 5d9f9110ab0f Pull complete
    ✓ aac3410d44e8 Pull complete
    ✓ 00ff9be65265 Pull complete
    ✓ 263b17b8a316 Pull complete
    ✓ 1607bd8b25ad Pull complete
    ✓ 84d043623632 Pull complete
    ✓ eb7388d52077 Pull complete
    ✓ da3704983bfc Pull complete
    ✓ 00a25e5f57db Pull complete
    ✓ 721c9f97bc85 Pull complete
    ✓ 662acff7199a Pull complete
    ✓ c8a63e2c6705 Pull complete
    ✓ 8aa523508773 Pull complete
    ✓ 12c62ec90179 Pull complete
[+] Running 3/3
  ✓ Network mahuaflinkplayground_default Created
  ✓ Container mahuaflinkplayground-jobmanager-1 Started
  ✓ Container mahuaflinkplayground-taskmanager-1 Started
root@ip-172-31-18-182:/home/ubuntu/bigdata/Mahua Flink Playground#
```

33. Once it is completed you should see two containers inside it. *This ensures that the job manager and task manager has been downloaded.*

34. Copy the Public IPv4 address of your instance that is in the bottom left corner.

The screenshot shows the AWS Management Console for an EC2 instance named 'i-01ef4da85cfbcf0c8 (MyFlink)'. The 'Details' tab is selected. On the left, the 'Instance summary' shows the instance ID, private IP address (172.31.44.246), and state (Running). On the right, the 'Public IPv4 address' is highlighted as 54.81.15.213. An inset terminal window shows the output of the 'cat /etc/passwd' command, displaying the 'root' user and the 'mahuaflinkplayground-taskmanager-1' container.

35. Go to your browser and paste this : [http://54.81.15.213:8081](http://54.81.15.213:8081/#/overview) (Make sure it is http and not https)

The screenshot shows the Apache Flink Dashboard in a web browser. The 'Overview' page is displayed, showing 'Available Task Slots' as 20. Below this, it shows 'Total Task Slots 20' and 'Task Managers 1'. The 'Running Job List' section is empty, with columns for 'Job Name' and 'Start Time'. The left sidebar contains navigation links for Overview, Jobs, Running Jobs, Completed Jobs, Task Managers, Job Manager, and Submit New Job.

36. You should be able to see the Apache Flink Dashboard like above.

37. Now, its time to run some examples.

38. I have prepared 3 examples to demonstrate the data analysis.

1. Word-Count (Filter and count persons age > 25)

To do word-count:

Run: `docker-compose exec jobmanager ./bin/flink run -py /opt/examples/word-count.py`

```
root@ip-172-31-18-182:/home/ubuntu/bigdata/Mahua Flink Playground# docker-compose exec jobmanager ./bin/flink run -py /opt/examples/word-count.py
Job has been submitted with JobID cb79e3552f93977961a9ce7fe2aef131
```

op	name	count_age
+I	Mahua	1
+I	Yagna	1
+I	Zubbi	1
-U	Yagna	1
+U	Yagna	2

```
5 rows in set
root@ip-172-31-18-182:/home/ubuntu/bigdata/Mahua Flink Playground#
```

2. Product-Sales (Calculate total units sold per product)

To run product-sales example:

Run: `docker-compose exec jobmanager ./bin/flink run -py /opt/examples/product-`

```
root@ip-172-31-18-182:/home/ubuntu/bigdata/Mahua Flink Playground# docker-compose exec jobmanager ./bin/flink run -py /opt/examples/product-sales.py
Job has been submitted with JobID ca3376857f5c6a79f12e14fe31b3b6cb
```

op	product_id	total_units
+I	P001	10
+I	P002	5
-U	P001	10
+U	P001	25
+I	P003	20
-U	P002	5
+U	P002	15

```
7 rows in set
sales.py
```

3. Employee data analysis (Employee age > 30 and sort them in descending order)

To run employee data analysis:

Run: `docker-compose exec jobmanager ./bin/flink run -py /opt/examples/employee-data-analysis.py`

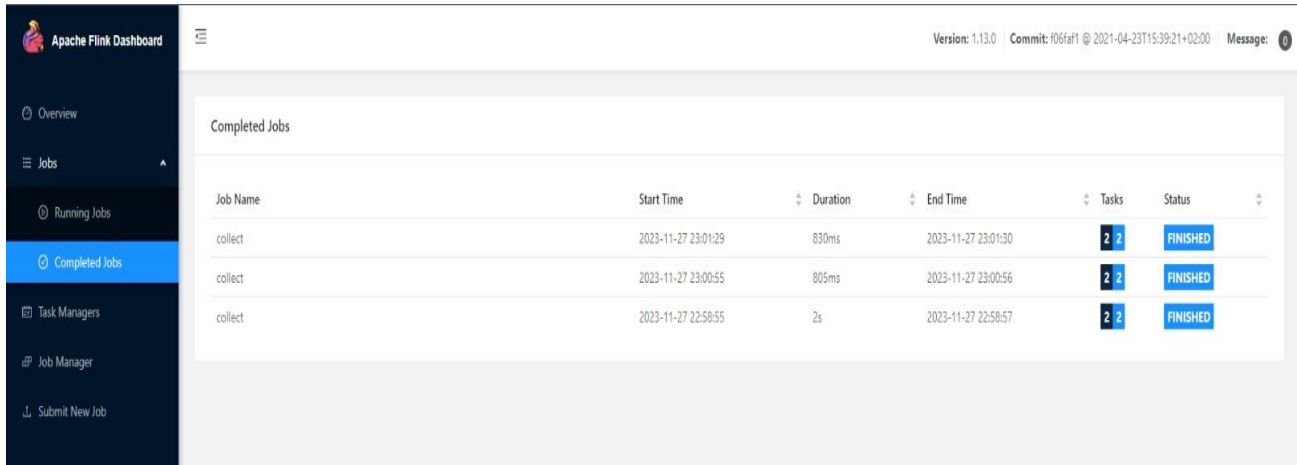
```
root@ip-172-31-18-182:/home/ubuntu/bigdata/Mahua Flink Playground# docker-compose exec jobmanager ./bin/flink run -py /opt/examples/employee-data-analysis.py
Job has been submitted with JobID 6149acd8f91e8a80f1c1e61a8a8974dc
```

employee_id	name	age
E001	John	45
E004	Emma	42
E003	Mike	32

```
3 rows in set
root@ip-172-31-18-182:/home/ubuntu/bigdata/Mahua Flink Playground#
```

The results show the analysis for all 3 examples. Also, you can check the status of the job that you ran on Apache Flink.

Navigate to <http://<<public-ip>>:8081> and click on Completed Jobs.



The screenshot shows the Apache Flink Dashboard interface. On the left is a dark sidebar with navigation links: Overview, Jobs (selected), Running Jobs, Completed Jobs (highlighted in blue), Task Managers, Job Manager, and Submit New Job. The main content area is titled 'Completed Jobs' and displays a table with the following data:

Job Name	Start Time	Duration	End Time	Tasks	Status
collect	2023-11-27 23:01:29	830ms	2023-11-27 23:01:30	2/2	FINISHED
collect	2023-11-27 23:00:55	805ms	2023-11-27 23:00:56	2/2	FINISHED
collect	2023-11-27 22:58:55	2s	2023-11-27 22:58:57	2/2	FINISHED

39. Great job ! You've finished this tutorial.