

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC - KỸ THUẬT MÁY TÍNH



HỆ ĐIỀU HÀNH

Báo cáo Seminar - Bài tập mở rộng

Thu thập dữ liệu người dùng từ hệ thống Moodle LMS (moodle.org) bằng Kafka

GVHD:	TS. Nguyễn Quang Hùng	
SV:	Nguyễn Tuấn Anh	1912600
	Mai Sỹ Chính	1912791
	Lê Văn Hoàng	1913429
	Trần Phương Tĩnh	1927038
	Trần Thanh Tuấn	2020114
	Nguyễn Duy Vũ	1915976

TP. HỒ CHÍ MINH, THÁNG 12/2021

Mục lục

1	Giới thiệu đề tài	5
2	LAMP	5
2.1	Giới thiệu về LAMP	5
2.2	Cài đặt LAMP	5
3	Moodle	8
3.1	Giới thiệu về Moodle	8
3.2	Cài đặt Moodle	9
3.3	Tạo tài khoản, khóa học và kiểm tra Logs Moodle	12
4	Kafka và Confluent Platform	17
4.1	Giới thiệu về Kafka và Confluent Platform	17
4.1.1	Kafka là gì?	17
4.1.2	Confluent Platform là gì?	17
4.2	Cài đặt Kafka và Confluent Platform	17
4.2.1	Confluent Platform	17
4.2.2	Zookeeper	20
4.2.3	Kafka	21
4.2.4	Control Center	22
4.2.5	Khởi động và dừng Confluent Platform	23
4.3	Quick start	24
4.3.1	Tạo Kafka topic thử nghiệm	24
4.3.2	Sinh dữ liệu giả	26
5	JDBC Connector cho Confluent Platform	30
5.1	Giới thiệu về JDBC Connector	30
5.2	Cài đặt JDBC Connector	30
5.3	Cài đặt JDBC Driver	30
5.4	Tạo Kafka topic	31
5.5	Thêm JDBC Connector vào Confluent Platform	34
6	Demo	40
6.1	Login	41
6.2	Logout	43
6.3	View course	45
6.4	View, start and submit quiz attempt	47
7	Tổng kết	50

Danh mục hình ảnh

2.1	Tải bản cài đặt LAMP 7.4.8 về thành công	5
2.2	Khởi chạy file cài đặt	5
2.3	Giao diện cài đặt LAMP Stack ban đầu	6
2.4	Chọn thư mục cài đặt	6
2.5	Đặt mật khẩu cho MySQL root account	6
2.6	Hoàn thành setting và bắt đầu cài đặt	7
2.7	Hoàn thành cài đặt và giao diện LAMP Stack	7
2.8	Giao diện trang Apache Web Server	8
2.9	Giao diện trang phpMyAdmin	8
3.1	Bản cài đặt Moodle 3.11.4	9
3.2	Sao chép bản cài đặt Moodle 3.11.4 vào thư mục htdocs của thư mục cài đặt LAMP Stack	9
3.3	Giao diện bắt đầu cài đặt Moodle	10
3.4	Các bước setting	10
3.5	Quá trình cài đặt	11
3.6	Giao diện trang Dashboard sau khi cài đặt hoàn tất	11
3.7	Tạo một khóa học	12
3.8	Tạo một tài khoản học sinh	13
3.9	Chọn khóa học muốn ghi danh cho tài khoản	13
3.10	Chọn tài khoản để ghi danh vào khóa học	14
3.11	Thao tác để xem Logs Moodle	14
3.12	Kết quả hiển thị Logs	15
3.13	Dữ liệu Logs được lưu trữ trong MySQL	15
3.14	Đăng nhập tài khoản học sinh và xem khóa học	16
3.15	Kiểm tra Logs hiển thị thông tin học sinh xem khóa học	16
3.16	Dữ liệu Logs đã được cập nhật trong database	17
4.1	Đăng ký tải Confluent Platform	18
4.2	Tải Confluent	18
4.3	Set biến môi trường	19
4.4	Thêm biến môi trường	19
4.5	Kiểm tra đã thêm biến môi trường thành công	19
4.6	Giải nén thư mục confluent sau khi tải về	20
4.7	Vào file zookeeper.properties	20
4.8	Cấu hình zookeeper	21
4.9	Vào file server.properties	21
4.10	Bỏ chú thích ở broker.id	21
4.11	Cấu hình zookeeper.connect	22
4.12	Control Center 1	22
4.13	Control Center 2	22
4.14	Thay đổi đường dẫn Control Center	22
4.15	Vào file server.properties	23
4.16	Cấu hình Kafka server	23
4.17	Vào file connect-distributed.properties	23
4.18	Thêm vào các dòng như trên	23
4.19	Khởi chạy Confluent Platform	24
4.20	Giao diện	24

4.21 Overview	25
4.22 Tạo topic	25
4.23 users topic	26
4.24 Xem lại các topic	26
4.25 Chọn connect-default	27
4.26 Chọn Add connector	27
4.27 Chọn Datagen Connector	27
4.28 Chỉnh sửa Name và Key coverter class	28
4.29 Chỉnh sửa kafka.topic, max.interval và quickstart	28
4.30 Chọn Next	28
4.31 Chọn Launch	29
4.32 Xem lại các Topic	29
4.33 Dữ liệu giả đã được sinh ra	29
4.34 Dữ liệu ở định dạng Avro	30
5.1 Cài đặt JDBC Connector	30
5.2 Downloads JDBC Driver	31
5.3 Giải nén file download	31
5.4 Copy file	31
5.5 Cài biến môi trường 1	32
5.6 Cài biến môi trường 2	32
5.7 Cài biến môi trường 3	32
5.8 Khởi động Confluent Platform	32
5.9 Tạo Kafka topic 1	33
5.10 Tạo Kafka topic 2	33
5.11 Tạo Kafka topic 3	34
5.12 Thêm JDBC Connector 1	34
5.13 Thêm JDBC Connector 2	35
5.14 Thêm JDBC Connector 3	35
5.15 Các bước cài đặt JDBC Connector 1	36
5.16 Các bước cài đặt JDBC Connector 2	37
5.17 Các bước cài đặt JDBC Connector 3	38
5.18 Cấu hình JDBC Connector	39
5.19 Cấu hình JDBC Connector	39
5.20 Kiểm tra kết nối JDBC Connector 1	40
5.21 Kiểm tra kết nối JDBC Connector 2	40
6.1 Bảng tdl_user	41
6.2 Bảng tdl_logstore_standard_log trước khi login	41
6.3 Confluent trước khi login	42
6.4 Bảng tdl_logstore_standard_log sau khi login	42
6.5 Confluent sau khi login	43
6.6 Bảng tdl_logstore_standard_log trước khi logout	43
6.7 Confluent trước khi logout	44
6.8 Bảng tdl_logstore_standard_log sau khi logout	44
6.9 Confluent sau khi logout	45
6.10 Bảng tdl_logstore_standard_log trước khi view course	45
6.11 Confluent trước khi view course	46
6.12 Bảng tdl_logstore_standard_log sau khi view course	46



6.13	Confluent sau khi view course	47
6.14	Bảng tdl_logstore_standard_log trước khi view, start and submit quiz attempt	47
6.15	Confluent trước khi view, start and submit quiz attempt	48
6.16	Bảng tdl_logstore_standard_log sau khi view, start and submit quiz attempt	49
6.17	Confluent sau khi view, start and submit quiz attempt (1)	49
6.18	Confluent sau khi view, start and submit quiz attempt (2)	50
6.19	Confluent sau khi view, start and submit quiz attempt (3)	50

1 Giới thiệu đề tài

Những hoạt động của người học trên trang Moodle có thể phản ánh khả năng học tập của họ, ví dụ như tần suất truy cập vào trang môn học nhiều và không bỏ lỡ bất kỳ video hay bài quiz nào thì người học này có thể đạt kết quả học tập tốt. Trong đề tài này, nhóm thực hiện việc thu thập dữ liệu là các hoạt động của người học ở trang Moodle, dữ liệu này được realtime streaming sang Kafka. Từ đó ta có thể xử lý dữ liệu và huấn luyện mô hình để dự đoán kết quả học tập của người học.

2 LAMP

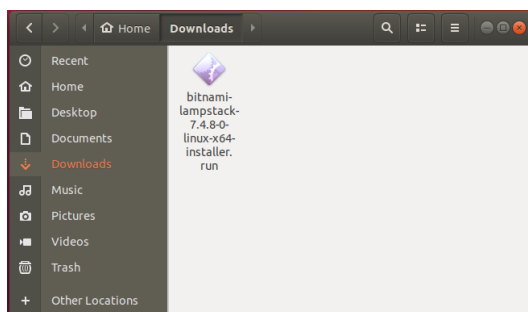
2.1 Giới thiệu về LAMP

LAMP là chữ viết tắt thường được dùng để chỉ sự sử dụng các phần mềm Linux, Apache, MySQL và ngôn ngữ văn lệnh PHP hay Perl hay Python để tạo nên một môi trường máy chủ Web có khả năng chứa và phân phối các trang Web động. Bốn phần mềm nói trên tạo thành một gói phần mềm LAMP. LAMP với lợi thế miễn phí, mã nguồn mở, đang đối đầu với thể độc quyền WISA (Windows-IIS-SQL Sever-ASP) của Microsoft vốn với giá khá cao.

Trong Bài tập Mở rộng này, chúng ta sẽ cài đặt Moodle trên một LAMP Server, do đó bước đầu tiên chúng ta cần làm là cài đặt LAMP.

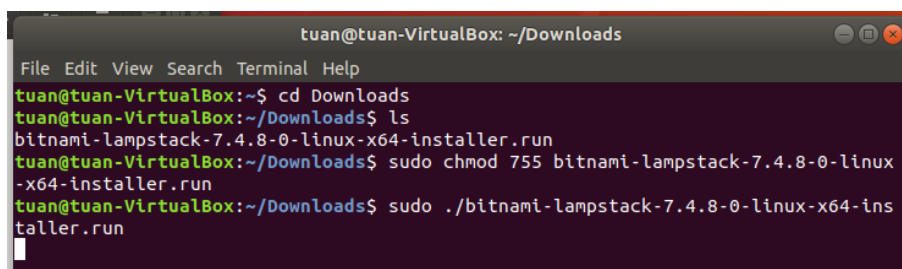
2.2 Cài đặt LAMP

Chúng ta sẽ sử dụng phiên bản LAMP 7.4.8 có MySQL 8.0.20, có thể tải bản cài đặt phiên bản này tại địa chỉ: <http://downloads.bitnami.com/files/stacks/lampstack/7.4.8-0/bitnami-lampstack-7.4.8-0-linux-x64-installer.run> (copy đường dẫn này dán vào thanh địa chỉ trình duyệt và tải về).



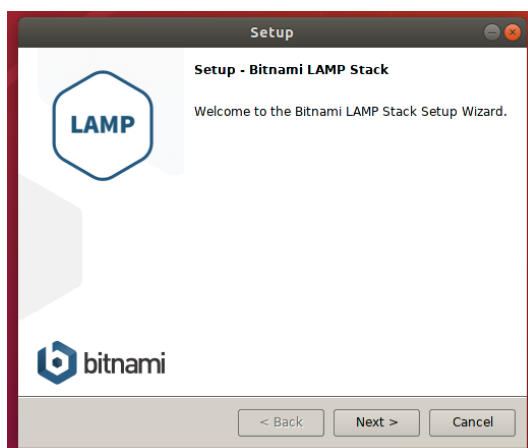
Hình 2.1: Tải bản cài đặt LAMP 7.4.8 về thành công

Sau khi tải bản cài đặt LAMP 7.4.8, ta tạo một thư mục để cài đặt LAMP Stack tại thư mục này (/home/tuan/lamp). Để thực hiện chạy bản cài đặt, ta thực hiện các lệnh như sau:



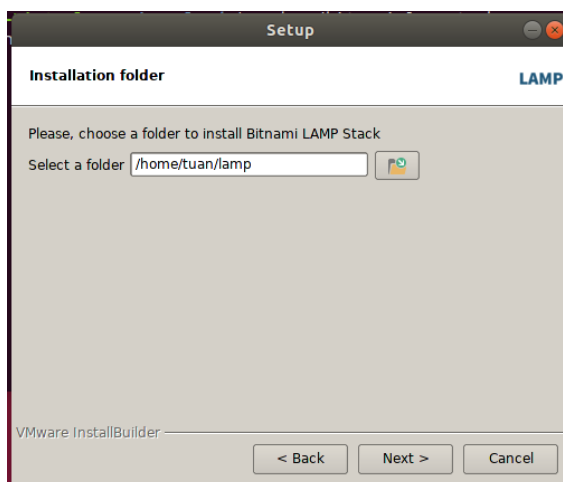
Hình 2.2: Khởi chạy file cài đặt

Sau khi khởi chạy file cài đặt, giao diện cài đặt LAMP Stack hiển thị ra như hình bên dưới:



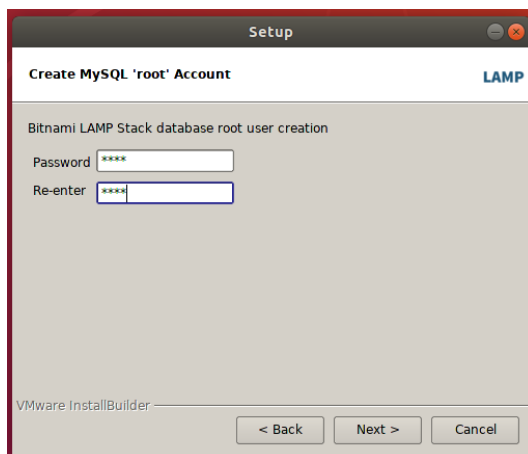
Hình 2.3: Giao diện cài đặt LAMP Stack ban đầu

Tại bước chọn thư mục cài đặt, ta chọn thư mục đã tạo trước đó (/home/tuan/lamp).



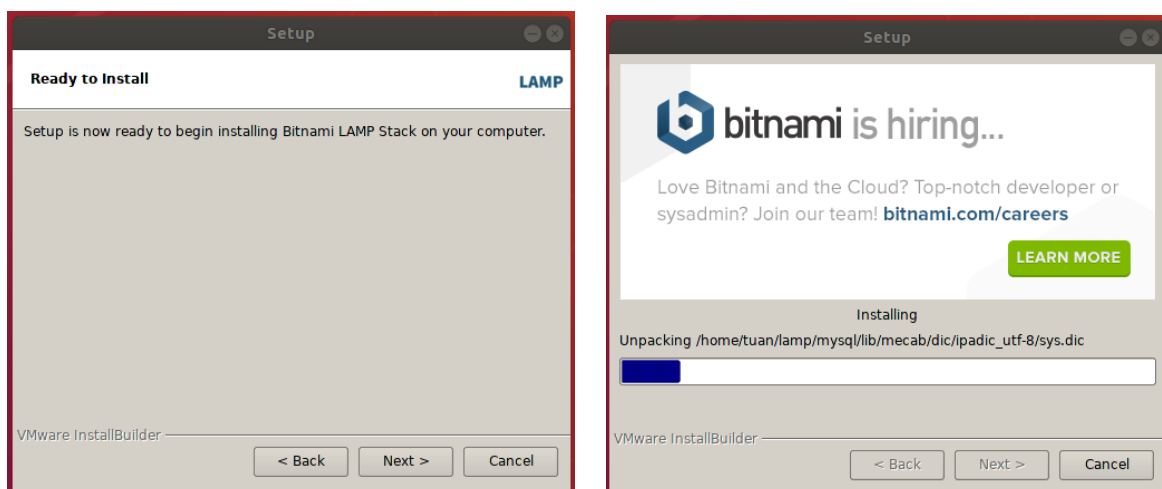
Hình 2.4: Chọn thư mục cài đặt

Tại bước đặt mật khẩu cho MySQL root account, nhập mật khẩu mà bạn mong muốn (bắt buộc và mật khẩu dài hơn hoặc bằng 6 ký tự).



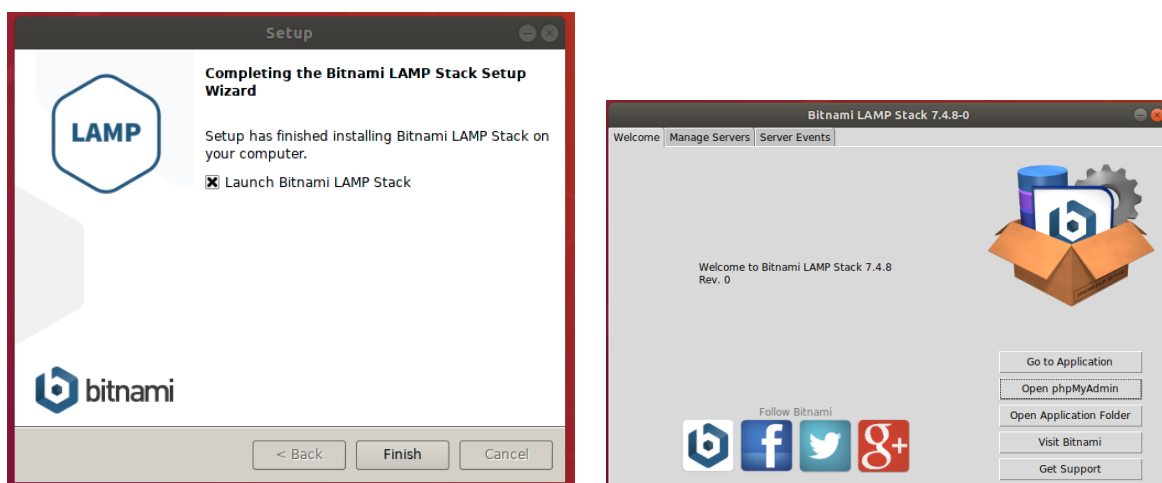
Hình 2.5: Đặt mật khẩu cho MySQL root account

Sau khi hoàn thành quá trình setting, ta chọn "Next" để bắt đầu quá trình cài đặt.



Hình 2.6: Hoàn thành setting và bắt đầu cài đặt

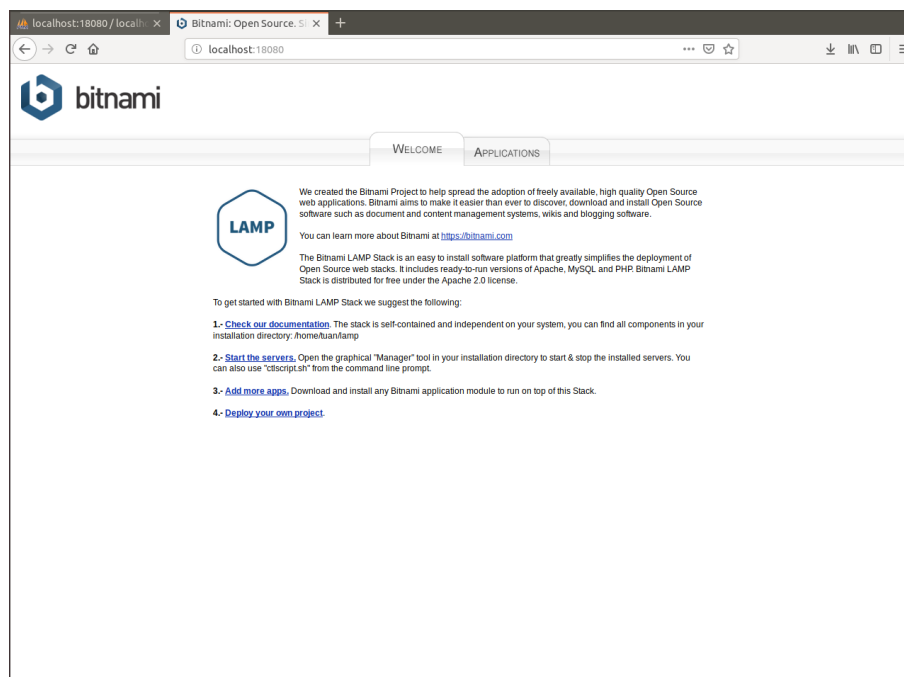
Sau khi cài đặt hoàn tất, ta chọn "Finish" và giao diện LAMP Stack được khởi động như bên dưới:



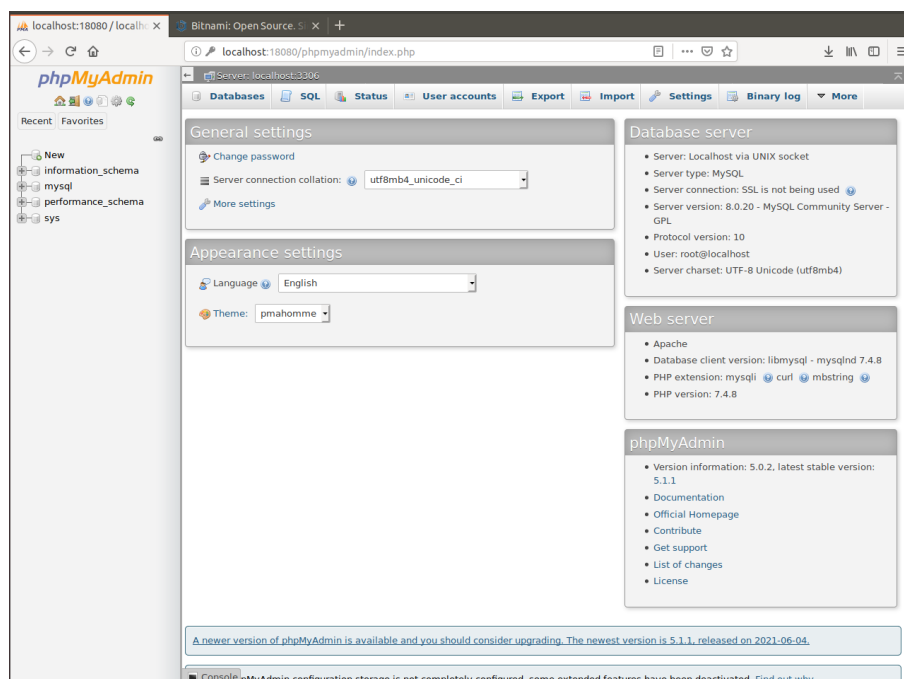
Hình 2.7: Hoàn thành cài đặt và giao diện LAMP Stack

Ta tiếp tục thực hiện tùy chỉnh port của Apache Web Server để tránh tình trạng xung đột/trùng port (nếu không trùng có thể sử dụng port mặc định 80), ta thực hiện như sau: tại mục Manage Servers, ta chọn "Configure" của Apache Web Server, mở config ("Open Conf File") chỉnh Listen 18080 và ServerName localhost:18080 (điều chỉnh port cao để tránh xung đột).

Để kiểm tra quá trình cài đặt đã thành công chưa, ta kiểm tra bằng cách truy cập vào trang Apache Web Server bằng địa chỉ: <http://localhost:18080/> và trang phpMyAdmin bằng địa chỉ: <http://localhost:18080/phpmyadmin/> (tại lần đầu truy cập sẽ yêu cầu nhập tên tài khoản và mật khẩu, nhập tài khoản là "root" và mật khẩu đã đặt trước đó khi cài đặt). Kết quả như sau:



Hình 2.8: Giao diện trang Apache Web Server



Hình 2.9: Giao diện trang phpMyAdmin

Vậy ta đã hoàn thành bước cài đặt LAMP Stack để chuẩn bị cài đặt Moodle.

3 Moodle

3.1 Giới thiệu về Moodle

Moodle là một hệ thống quản lý học tập (Learning Management System - LMS hoặc người ta còn gọi là Course Management System hoặc VLE - Virtual Learning Environment) mã nguồn mở (do đó miễn

phí và có thể chỉnh sửa được mã nguồn), cho phép tạo các khóa học trên mạng Internet hay các website học tập trực tuyến.

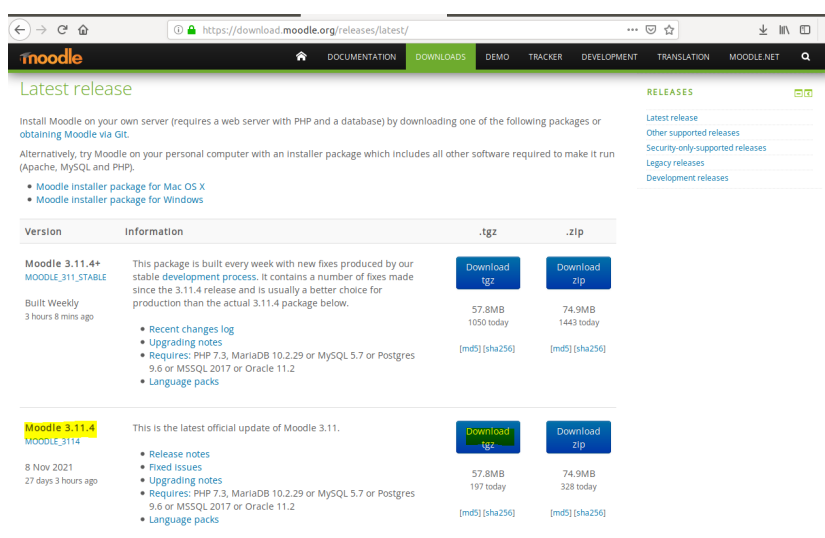
Moodle (viết tắt của Modular Object-Oriented Dynamic Learning Environment) được sáng lập năm 1999 bởi Martin Dougiamas, người tiếp tục điều hành và phát triển chính của dự án. Moodle nổi bật là thiết kế hướng tới giáo dục, dành cho những người làm trong lĩnh vực giáo dục. Moodle rất dễ dùng với giao diện trực quan, giáo viên chỉ mất một thời gian ngắn để làm quen và có thể sử dụng thành thạo. Giáo viên có thể tự cài và nâng cấp Moodle. Do thiết kế dựa trên module nên Moodle cho phép bạn chỉnh sửa giao diện bằng cách dùng các theme có trước hoặc tạo thêm một theme mới cho riêng mình.

Tài liệu hỗ trợ của Moodle rất đồ sộ và chi tiết, khác hẳn với nhiều dự án mã nguồn mở khác. Moodle phát triển dựa trên PHP (Ngôn ngữ được dùng bởi các công ty Web lớn như Yahoo, Flickr, Baidu, Digg, CNET) có thể mở rộng từ một lớp học nhỏ đến các trường đại học lớn trên 50 000 sinh viên (ví dụ đại học Open PolyTechnique của Newzealand hoặc sắp tới đây là đại học mở Anh - Open University of UK, trường đại học cung cấp đào tạo từ xa lớn nhất châu Âu, và đại học mở Canada, Athabasca University). Bạn có thể dùng Moodle với các database mã nguồn mở như MySQL hoặc PostgreSQL. Phiên bản 1.7 sẽ hỗ trợ thêm các database thương mại như Oracle, Microsoft SQL để các bạn có thêm nhiều cơ hội lựa chọn.

Trong Bài tập Mở rộng này, ta sẽ tập trung vào Logging System của Moodle, cụ thể là việc lưu trữ nhật ký/log về việc truy cập vào khóa học của tài khoản học sinh/sinh viên.

3.2 Cài đặt Moodle

Chúng ta sẽ sử dụng phiên bản Moodle 3.11.4, có thể tải bản cài đặt phiên bản này [tại đây](#) (file .tgz).



Hình 3.1: Bản cài đặt Moodle 3.11.4

Sau khi tải bản cài đặt, ta sao chép bản cài đặt vào thư mục **htdocs** của thư mục cài đặt LAMP Stack như sau:

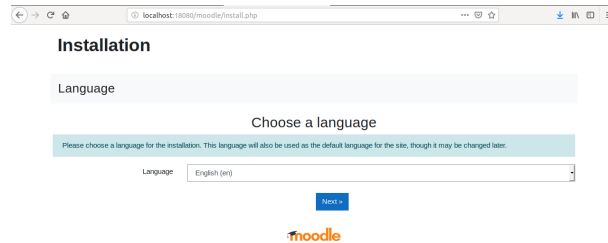
```
tuan@tuan-VirtualBox:~/Downloads$ ls
bitnami-lampstack-7.4.8-0-linux-x64-installer.run moodle-3.11.4.tgz
tuan@tuan-VirtualBox:~/Downloads$ sudo cp moodle-3.11.4.tgz /home/tuan/lamp/apache2/htdocs/
[sudo] password for tuan:
tuan@tuan-VirtualBox:~/Downloads$ cd /home/tuan/lamp/apache2/htdocs/
tuan@tuan-VirtualBox:~/lamp/apache2/htdocs$ ls
503.html bitnami.css favicon.ico img index.html moodle-3.11.4.tgz
tuan@tuan-VirtualBox:~/lamp/apache2/htdocs$
```

Hình 3.2: Sao chép bản cài đặt Moodle 3.11.4 vào thư mục **htdocs** của thư mục cài đặt LAMP Stack

Sau đó chúng ta giải nén file cài đặt bằng lệnh `sudo tar xvf moodle-3.11.4.tgz` để chuẩn bị cho quá trình cài đặt Moodle.

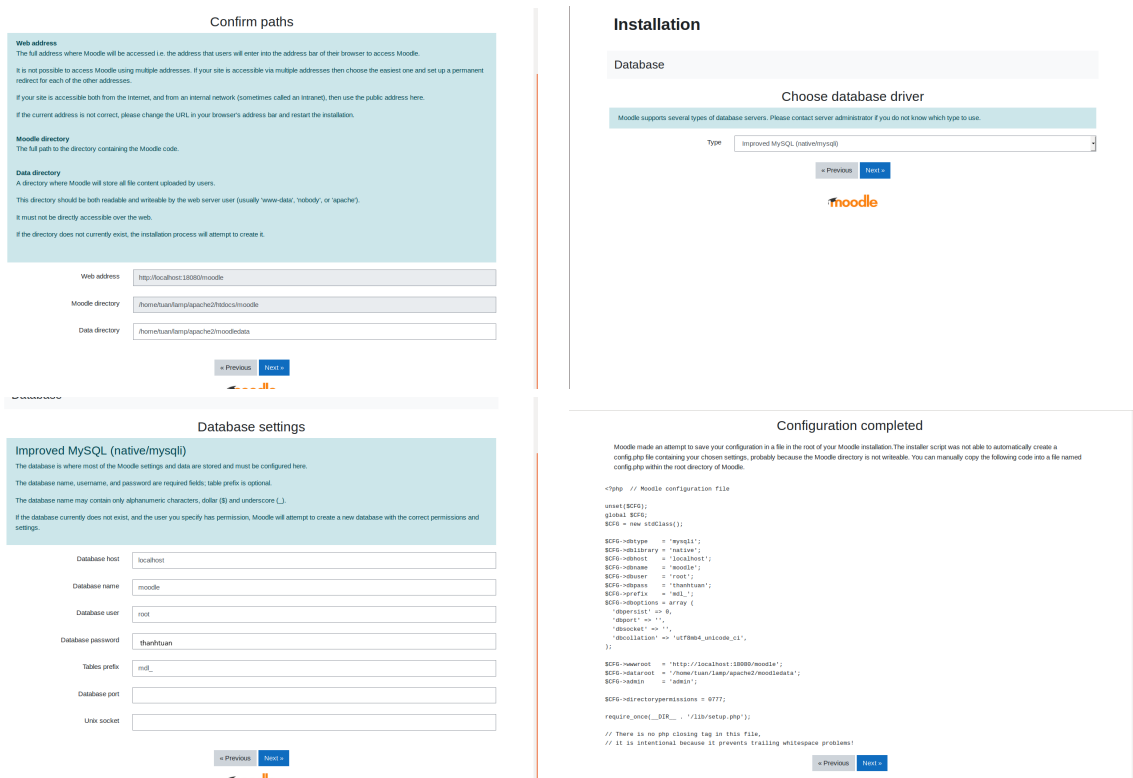
Ngoài ra, để tránh lỗi trong quá trình cài đặt, ta cần cài php-sodium (`sudo apt install php-sodium` (Ubuntu 20.04) hoặc `sudo apt install php-libsodium` (Ubuntu 18.04)). Sau đó tại file php.ini của LAMP Stack (.../php/etc/php.ini), ta điều chỉnh `max_input_vars = 5000`.

Để bắt đầu quá trình cài đặt, ta truy cập đường dẫn <http://localhost:18080/moodle>, khi đó sẽ tự động bắt đầu quá trình cài đặt như sau:



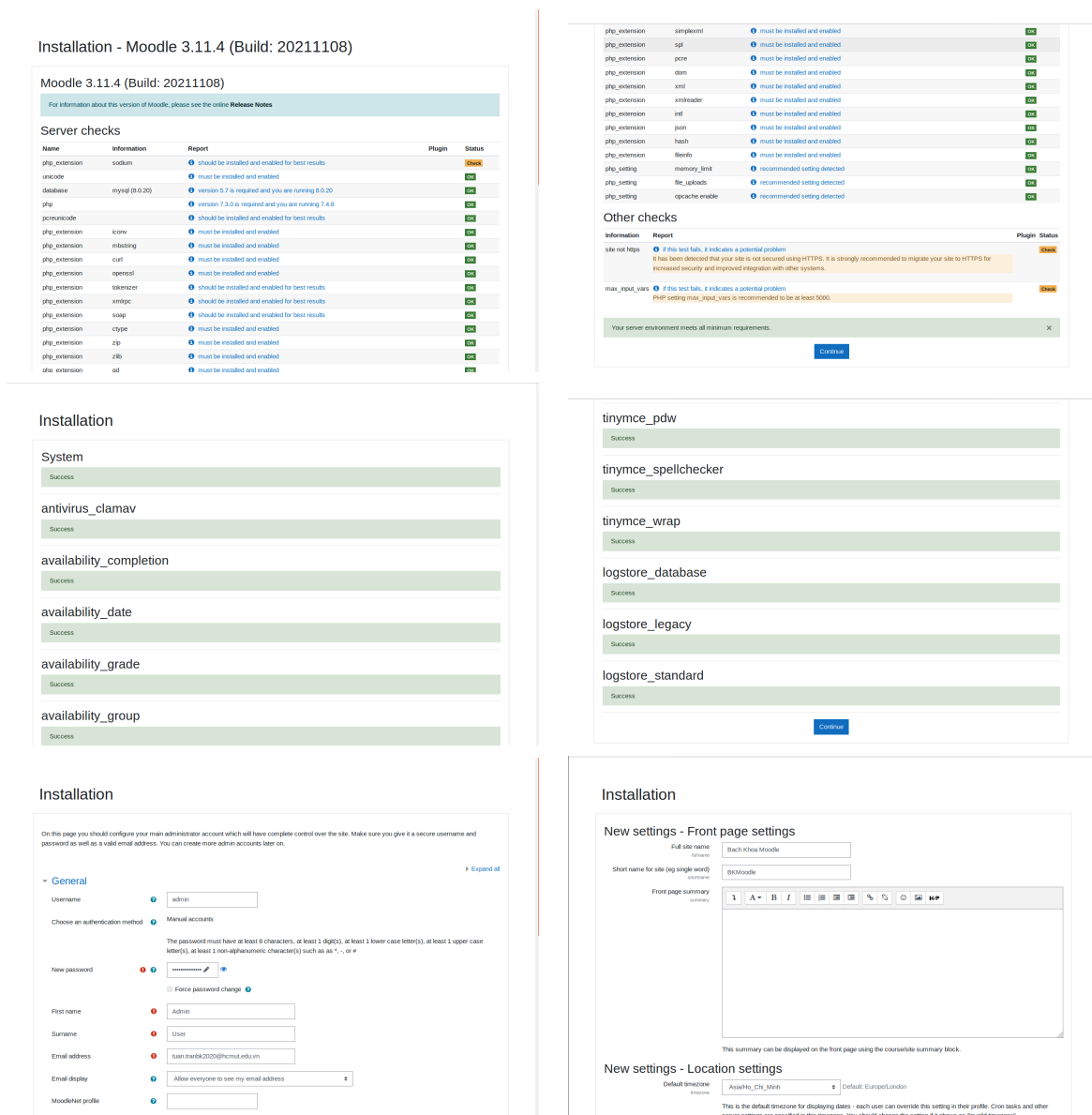
Hình 3.3: Giao diện bắt đầu cài đặt Moodle

Sau đó ta thực hiện tuần tự các bước setting như các hình ảnh dưới đây.



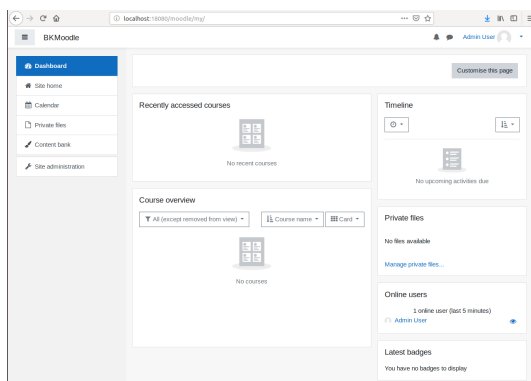
Hình 3.4: Các bước setting

Tại hình "Database Setting", nhập các thông tin như trong hình (Database User là "root" và Database password là mật khẩu đã thiết lập khi cài đặt LAMP), sau đó tại bước "Configuration completed", hệ thống sẽ yêu cầu chúng ta tạo file config.php trong thư mục moodle (.../htdocs/moodle/) với nội dung như trong hình, ta thực hiện tạo và chọn "Next" để tiếp tục. Hoàn thành các bước setting, tiếp theo đó sẽ là quá trình cài đặt như sau:



Hình 3.5: Quá trình cài đặt

Sau khi hoàn tất toàn bộ các bước cài đặt, hệ thống tự động chuyển đến trang Dashboard thể hiện đã cài đặt thành công.



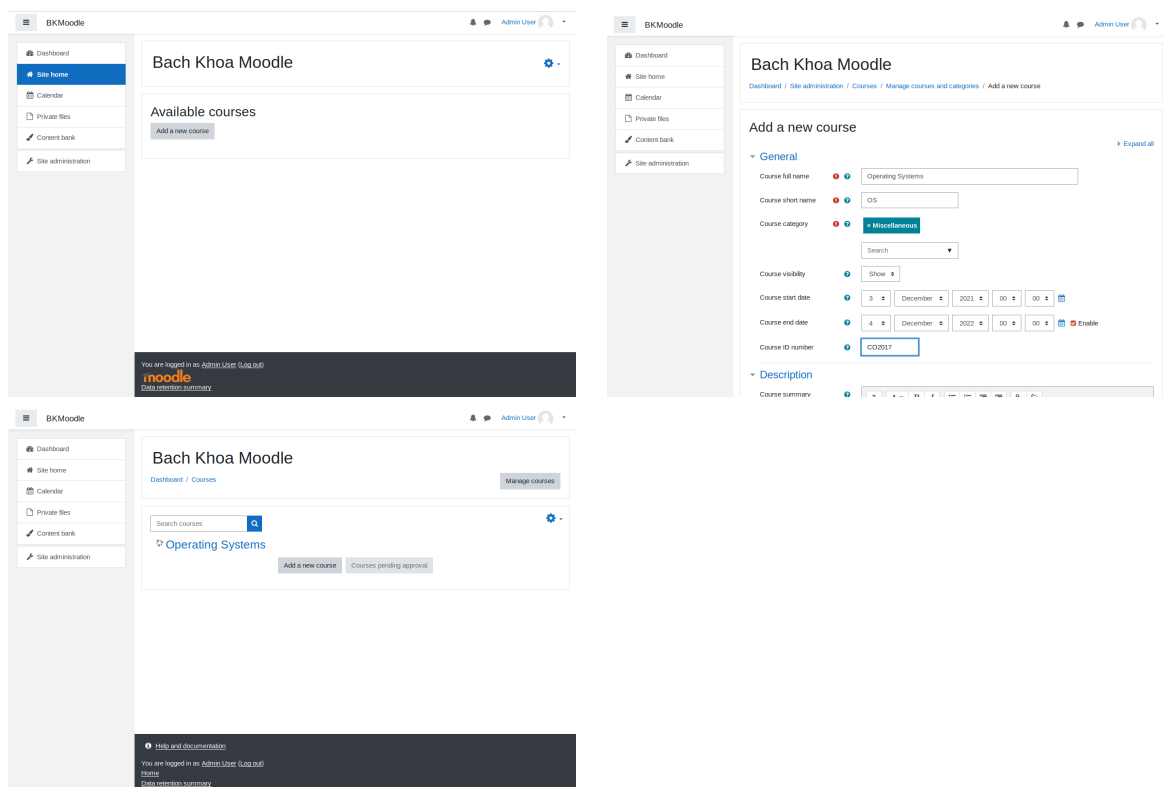
Hình 3.6: Giao diện trang Dashboard sau khi cài đặt hoàn tất

Hiện giờ, để có thể đăng nhập vào trang Moodle, ta chỉ cần truy cập đường dẫn <http://localhost:18080/moodle>.

3.3 Tạo tài khoản, khóa học và kiểm tra Logs Moodle

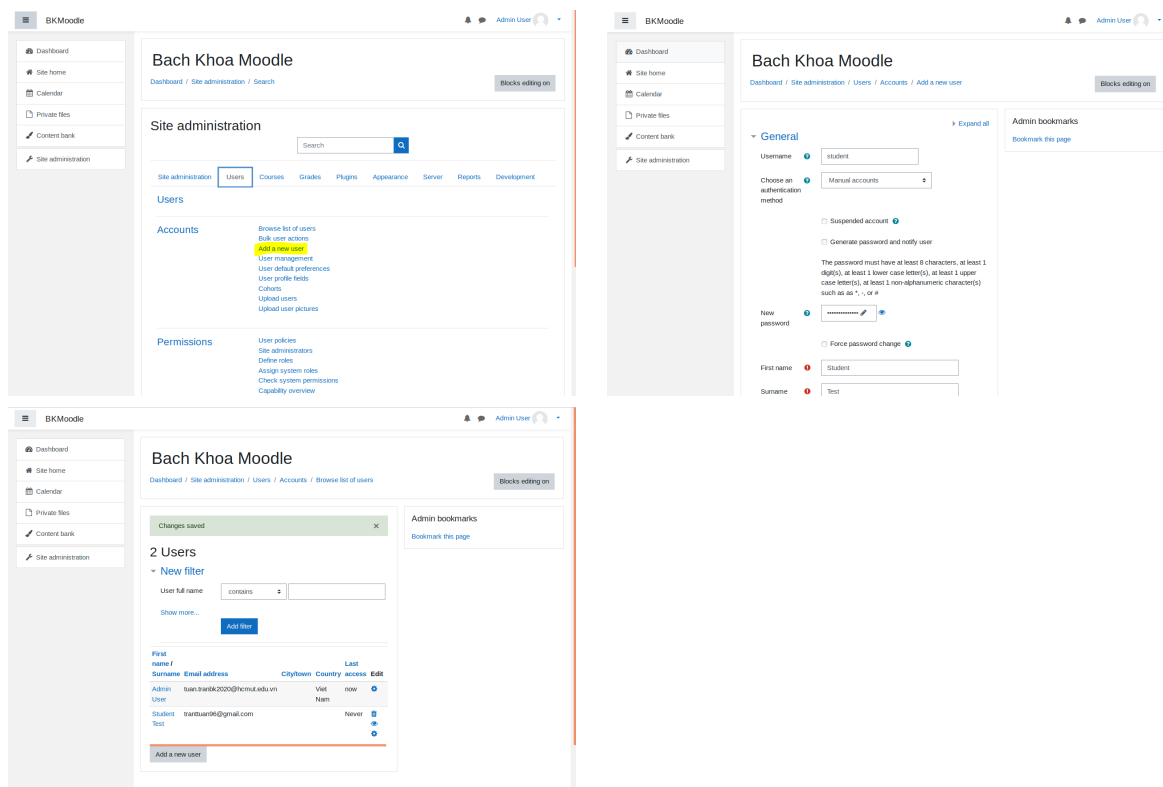
Để tìm hiểu Logging System của Moodle, cụ thể là việc lưu trữ nhật ký/log về việc truy cập vào khóa học của tài khoản học sinh/sinh viên, ta sẽ thực hiện tạo khóa học, tài khoản học sinh và kiểm tra Logs khi tài khoản này truy cập khóa học.

Đầu tiên, để tạo khóa học, ta vào **Site home** -> **Add a new course**. Sau đó điền các thông tin của khóa học muốn tạo, sau đó chọn **Save and return**, hệ thống sẽ quay về trang **Site home** và khóa học vừa thêm sẽ hiển thị.



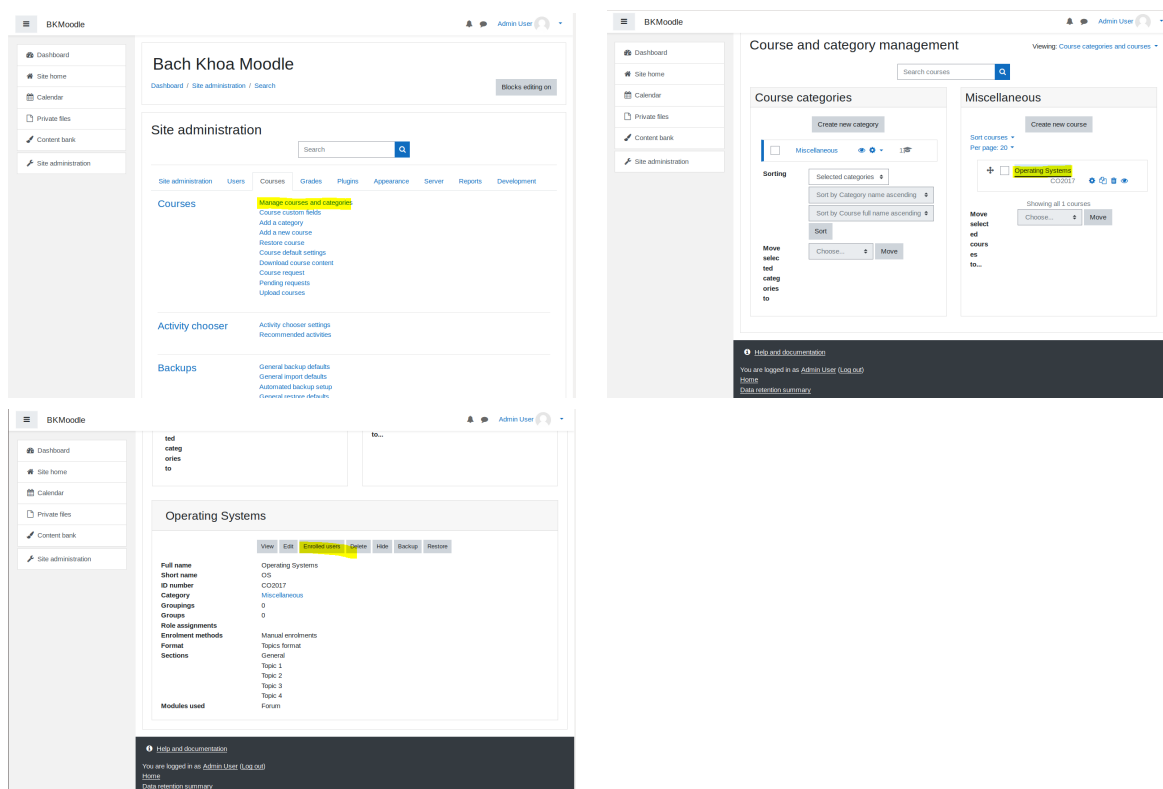
Hình 3.7: Tạo một khóa học

Tiếp theo, để tạo một tài khoản học sinh, ta vào **Site administration**, mục **User** -> **Add a new user**. Sau đó điền các thông tin của tài khoản học sinh muốn tạo, sau đó chọn **Create User**, hệ thống sẽ chuyển đến trang **Create User** và hiển thị danh sách các tài khoản, ta thấy có tài khoản Admin và tài khoản học sinh vừa thêm.



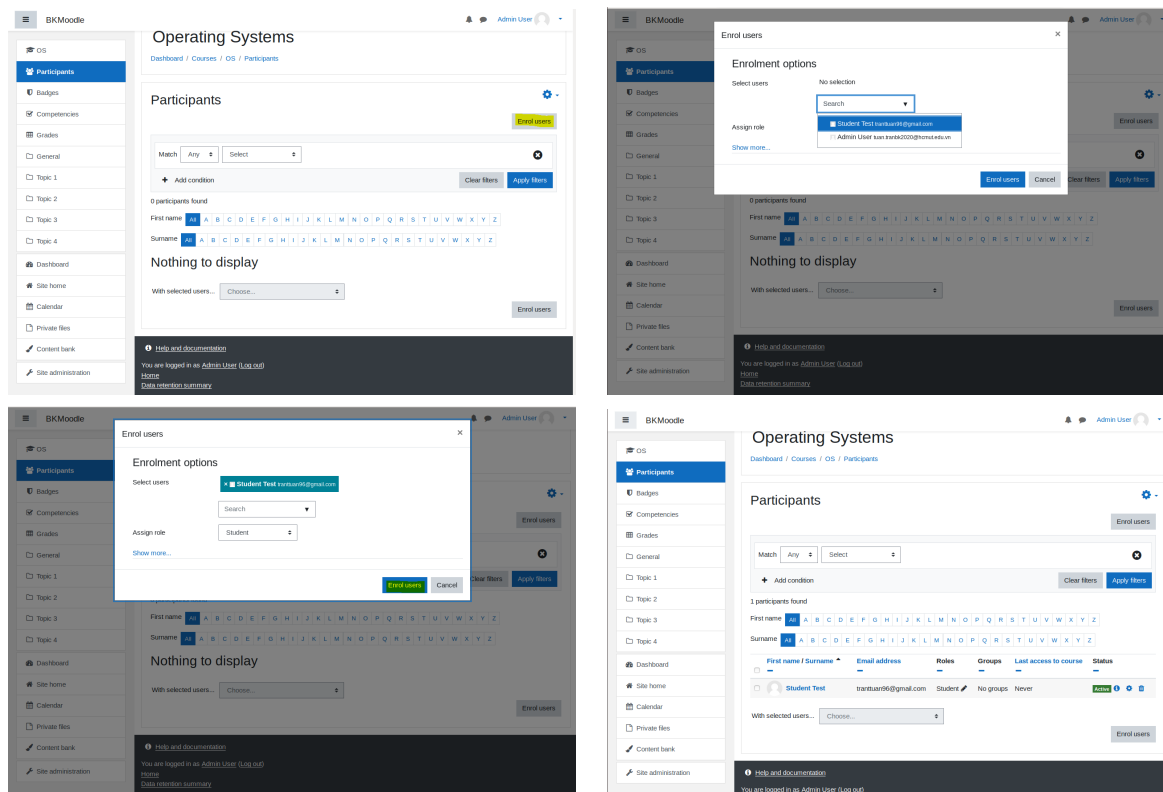
Hình 3.8: Tạo một tài khoản học sinh

Kế đến, ta ghi danh cho tài khoản học sinh này vào khóa học đã tạo, ta vào **Site administration**, mục **Courses** -> **Manage courses and categories**.



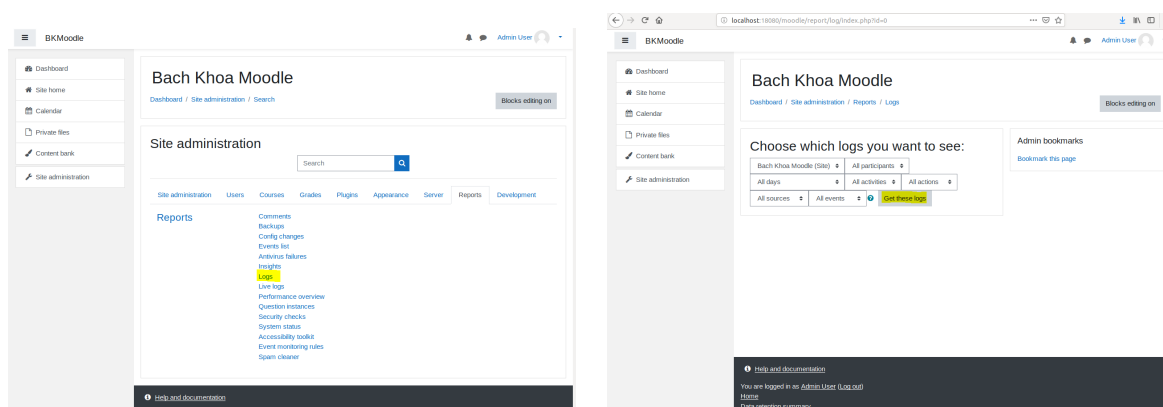
Hình 3.9: Chọn khóa học muốn ghi danh cho tài khoản

Tại giao diện này ta sẽ thấy có khóa học vừa tạo, ta nhấn chọn vào khóa học, sau đó kéo xuống dưới cùng sẽ có thông tin chi tiết của khóa học, tại đó, ta chọn mục **Enrolled Users**. Giao diện sẽ chuyển đến trang hiển thị các tài khoản đã ghi danh khóa học, hiện giờ khóa học này chưa có tài khoản nào, ta chọn **Enrol users** để ghi danh tài khoản, giao diện hiển thị pop-up để chọn tài khoản, ta chọn tài khoản học sinh vừa tạo và nhấn nút **Enrol users** để ghi danh, sau đó ta thấy tài khoản trên đã được ghi danh vào khóa học.



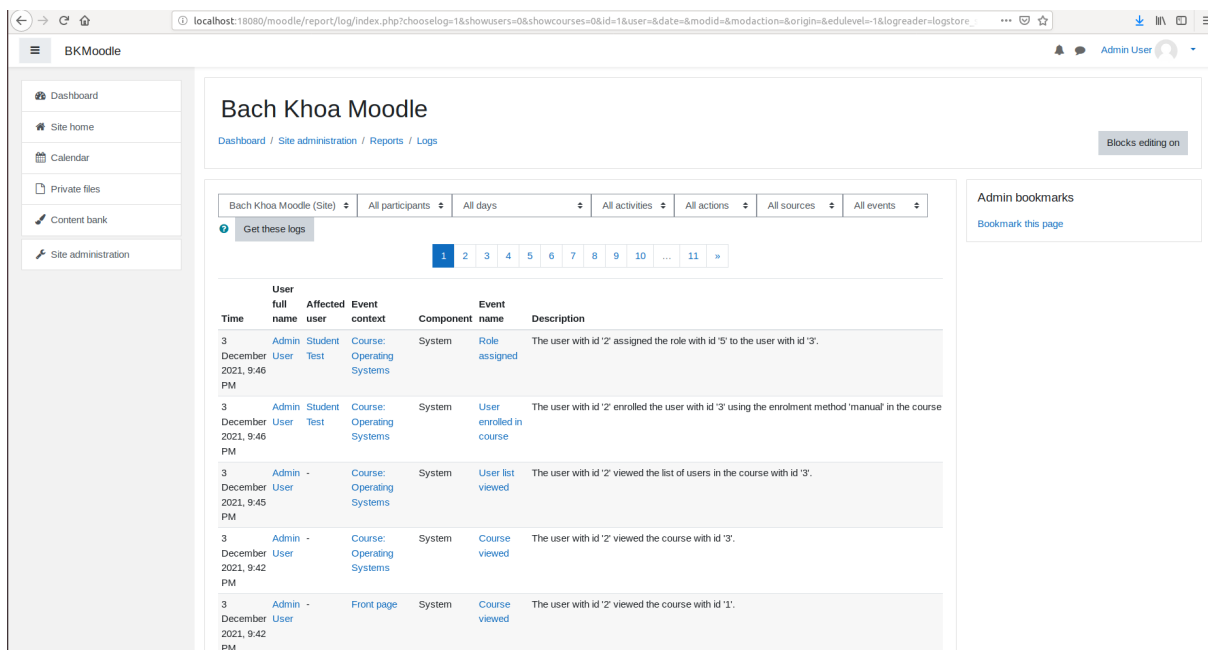
Hình 3.10: Chọn tài khoản để ghi danh vào khóa học

Bây giờ chúng ta sẽ thử kiểm tra nhật ký/Logs của Moodle, để kiểm tra ta vào **Site administration**, mục **Reports -> Logs**. Tại đây chúng ta chọn **Get these logs** để xem toàn bộ nhật ký.



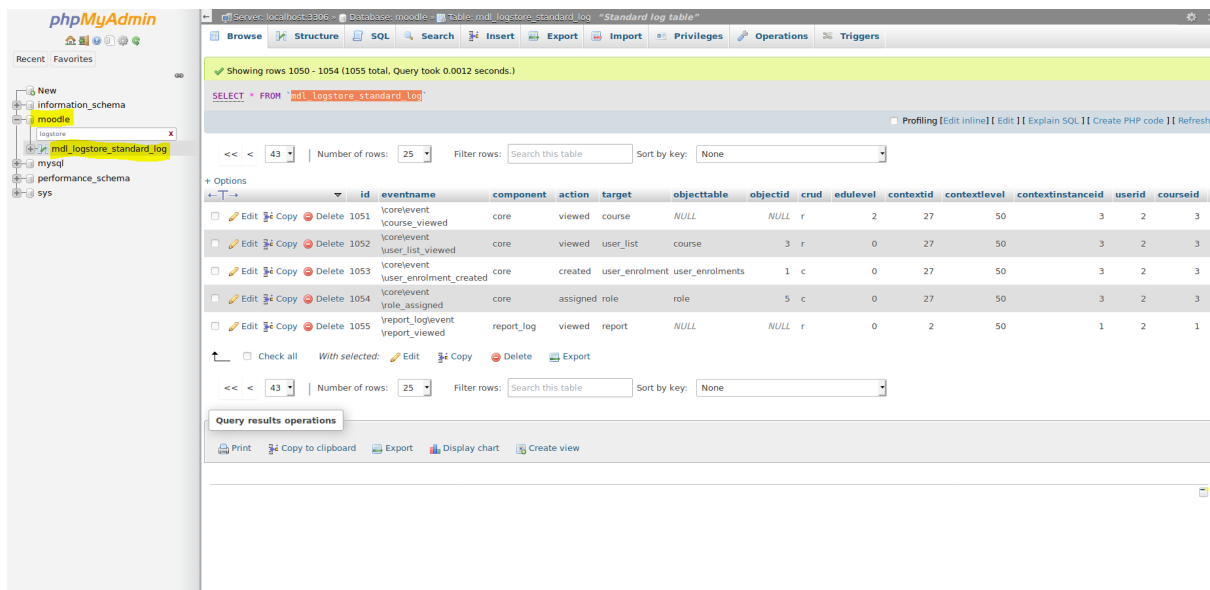
Hình 3.11: Thao tác để xem Logs Moodle

Dưới đây là kết quả hiển thị của Logs, ta thấy như trong hình có các hoạt động như ghi danh tài khoản vào khóa học, xem danh sách tài khoản,... như ta đã làm.



Hình 3.12: Kết quả hiển thị Logs

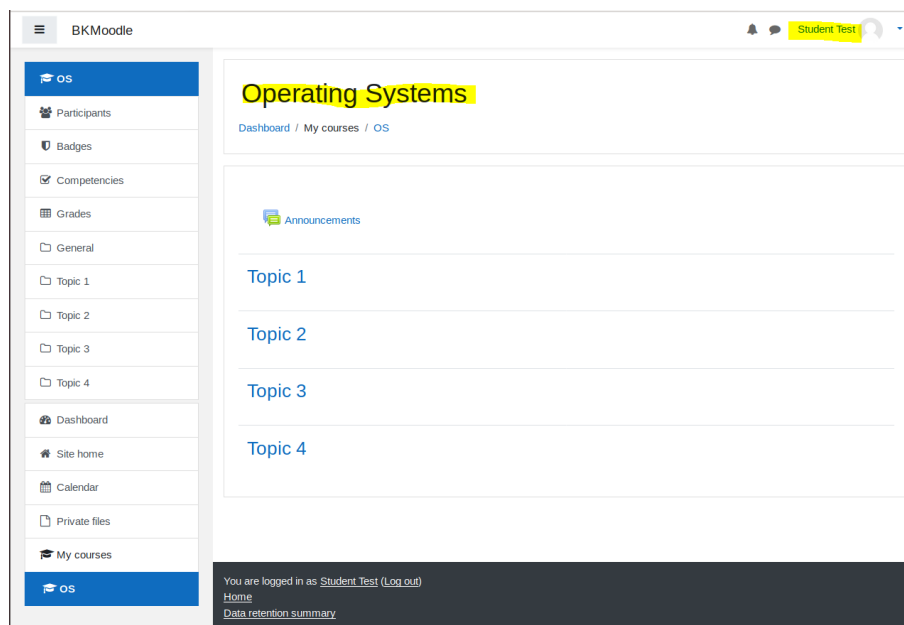
Trong MySQL, các dữ liệu về nhật ký được lưu trong **database moodle** như đã thiết lập lúc cài đặt, và trong bảng **logstore_standard_log**, ta có thể kiểm tra tại phpMyAdmin như trong hình ảnh sau:



Hình 3.13: Dữ liệu Logs được lưu trữ trong MySQL

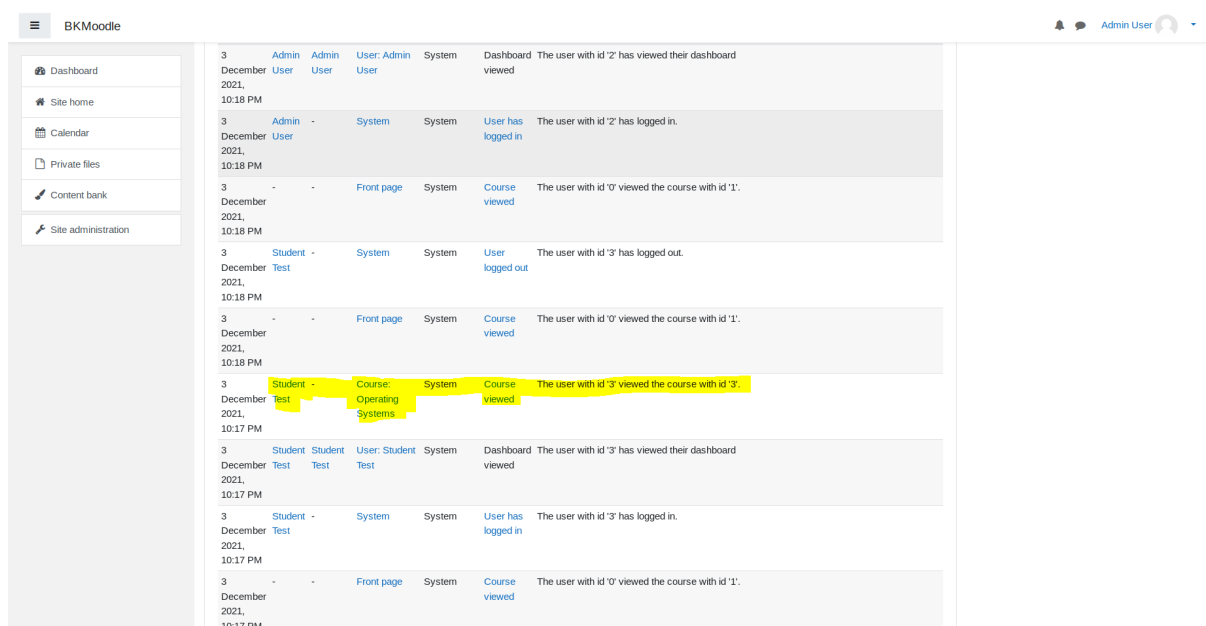
Cuối cùng, ta sẽ thử đăng nhập vào tài khoản học sinh vừa tạo và truy cập vào khóa học, sau đó kiểm tra lại Logs để xem thông tin có được ghi nhận không.

Ta truy cập tài khoản học sinh và truy cập vào khóa học như hình ảnh dưới đây:



Hình 3.14: Đăng nhập tài khoản học sinh và xem khóa học

Sau đó, ta đăng nhập lại bằng tài khoản admin và thực hiện kiểm tra Logs, kết quả như hình bên dưới, ta thấy Logs đã ghi nhận và hiển thị thông tin tài khoản học sinh *Student Test* đã truy cập/xem khóa học *Operating Systems*:



Hình 3.15: Kiểm tra Logs hiển thị thông tin học sinh xem khóa học

Ta cũng thử kiểm tra trong database của MySQL, và nhận thấy dữ liệu cũng đã được cập nhật như hình ảnh bên dưới:

id	eventname	component	action	target	objecttable	objectid	crud	edulevel	contextid	contextlevel	contextinstanceid	userid	courseid	relateduserid	anonymos
1053	!core!event	core	created	user_enrolment	user_enrolments	1	c	0	27	50	3	2	3	3	0
1054	!core!event	core	assigned	role	role	5	c	0	27	50	3	2	3	3	0
1055	!core!event	report_log	viewed	report	NULL	NULL	r	0	2	50	1	2	1	0	0
1056	!core!event	core	loggedout	user	user	2	r	0	1	10	0	2	0	NULL	0
1057	!core!event	core	viewed	course	NULL	NULL	r	2	2	50	1	0	1	NULL	0
1058	!core!event	core	loggedin	user	user	3	r	0	1	10	0	3	0	NULL	0
1059	!core!event	core	viewed	dashboard	NULL	NULL	r	0	28	30	3	3	0	3	0
1060	!core!event	core	viewed	course	NULL	NULL	r	2	27	50	3	3	3	NULL	0
1061	!core!event	core	viewed	course	NULL	NULL	r	2	2	50	1	0	1	NULL	0
1062	!core!event	core	loggedout	user	user	3	r	0	1	10	0	3	0	NULL	0
1063	!core!event	core	viewed	course	NULL	NULL	r	2	2	50	1	0	1	NULL	0
1064	!core!event	core	loggedin	user	user	2	r	0	1	10	0	2	0	NULL	0
1065	!core!event	core	viewed	dashboard	NULL	NULL	r	0	5	30	2	2	0	2	0
1066	!core!event	report_log	viewed	report	NULL	NULL	r	0	2	50	1	2	1	0	0

Hình 3.16: Dữ liệu Logs đã được cập nhật trong database

Vậy ta đã tìm hiểu về Moodle, cài đặt Moodle trên LAMP Server, và tìm hiểu Logs của Moodle được lưu trữ như thế nào.

4 Kafka và Confluent Platform

4.1 Giới thiệu về Kafka và Confluent Platform

4.1.1 Kafka là gì?

Kafka là nền tảng streaming phân tán, có thể mở rộng và là sản phẩm mã nguồn mở. Dự án Kafka ban đầu được phát triển bởi LinkedIn sau đó trở thành dự án Apache mã nguồn mở vào năm 2011. Kafka được viết bằng ngôn ngữ Scala và Java. Nó được viết ra nhằm mục đích cung cấp một nền tảng mà có độ trễ thấp và thông lượng cao cho việc xử lý các nguồn cấp dữ liệu theo thời gian thực.

4.1.2 Confluent Platform là gì?

Confluent được tạo ra bởi những người sáng lập ra Apache Kafka. Confluent Platform giúp chúng ta xây dựng các ứng dụng streaming một cách đơn giản bằng cách tích hợp dữ liệu từ rất nhiều nguồn tới một nơi tập trung đó là Kafka.

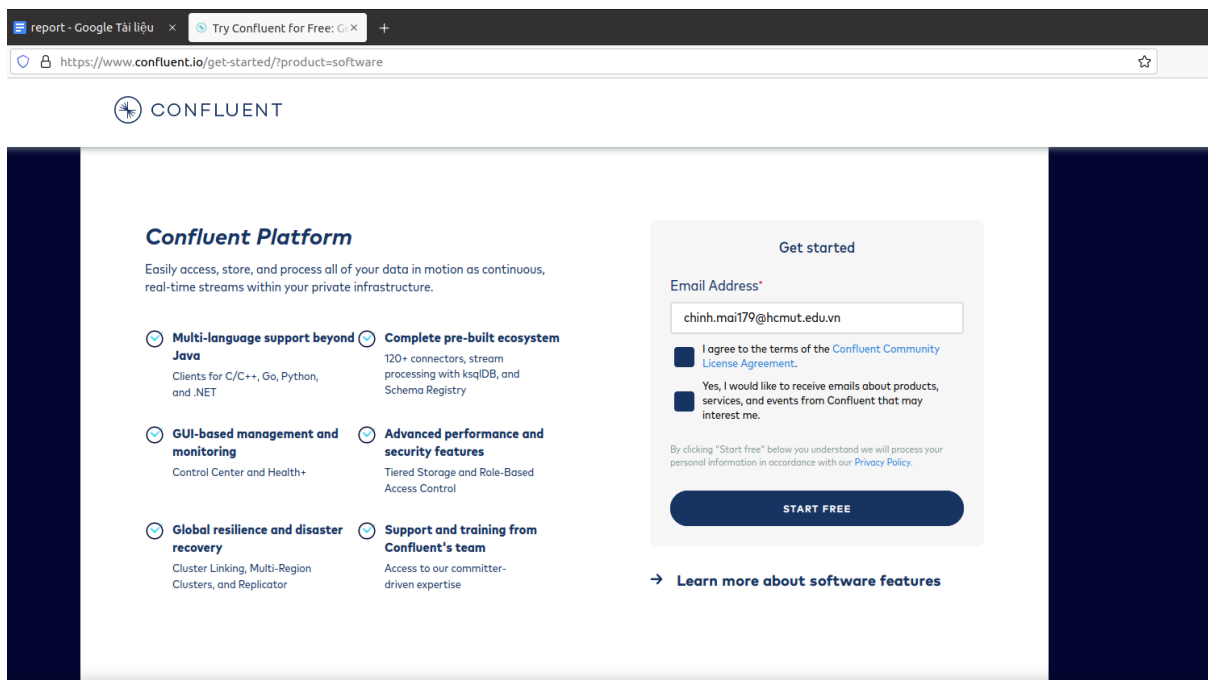
Confluent Platform giúp cho việc xây dựng các data pipeline thời gian thực và các ứng dụng streaming trở nên dễ dàng hơn bằng cách tổng hợp dữ liệu từ nhiều nguồn, tại nhiều vị trí vào một platform streamming dữ liệu duy nhất. Confluent Platform cho phép bạn tập trung vào các logic nghiệp vụ hơn, thay vì lo lắng về các cơ chế cơ bản như cách dữ liệu được vận chuyển hoặc sự tương tác giữa các hệ thống khác nhau. Cụ thể, Confluent Platform đơn giản hóa việc kết nối các nguồn dữ liệu với Kafka, xây dựng các ứng dụng với Kafka, cũng như bảo mật, giám sát và quản lý cơ sở hạ tầng Kafka của bạn.

4.2 Cài đặt Kafka và Confluent Platform

4.2.1 Confluent Platform

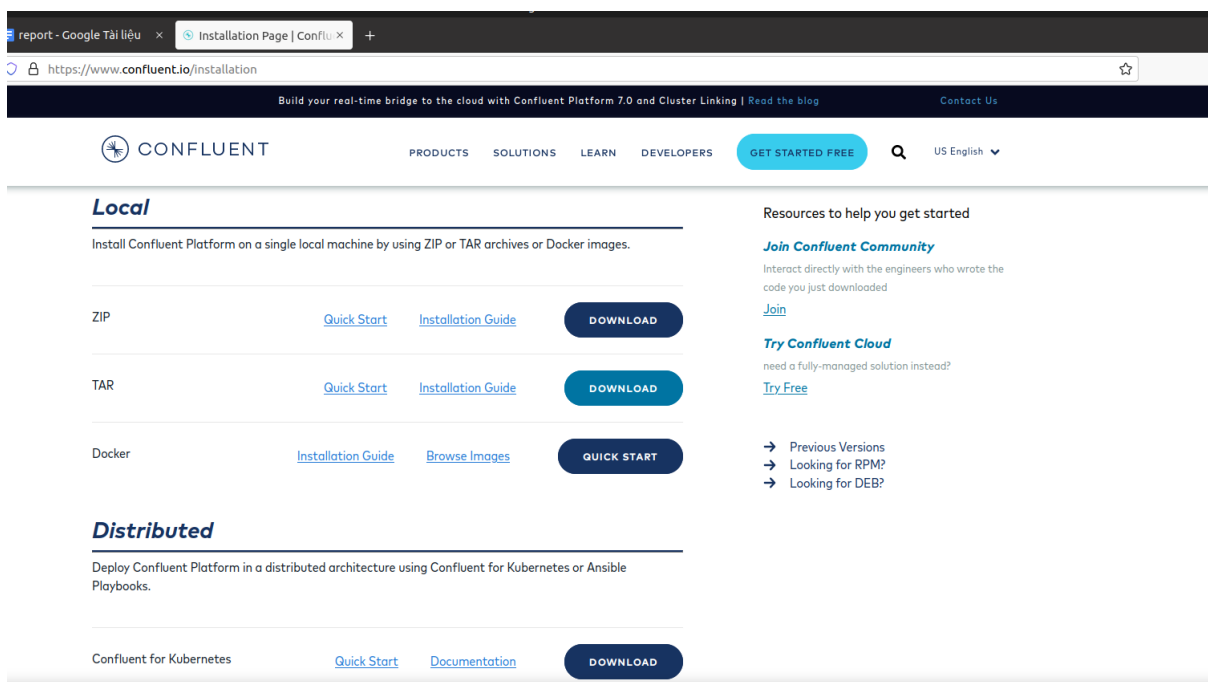
- Cài Java và Javac: [Java installation](#)

- Cài Confluent: [Confluent Platform installation](#)



Hình 4.1: Đăng kí tải Confluent Platform

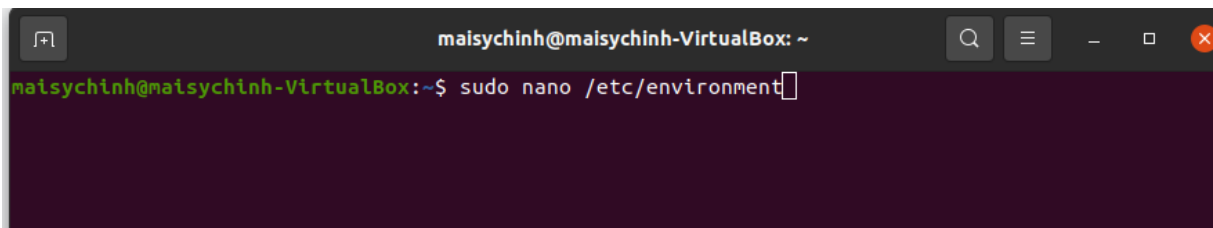
- Tải về file tar hoặc zip ở mục Local



Hình 4.2: Tải Confluent

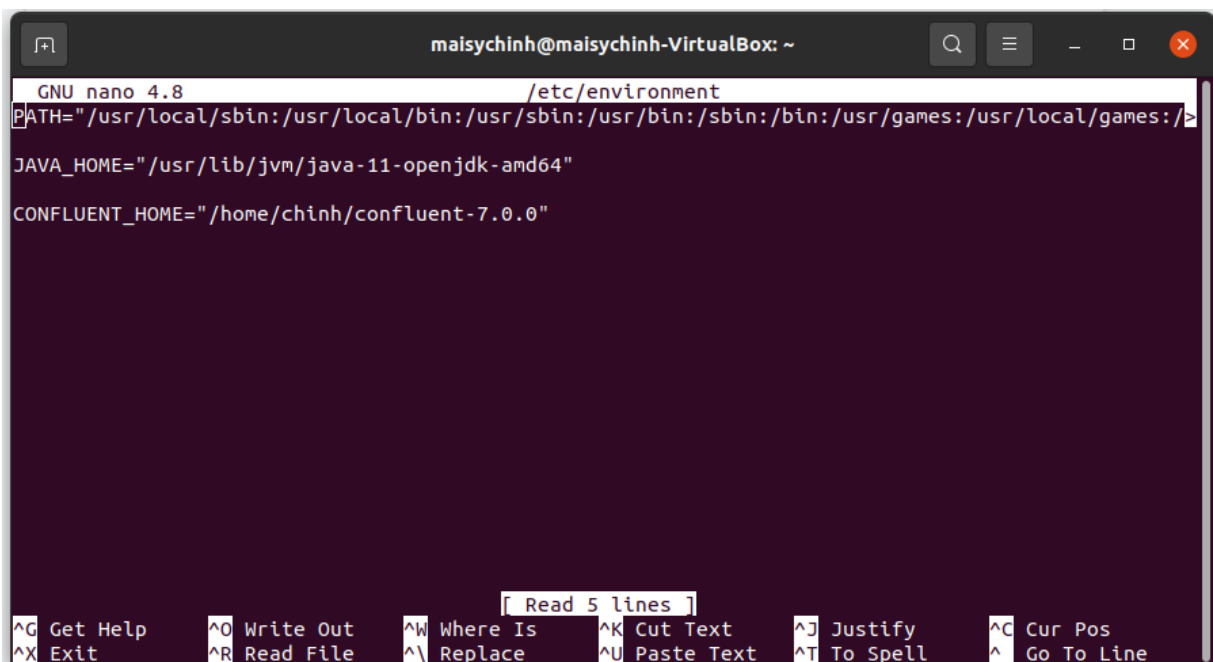
Note: Cài đặt curl trước khi tải **sudo apt install curl**. Nếu cài rồi mà chạy không được thì ta set biến môi trường bằng cách thêm **:/snap/bin** vào sau PATH

- Set biến môi trường:



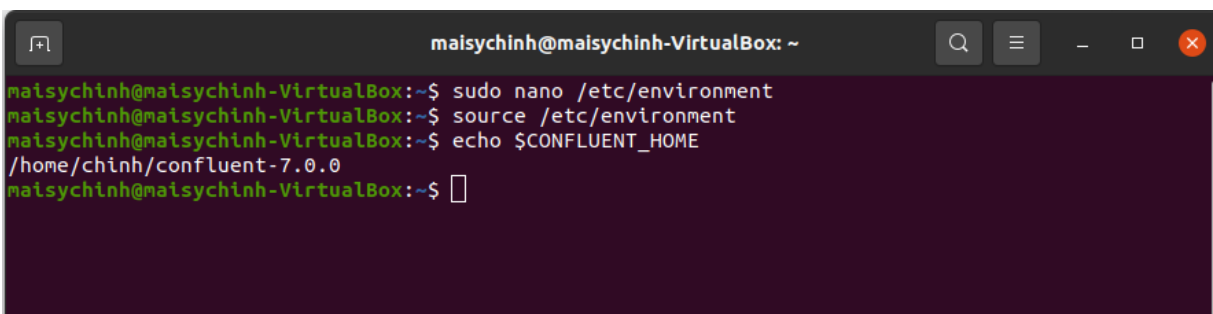
Hình 4.3: Set biến môi trường

- Thêm: `:$CONFLUENT_HOME/bin:$JAVA_HOME` vào cuối PATH
- Thêm dòng: `JAVA_HOME="/usr/lib/jvm/java-11-openjdk-amd64"`
- Thêm dòng: `CONFLUENT_HOME="/home/tinh/confluent-7.0.0"` (sửa tên user)



Hình 4.4: Thêm biến môi trường

- Control O và enter để save, Control X để thoát.
- Load lại tệp: `$ source /etc/environment`
- Vào terminal: `$ echo $CONFLUENT_HOME` và `$ echo $JAVA_HOME` để kiểm tra.



Hình 4.5: Kiểm tra đã thêm biến môi trường thành công

4.2.2 Zookeeper

- Zookeeper là gì: Zookeeper đóng vai trò là nơi lưu trữ dữ liệu phân tán dạng key-value. Nó được tối ưu hóa cho tác vụ đọc nhanh nhưng ghi chậm. Kafka sử dụng Zookeeper để thực hiện việc bầu chọn leader của Kafka broker và topic partition. Zookeeper cũng được thiết kế cho khả năng chịu lỗi cao, do đó Kafka phụ thuộc khá nhiều vào Zookeeper.
- Tải về và giải nén ta được thư mục confluent-7.0.0

```
maisychnh@maisychnh-VirtualBox:/home/chinh$ ls
confluent-7.0.0.tar.gz
maisychnh@maisychnh-VirtualBox:/home/chinh$ sudo tar xf confluent-7.0.0.tar.gz
maisychnh@maisychnh-VirtualBox:/home/chinh$ ls
confluent-7.0.0  confluent-7.0.0.tar.gz
maisychnh@maisychnh-VirtualBox:/home/chinh$
```

Hình 4.6: Giải nén thư mục confluent sau khi tải về

- Cấu hình file **zookeeper.properties** trong thư mục kafka
 - Điều hướng đến **/etc/kafka/zookeeper.properties**

```
maisychnh@maisychnh-VirtualBox:/home/chinh/confluent-7.0.0/etc$ cd kafka
maisychnh@maisychnh-VirtualBox:/home/chinh/confluent-7.0.0/etc/kafka$ sudo vim zookeeper.properties
```

Hình 4.7: Vào file zookeeper.properties

Ở file **zookeeper.properties**, ta thêm các biến như sau:

- tickTime=2000
- dataDir=/var/lib/zookeeper/
- clientPort=2181

```

n
S
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
#   http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
# the directory where the snapshot is stored.
Z tickTime=2000
dataDir=/var/lib/zookeeper/
# the port at which the clients will connect
clientPort=2181
# disable the per-ip limit on the number of connections since this is a non-production config
#maxClientCnxns=0
# Disable the adminserver by default to avoid port conflicts.
# Set the port to something non-conflicting if choosing to enable this
#admin.enableServer=false
# admin.serverPort=8080
-- INSERT --

```

Hình 4.8: Cấu hình zookeeper

4.2.3 Kafka

- Điều hướng đến `/etc/kafka/server.properties` để cấu hình lại Kafka

```

maisychinh@maisychinh-VirtualBox: /home/chinh/confluent-7.0.0/etc/kafka
maisychinh@maisychinh-VirtualBox:/home/chinh/confluent-7.0.0/etc/kafka$ sudo vim server.properties

```

Hình 4.9: Vào file server.properties

- Tái cấu hình broker bằng cách bỏ comment ở `broker.id=0`

```

# see kafka.server.KafkaConfig for additional details and defaults

##### Server Basics #####
# The id of the broker. This must be set to a unique integer for each broker.
broker.id=0

##### Socket Server Settings #####

```

Hình 4.10: Bỏ chú thích ở broker.id

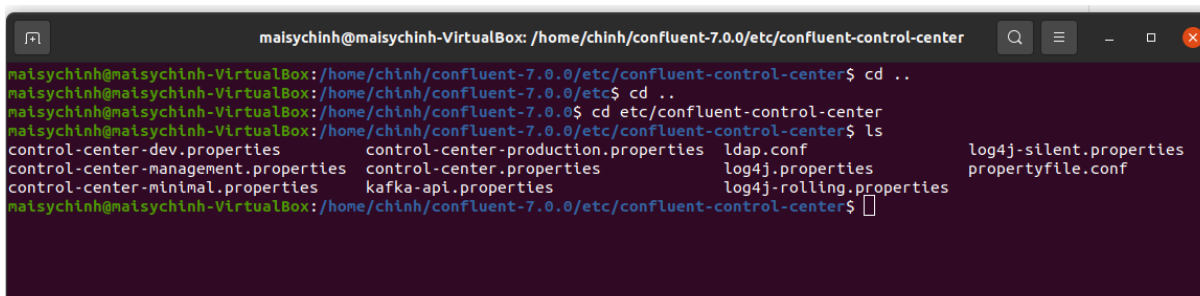
- Kết nối với nhóm Zookeeper bằng cách thiết lập `zookeeper.connect` ở các node về cùng một giá trị. Thay thế các miền cổng localhost về một port chính. Ở đây ta thực hiện thêm dòng `zookeeper.connect=zookeeper:2181`

```
##### ZOOKEEPER #####
# Zookeeper connection string (see zookeeper docs for details).
# This is a comma separated host:port pairs, each corresponding to a zk
# server. e.g. "127.0.0.1:3000,127.0.0.1:3001,127.0.0.1:3002".
# You can also append an optional chroot string to the urls to specify the
# root directory for all kafka znodes.
zookeeper.connect=localhost:2181
```

Hình 4.11: Cấu hình zookeeper.connect

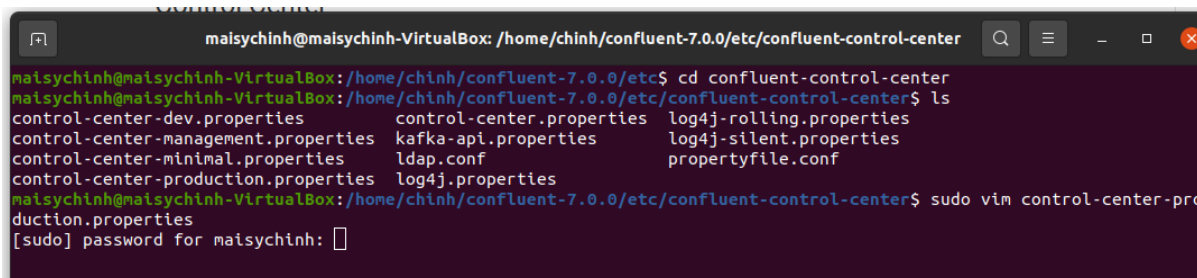
4.2.4 Control Center

- Copy nội dung của file **control-center-dev.properties** dán sang cả 2 file **control-center.properties** và file **control-center-production.properties**
- Chạy lệnh **sudo vim control-center-production.properties**:



```
maisychnh@maisychnh-VirtualBox: /home/chinh/confluent-7.0.0/etc/confluent-control-center$ cd ..
maisychnh@maisychnh-VirtualBox: /home/chinh/confluent-7.0.0/etc$ cd ..
maisychnh@maisychnh-VirtualBox: /home/chinh/confluent-7.0.0$ cd etc/confluent-control-center
maisychnh@maisychnh-VirtualBox: /home/chinh/confluent-7.0.0/etc/confluent-control-center$ ls
control-center-dev.properties      control-center-production.properties  ldap.conf          log4j-silent.properties
control-center-management.properties  control-center.properties             log4j.properties  propertyfile.conf
control-center-minimal.properties    kafka-api.properties                 log4j-rolling.properties
maisychnh@maisychnh-VirtualBox: /home/chinh/confluent-7.0.0/etc/confluent-control-center$
```

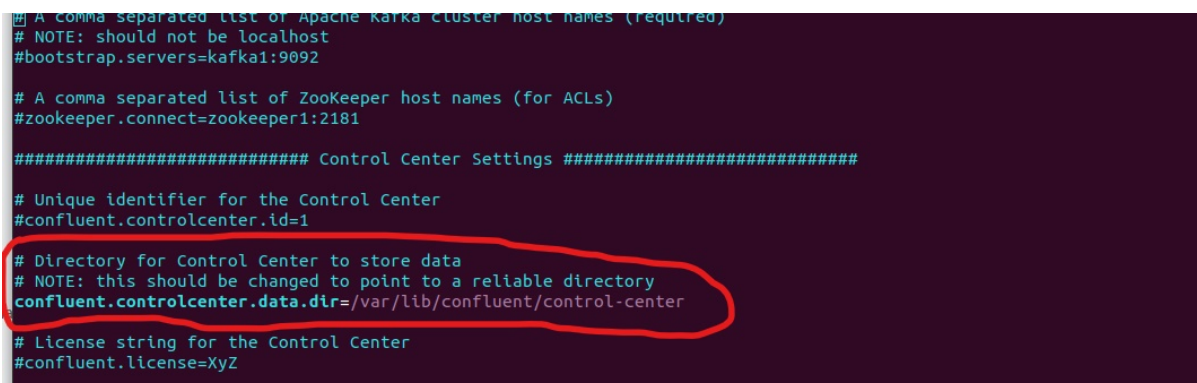
Hình 4.12: Control Center 1



```
maisychnh@maisychnh-VirtualBox: /home/chinh/confluent-7.0.0/etc/confluent-control-center$ cd confluent-control-center
maisychnh@maisychnh-VirtualBox: /home/chinh/confluent-7.0.0/etc/confluent-control-center$ ls
control-center-dev.properties      control-center.properties             log4j-rolling.properties
control-center-management.properties  kafka-api.properties                 log4j-silent.properties
control-center-minimal.properties    ldap.conf                             propertyfile.conf
control-center-production.properties  log4j.properties
maisychnh@maisychnh-VirtualBox: /home/chinh/confluent-7.0.0/etc/confluent-control-center$ sudo vim control-center-pro
duction.properties
[sudo] password for maisychnh:
```

Hình 4.13: Control Center 2

- Tìm đến dòng **confluent.controlcenter.data.dir** và thay đổi đường dẫn bằng **/var/lib/confluent/control-center**:



```
# A comma separated list of Apache Kafka cluster host names (required)
# NOTE: should not be localhost
#bootstrap.servers=kafka1:9092

# A comma separated list of ZooKeeper host names (for ACLs)
#zookeeper.connect=zookeeper1:2181

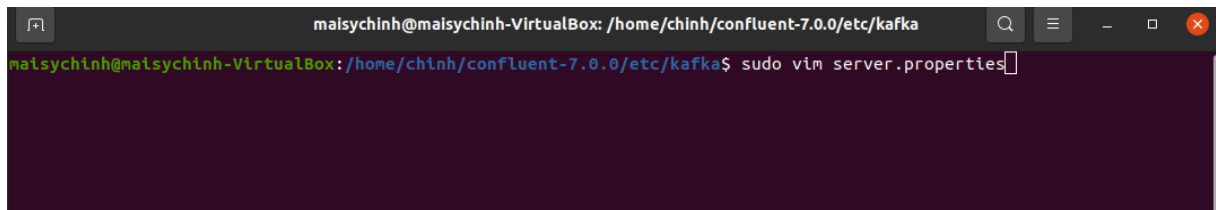
##### Control Center Settings #####
# Unique identifier for the Control Center
#confluent.controlcenter.id=1

# Directory for Control Center to store data
# NOTE: this should be changed to point to a reliable directory
confluent.controlcenter.data.dir=/var/lib/confluent/control-center

# License string for the Control Center
#confluent.license=XyZ
```

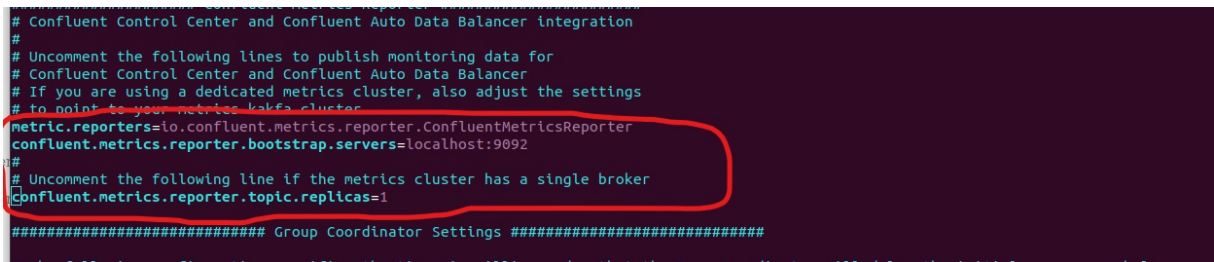
Hình 4.14: Thay đổi đường dẫn Control Center

- Cấu hình file **server.properties**



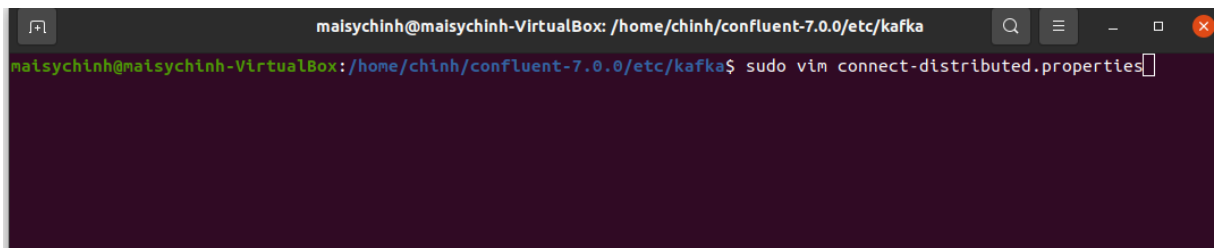
Hình 4.15: Vào file server.properties

- Cấp quyền cho phép Confluent Metrics Reporter bằng cách thiết lập như hình bên dưới.



Hình 4.16: Cấu hình Kafka server

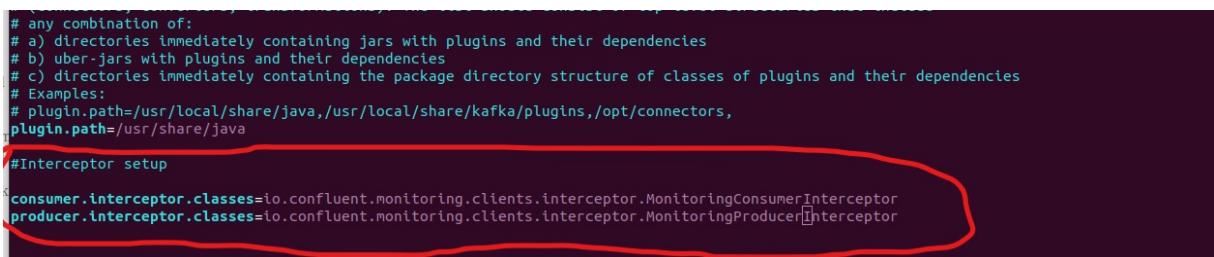
- Cấu hình file **connect-distributed.properties**



Hình 4.17: Vào file connect-distributed.properties

- Thêm đoạn code sau vào cuối file

consumer.interceptor.classes=io.confluent.monitoring.clients.interceptor.MonitoringConsumerInterceptor
producer.interceptor.classes=io.confluent.monitoring.clients.interceptor.MonitoringProducerInterceptor



Hình 4.18: Thêm vào các dòng như trên

4.2.5 Khởi động và dừng Confluent Platform

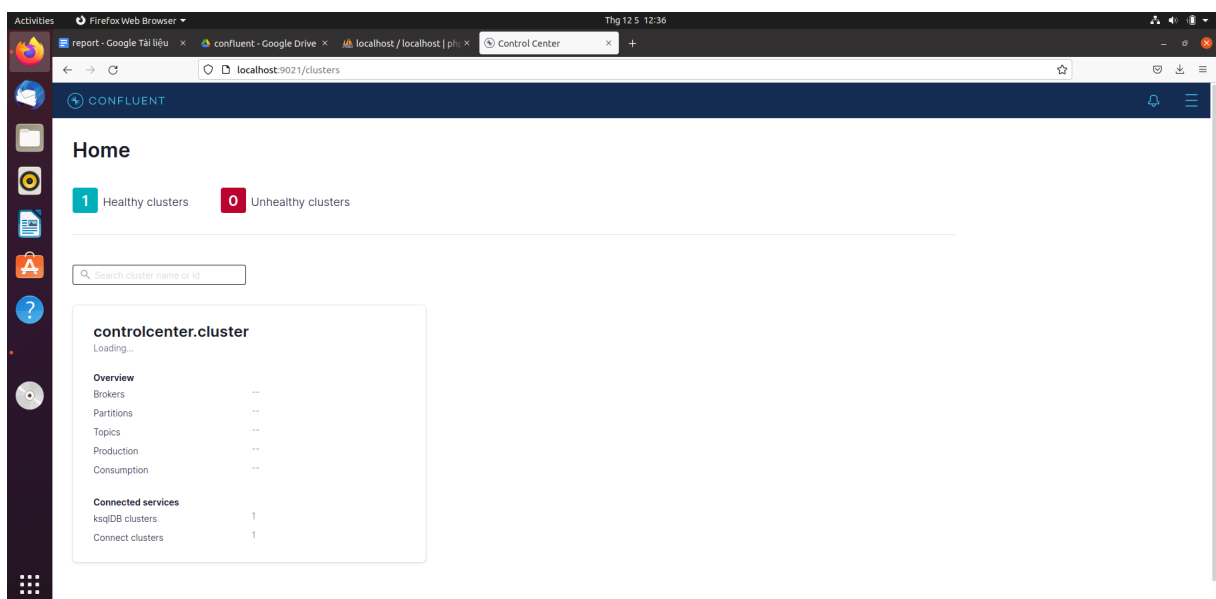
- Để chạy, ta dùng lệnh: **\$ confluent local services start**

```
maisy chinh@maisy chinh-VirtualBox:~$ sudo nano /etc/environment
maisy chinh@maisy chinh-VirtualBox:~$ source /etc/environment
maisy chinh@maisy chinh-VirtualBox:~$ echo $CONFLUENT_HOME
/home/chinh/confluent-7.0.0
maisy chinh@maisy chinh-VirtualBox:~$ confluent local services start
The local commands are intended for a single-node development environment only,
NOT for production usage. https://docs.confluent.io/current/cli/index.html

Using CONFLUENT_CURRENT: /tmp/confluent.579649
Starting ZooKeeper
ZooKeeper is [UP]
Starting Kafka
Kafka is [UP]
Starting Schema Registry
Schema Registry is [UP]
Starting Kafka REST
Kafka REST is [UP]
Starting Connect
Connect is [UP]
Starting ksqlDB Server
ksqlDB Server is [UP]
Starting Control Center
Control Center is [UP]
maisy chinh@maisy chinh-VirtualBox:~$
```

Hình 4.19: Khởi chạy Confluent Platform

- Truy cập link: <http://localhost:9021>



Hình 4.20: Giao diện

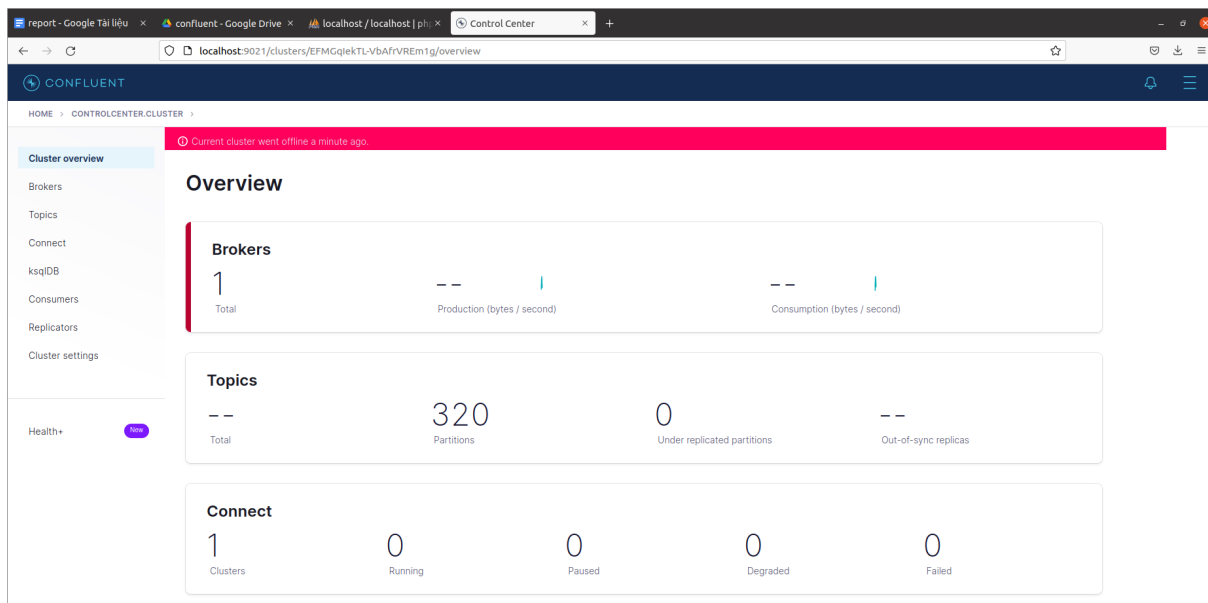
- Để dừng, ta dùng lệnh: `$ confluent local services stop`

4.3 Quick start

4.3.1 Tạo Kafka topic thử nghiệm

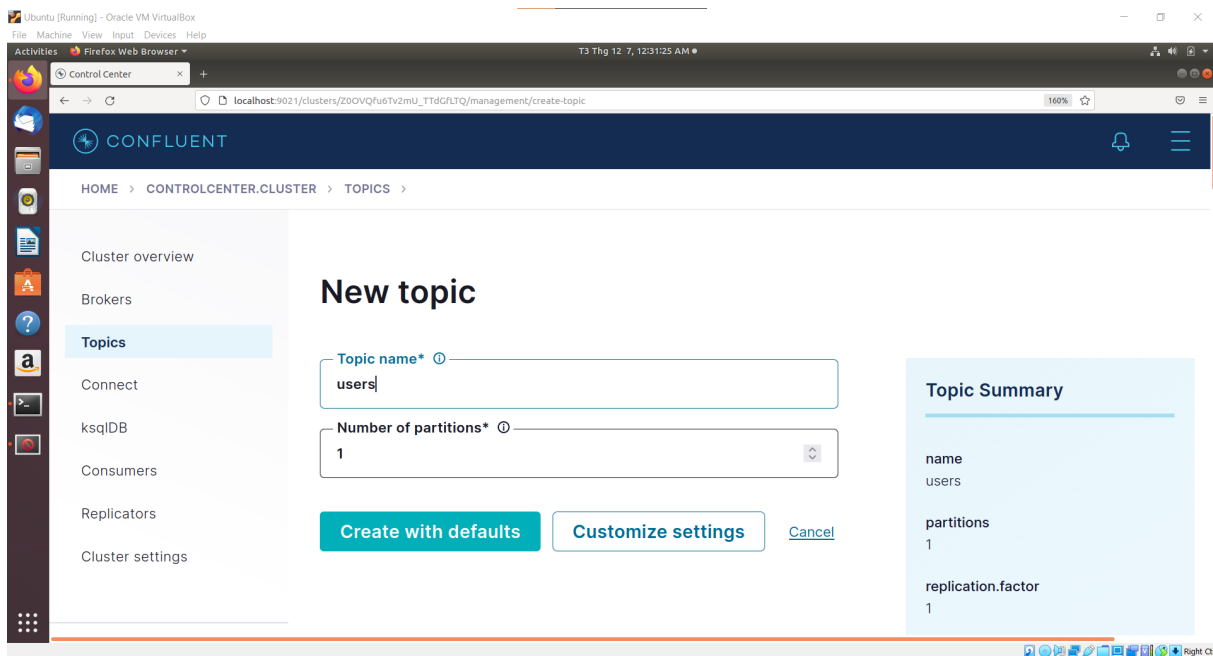
- Tạo users topic
 - Điều hướng đến Trung tâm điều khiển tại <http://localhost:9021>. Có thể mất một hoặc hai phút để khởi động và tải.

- Click vào **controlcenter.cluster**.



Hình 4.21: Overview

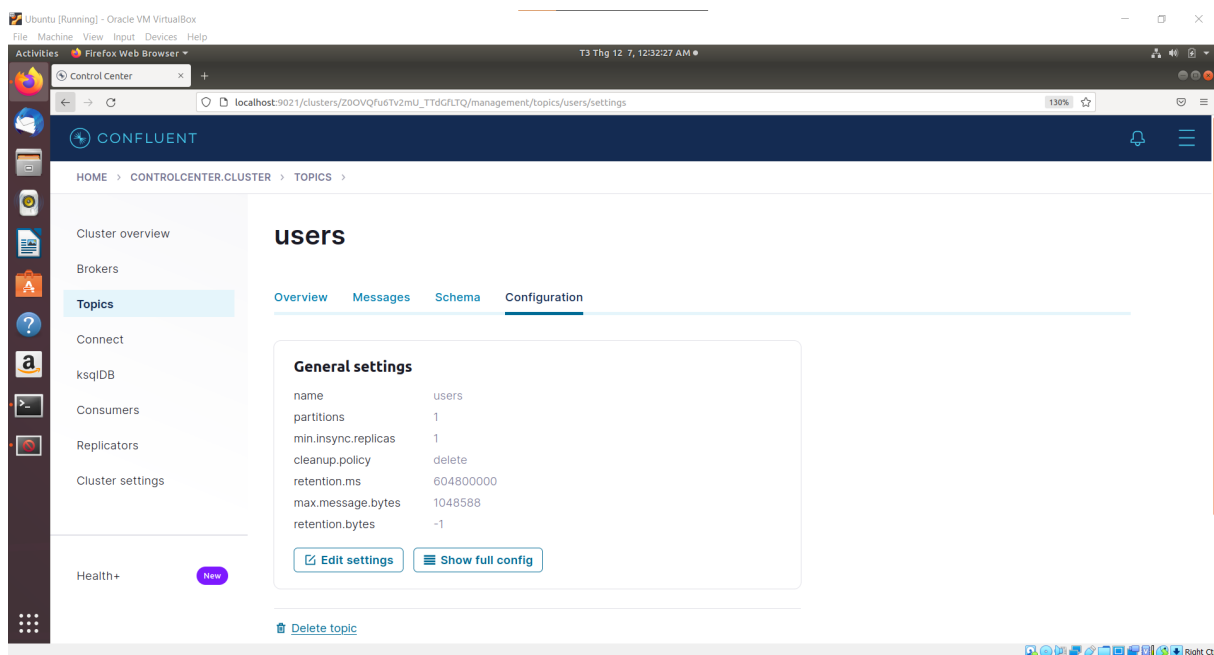
- Trên thanh menu, click vào **Topics** để mở danh sách các topic. Click vào **Add a topic** để tạo users topic.



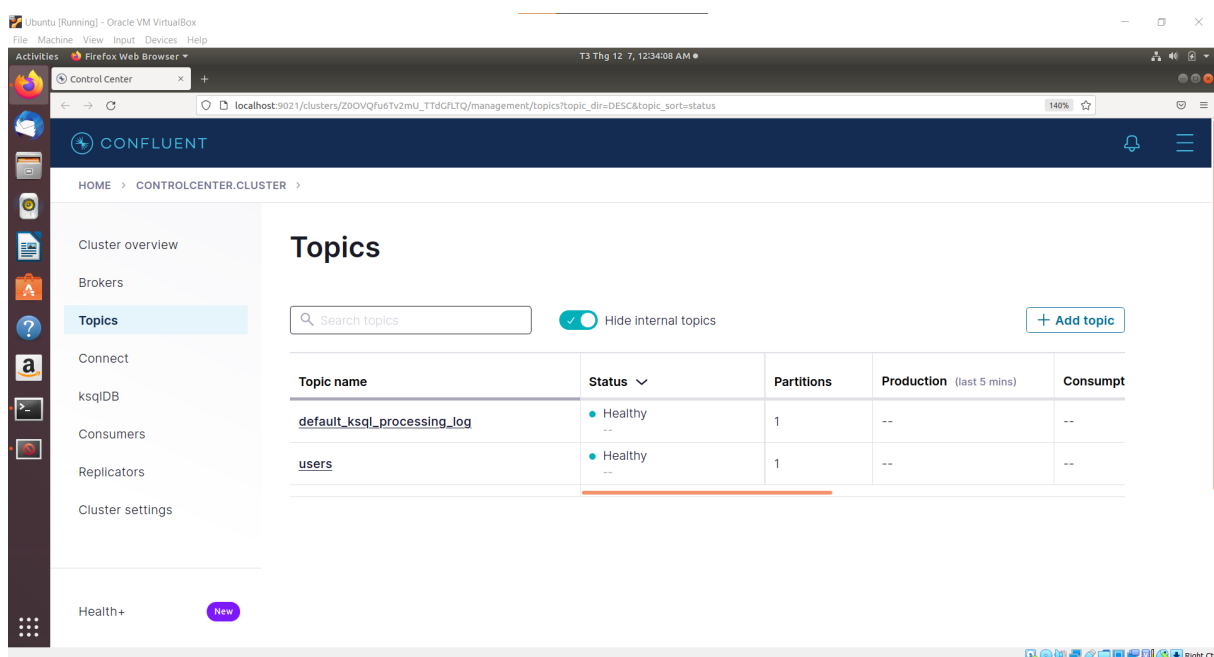
Hình 4.22: Tạo topic

Sau khi đặt tên cho topic, Click **Create with defaults** để tạo topic.

- Xem lại topic vừa tạo



Hình 4.23: users topic



Hình 4.24: Xem lại các topic

4.3.2 Sinh dữ liệu giả

- Cài đặt nguồn cấp dữ liệu giả **Datagen Source Connector**
 - Chạy lệnh `confluent-hub install confluentinc/kafka-connect-datagen:0.5.2` để cài đặt
- Tạo dữ liệu giả
 - Vào tab **Connect** chọn **connect-default**

Cluster overview	Connect clusters					
Brokers	<input type="text" value="Search"/>					
Topics						
Connect						
ksqlDB						
Consumers						
Replicators						
Cluster settings						

Cluster name	Total connectors	Running connectors	Degraded connectors	Failed connectors	Paused connectors
connect-default	1	1	0	0	0

Hình 4.25: Chọn connect-default

- Chọn Add connector

Cluster overview	Connectors					
Brokers						
Topics						
Connect						
ksqlDB						
Consumers						
Replicators						
Cluster settings						

Connector name	Status	Category	Plugin name	Topics	Number of tasks
datagen-pageviews	Running	Source	DatagenConnector	--	1

Hình 4.26: Chọn Add connector

- Chọn Datagen Connector

Cluster overview	Browse					
Brokers						
Topics						
Connect						
ksqlDB						
Consumers						
Replicators						
Cluster settings						

Connector name	Status	Category	Plugin name	Topics	Number of tasks
datagen-pageviews	Running	Source	DatagenConnector	--	1

Hình 4.27: Chọn Datagen Connector

- Chỉnh sửa lần lượt:

+ Name: datagen-users

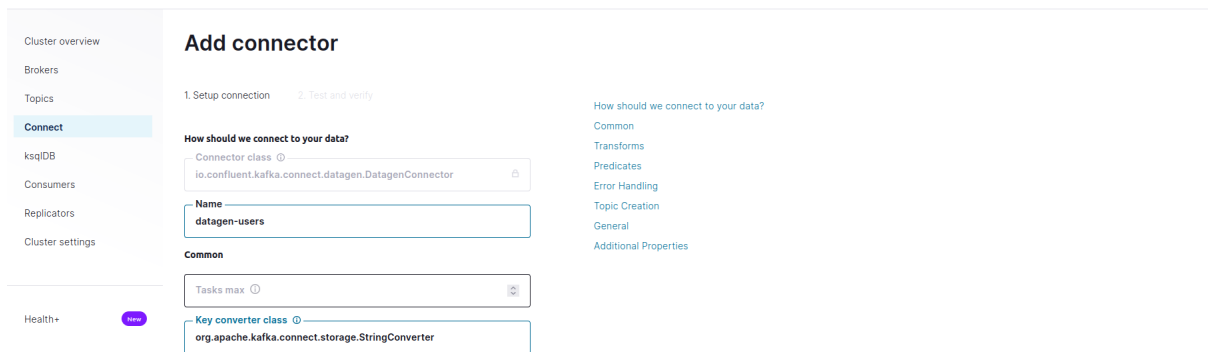
+ Key converter class:

org.apache.kafka.connect.storage.StringConverter

+ kafka.topic: users

+ max.interval:1000

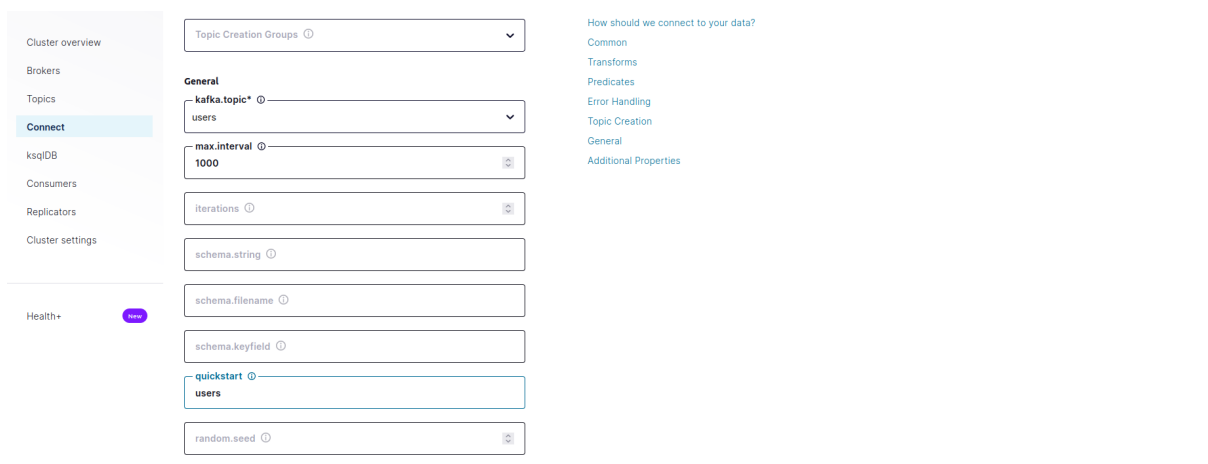
+ quickstart:users



The screenshot shows the 'Add connector' dialog with the following details:

- Connector class:** `io.confluent.kafka.connect.datagen.DatagenConnector`
- Name:** `datagen-users`
- Key converter class:** `org.apache.kafka.connect.storage.StringConverter`

Hình 4.28: Chỉnh sửa Name và Key coverter class

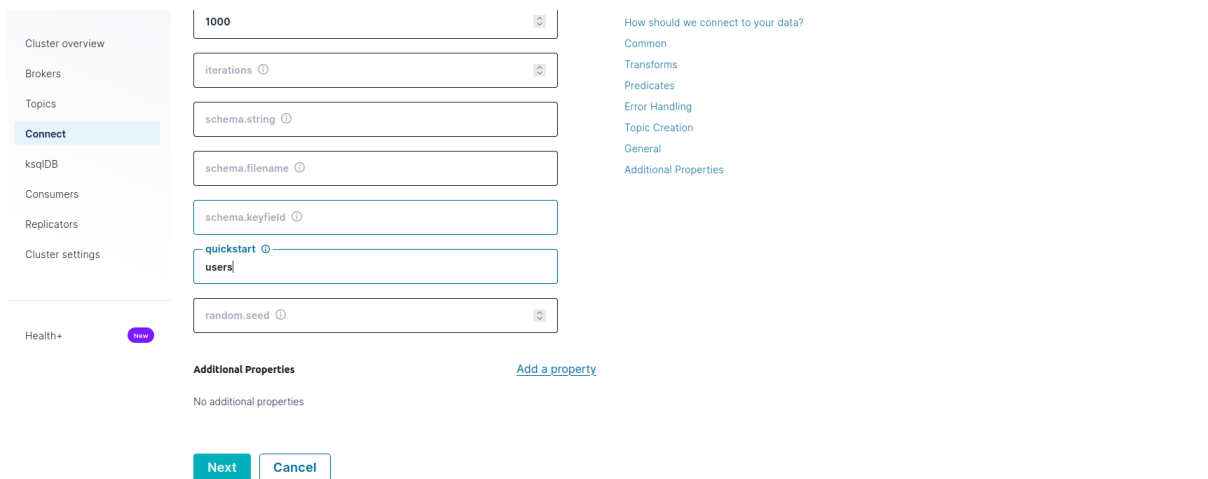


The screenshot shows the 'General' tab of the 'Add connector' dialog with the following details:

- Topic Creation Groups:** `users`
- kafka.topic*:** `users`
- max.interval:** `1000`
- quickstart:** `users`
- random.seed:** (empty)

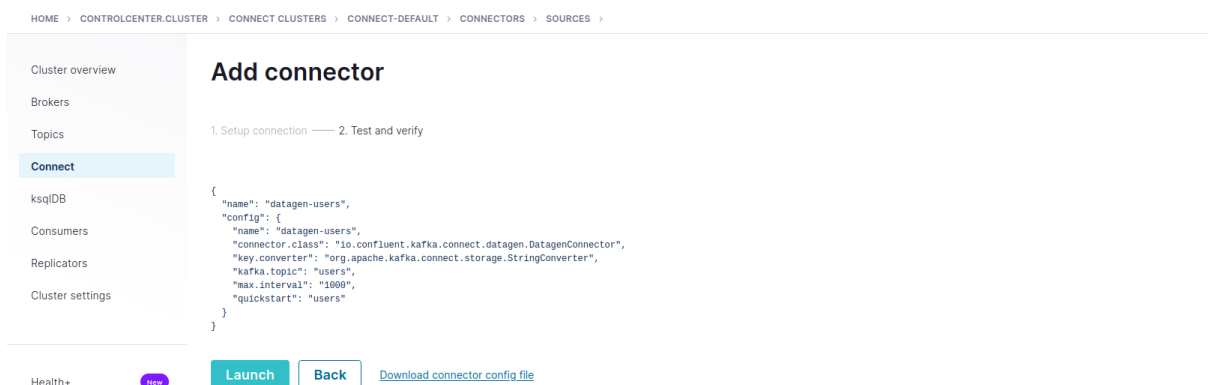
Hình 4.29: Chỉnh sửa kafka.topic, max.interval và quickstart

- Sau khi chỉnh sửa xong, chọn **Next** -> **Launch**



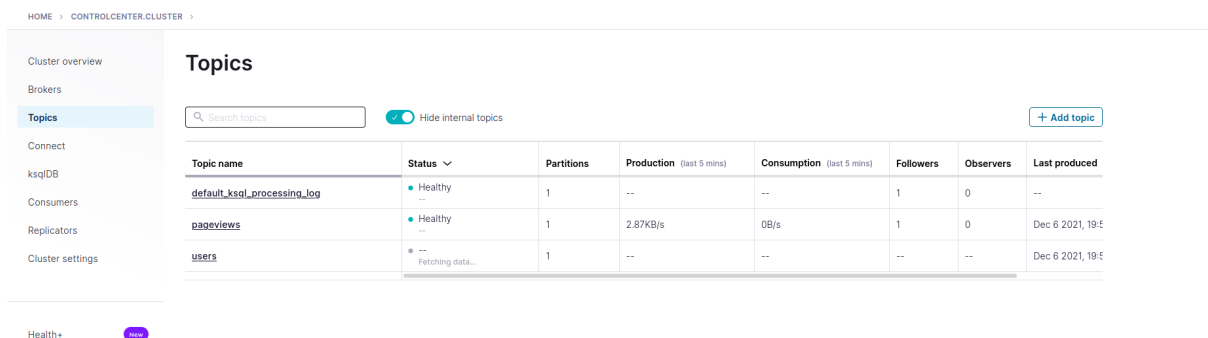
The screenshot shows the 'Additional Properties' tab of the 'Add connector' dialog. The 'Next' button is highlighted, indicating the user should proceed to the next step.

Hình 4.30: Chọn Next



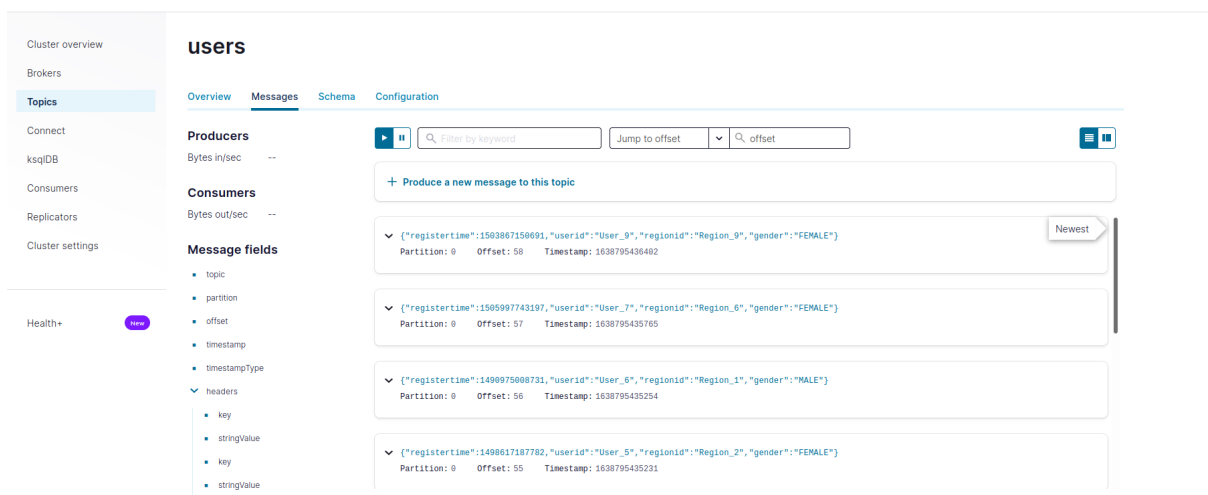
Hình 4.31: Chọn Launch

- Kiểm tra dữ liệu tạo ra
 - Trở về tab **Topics** sau đó chọn **users**



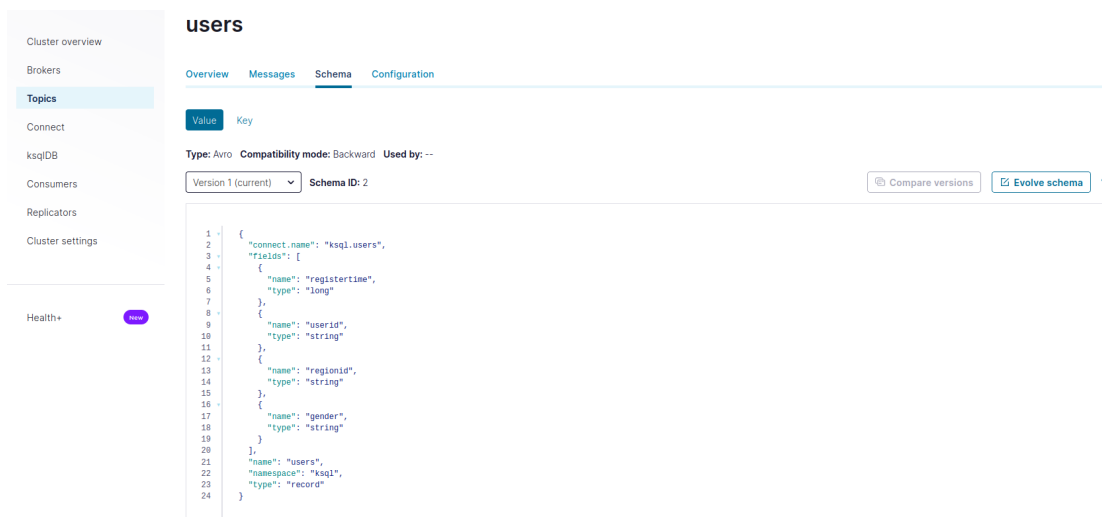
Hình 4.32: Xem lại các Topic

- Dữ liệu giả đã được sinh ra



Hình 4.33: Dữ liệu giả đã được sinh ra

- Dữ liệu ở định dạng Avro, định nghĩa cấu trúc của users messages



Hình 4.34: Dữ liệu ở định dạng Avro

5 JDBC Connector cho Confluent Platform

5.1 Giới thiệu về JDBC Connector

JDBC Connector được sử dụng để trích xuất dữ liệu từ một database bất kỳ cùng với JDBC Driver vào Apache Kafka Topic. Chúng ta có thể sử dụng JDBC Sink để truy xuất dữ liệu từ Kafka Topic tới bất kỳ database nào cùng với JDBC Driver. JDBC Connector hỗ trợ nhiều loại database mà không cần yêu cầu custom code.

5.2 Cài đặt JDBC Connector

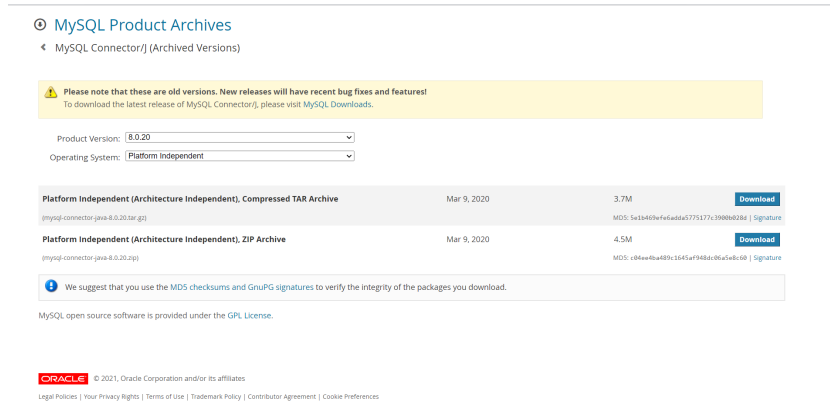
Chúng ta cài đặt **JDBC Connector** bằng cách gõ câu lệnh sau vào **cmd**:

```
confluent-hub install confluentinc/kafka-connect-jdbc:latest
```

Hình 5.1: Cài đặt JDBC Connector

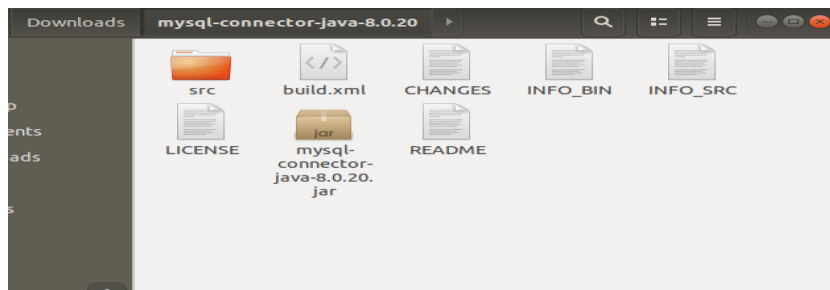
5.3 Cài đặt JDBC Driver

Đầu tiên, chúng ta tới trang web download JDBC Driver [tại đây](#).
Tiếp theo, lựa chọn **Product Version: 8.0.20** và **Operating System: Platform Independent**.



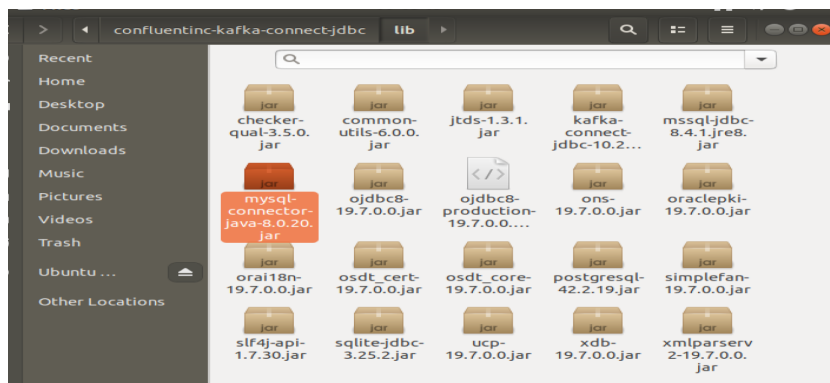
Hình 5.2: Downloads JDBC Driver

Sau đó, chúng ta lựa chọn file *.tar.gz* và tiến hành download.
Sau khi download xong thì chúng ta sẽ tiến hành giải nén.



Hình 5.3: Giải nén file download

Cuối cùng, chúng ta *copy* file **mysql-connector-java-8.0.20.jar** vào thư mục đã cài confluent ở trên `/home/tuananh/confluent-7.0.0/share/confluent-hub-components/confluentinc-kafka-connect-jdbc/lib/`.



Hình 5.4: Copy file

5.4 Tạo Kafka topic

Trước hết, chúng ta phải cài biến môi trường. Đầu tiên, chúng ta sử dụng câu lệnh **ls -al** tại thư mục `/home/tuananh/`.

```
tuananh@tuananh-virtual-machine:~$ ls -al
total 1701012
drwxr-xr-x 24 tuananh tuananh      4096 Thg 1  5 01:28 .
drwxr-xr-x  3 root    root        4096 Thg 1 25 00:35 ..
-rw-r----- 1 tuananh tuananh     4922 Thg 1  4 23:56 .bash_history
-rw-r--r--  1 tuananh tuananh      220 Thg 1 25 00:35 .bash_logout
-rw-r--r--  1 tuananh tuananh     3977 Thg 1  4 15:40 .bashrc
drwx----- 17 tuananh tuananh     4096 Thg 1  4 15:34 .cache
drwx----- 13 tuananh tuananh     4096 Thg 1  4 01:41 .config
drwxrwxr-x  2 tuananh tuananh     4096 Thg 1  4 16:48 .confluent
drwxr-xr-x  8 tuananh tuananh     4096 Thg 1 21 10:40 confluent-7.0.0
-rw-rw-r--  1 tuananh tuananh 1741664450 Thg 1  4 15:03 confluent-7.0.0.tar.gz
drwxrwxr-x  5 tuananh tuananh     4096 Thg 1  4 15:14 confluent-hub-client-la
test
drwx----- 3 root    root        4096 Thg 1  4 01:40 .dbus
drwxr-xr-x  2 tuananh tuananh     4096 Thg 1 25 00:38 Desktop
drwxr-xr-x  2 tuananh tuananh     4096 Thg 1 25 00:38 Documents
drwxr-xr-x  3 tuananh tuananh     4096 Thg 1  4 16:12 Downloads
-rw-r--r--  1 tuananh tuananh     8980 Thg 1 25 00:35 examples.desktop
```

Hình 5.5: Cài biến môi trường 1

Ta sử dụng câu lệnh **sudo vim .bashrc** để truy cập vào file **.bashrc**. Sau đó ta tạo biến môi trường theo hình dưới.

```

JAVA_HOME="/usr/lib/jvm/java-11-openjdk-amd64"
CONFLUENT_HOME="/home/tuananh/confluent-7.0.0"
export JAVA_HOME
export CONFLUENT_HOME
export PATH=$PATH:$JAVA_HOME/bin
export PATH=$PATH:$CONFLUENT_HOME/bin
".bashrc" 125L, 3977C                                     125,0-1      Bot

```

Hình 5.6: Cài biến môi trường 2

Kiểm tra xem cài đã thành công hay chưa.

```
tuananh@tuananh-virtual-machine:~$ confluent-hub
usage: confluent-hub <command> [ <args> ]

Commands are:
  help      Display help information
  install   Install a component from either Confluent Hub or from a local file
  e

See 'confluent-hub help <command>' for more information on a specific command.
```

Hình 5.7: Cài biến môi trường 3

Tiếp đến, chúng ta sử dụng câu lệnh **confluent local services start** để khởi động Confluent Platform.

```
confluent local services start
```

Your output should resemble:

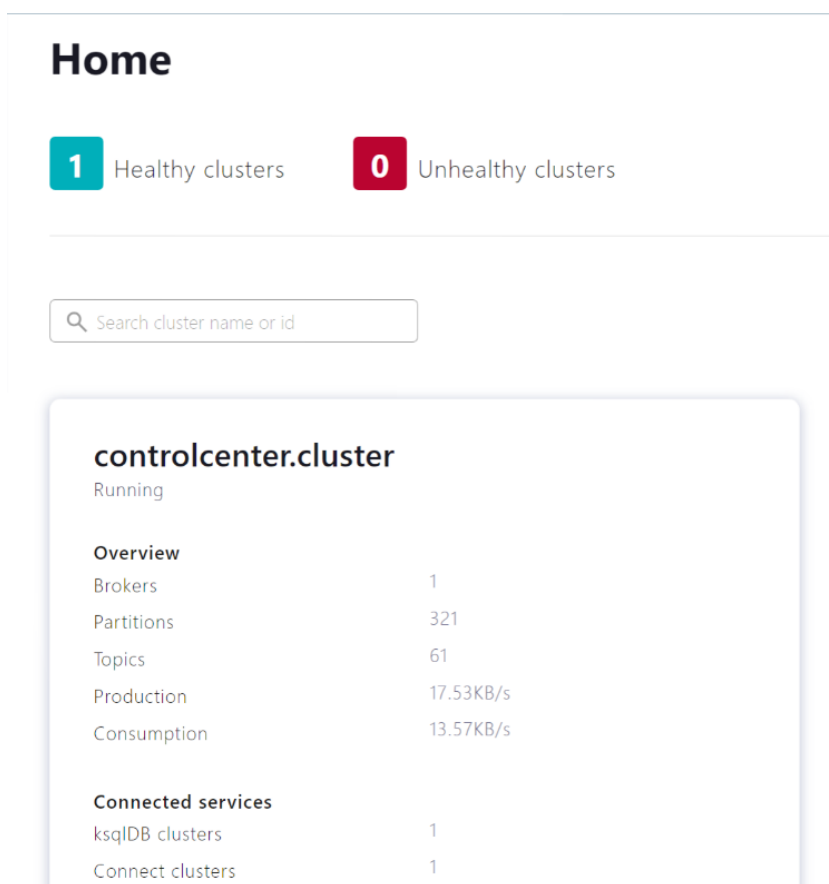
```

Starting Zookeeper
Zookeeper is [UP]
Starting Kafka
Kafka is [UP]
Starting Schema Registry
Schema Registry is [UP]
Starting Kafka REST
Kafka REST is [UP]
Starting Connect
Connect is [UP]
Starting KSQL Server
KSQL Server is [UP]
Starting Control Center
Control Center is [UP]

```

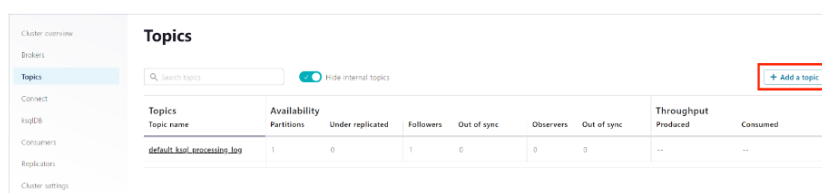
Hình 5.8: Khởi động Confluent Platform

Ta truy cập vào đường link localhost:9021/clusters.



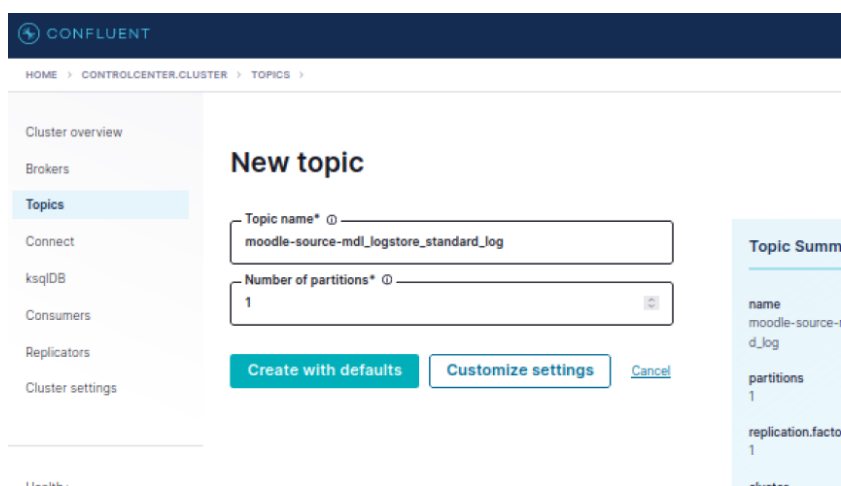
Hình 5.9: Tạo Kafka topic 1

Tiếp theo, 'click' vào **controlcenter.cluster**. Sau đó, chọn **Topic** trên *navigation menu*. Rồi 'click' vào **Add a topic** để tạo **moodle-source-mdl_logstore_standard_log** topic.



Hình 5.10: Tạo Kafka topic 2

Trong phần **Topic name**, chúng ta điền **moodle-source-mdl_logstore_standard_log**.

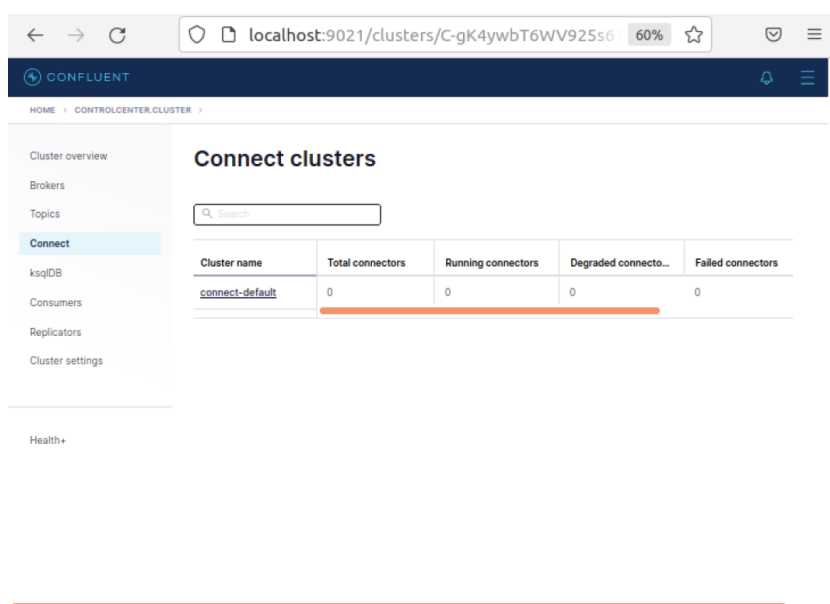


Hình 5.11: Tạo Kafka topic 3

'Click' vào **Create with defaults** để tạo Kafka Topic.

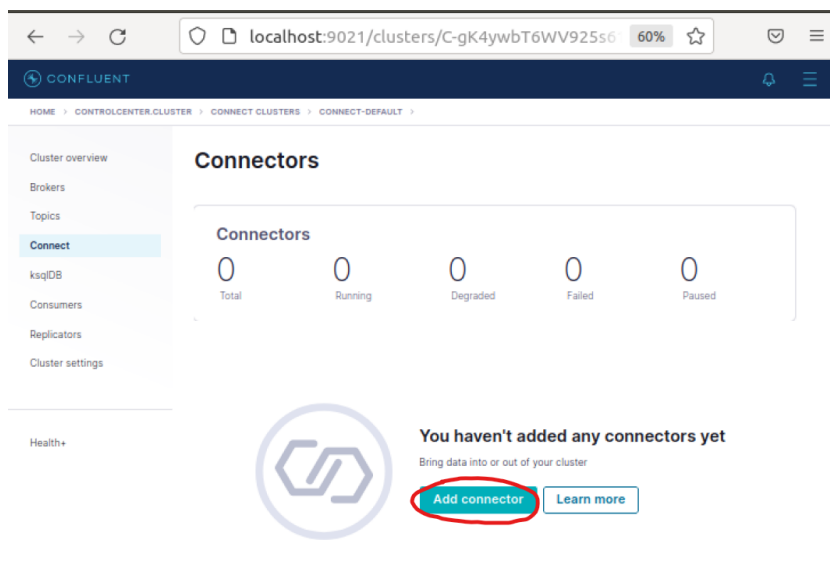
5.5 Thêm JDBC Connector vào Confluent Platform

Chọn **Connect** ở trên *navigation menu* sẽ được hình như dưới.



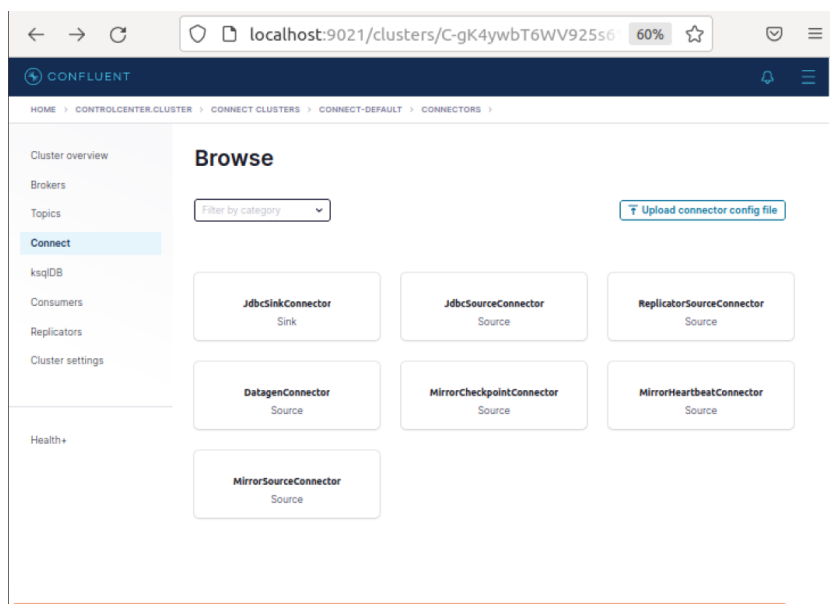
Hình 5.12: Thêm JDBC Connector 1

Tiếp theo chọn **connect-default** clusters trong Connect clusters list. Rồi chọn **Add connector** để tạo connector.



Hình 5.13: Thêm JDBC Connector 2

Chọn mục **JdbcSourceConnector**.



Hình 5.14: Thêm JDBC Connector 3

Khi đó sẽ hiện ra bảng setting. Chúng ta sẽ cài đặt như hình dưới đây.

How should we connect to your data?

Connector class ⓘ
io.confluent.connect.jdbc.JdbcSource ⓘ

Name
moodle-source

Common

Tasks max ⓘ
1

Key converter class ⓘ

Value converter class ⓘ

Header converter class ⓘ

Reload Action ⓘ

Transforms

Transforms ⓘ

Predicates

Predicates ⓘ

Error Handling

Retry Timeout for Errors ⓘ

Maximum Delay Between Retries for Errors ⓘ

Error Tolerance ⓘ

Log Errors ⓘ x

Log Error Details ⓘ x

Hình 5.15: Các bước cài đặt JDBC Connector 1

Topic Creation Groups ⓘ

Database

JDBC URL* ⓘ
jdbc:mysql://localhost:3306/moodle

JDBC User ⓘ
root

JDBC Password ⓘ
.....

JDBC connection attempts ⓘ

JDBC connection backoff in milliseconds ⓘ

Table Whitelist ⓘ
mdl_logstore_standard_log x

Database Dialect ⓘ
MySqlDatabaseDialect

Mode

Table Loading Mode ⓘ
incrementing

Incrementing Column Name ⓘ
id

Validate Non Null ⓘ x

Query ⓘ

Quote Identifiers ⓘ

Query suffix ⓘ

Connector

Table Types ⓘ

Roll Interval (sec) ⓘ

Hình 5.16: Các bước cài đặt JDBC Connector 2

Connector

Table Types ⓘ

Poll Interval (ms) ⓘ

Max Rows Per Batch ⓘ

Metadata Change Monitoring Interval (ms) ⓘ

Topic Prefix ⓘ
moodle-source-

Delay Interval (ms) ⓘ

DB time zone ⓘ

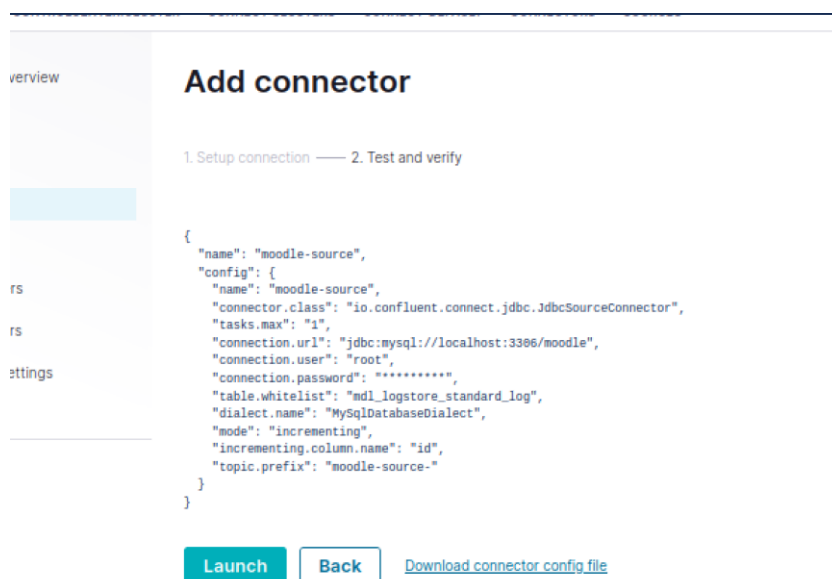
Additional Properties [Add a property](#)

No additional properties

Next **Cancel**

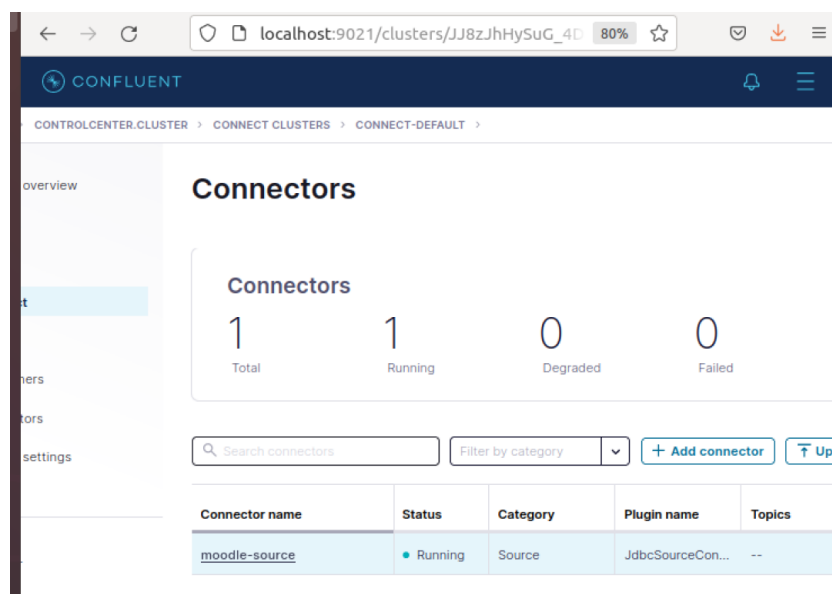
Hình 5.17: Các bước cài đặt JDBC Connector 3

Chọn *Next* để chuyển sang bước tiếp theo.



Hình 5.18: Cấu hình JDBC Connector

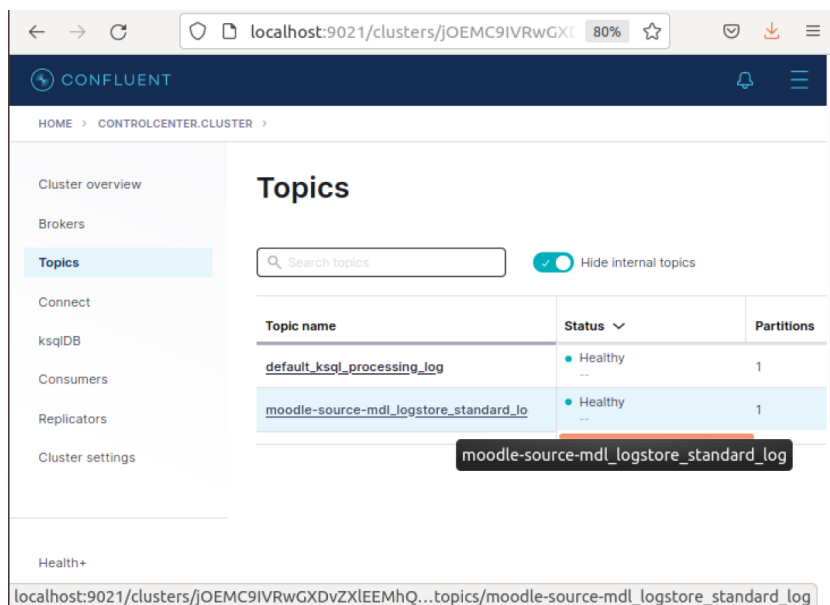
Chọn **Launch** sau khi đã kiểm tra cấu hình của JDBC Connector.



Hình 5.19: Cấu hình JDBC Connector

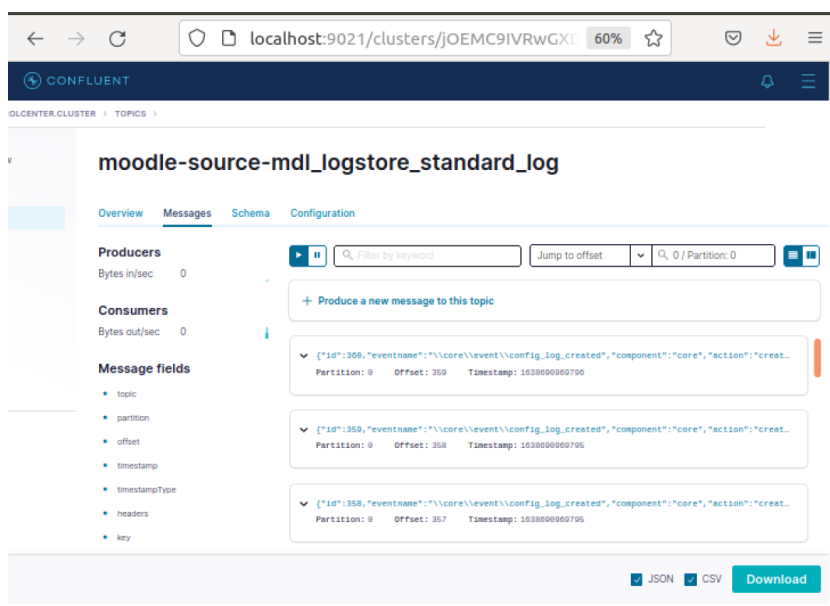
Nếu được kết quả là **Status: Running** như hình trên thì chúng ta đã cài đặt thành công. Bước cuối cùng, chúng ta sẽ kiểm tra lại kết nối.

Chọn **Topic** ở trên *navigation menu*. Sau đó chọn **moodle-source-mdl_logstore_standard_log**.



Hình 5.20: Kiểm tra kết nối JDBC Connector 1

Chọn **Message** để hiển thị thông tin chi tiết.

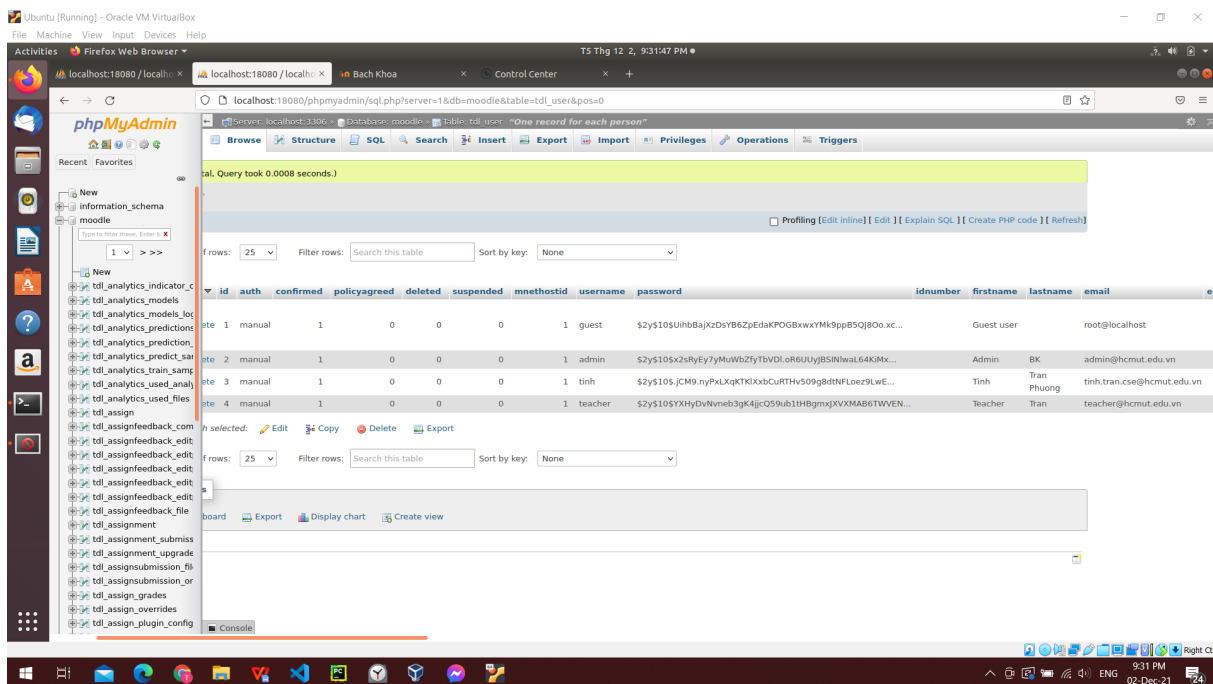


Hình 5.21: Kiểm tra kết nối JDBC Connector 2

6 Demo

Ta dùng tài khoản có username tỉnh với id=3 và role student để đăng nhập vào moodle. Trong moodle có course Operating system với courseid=2.

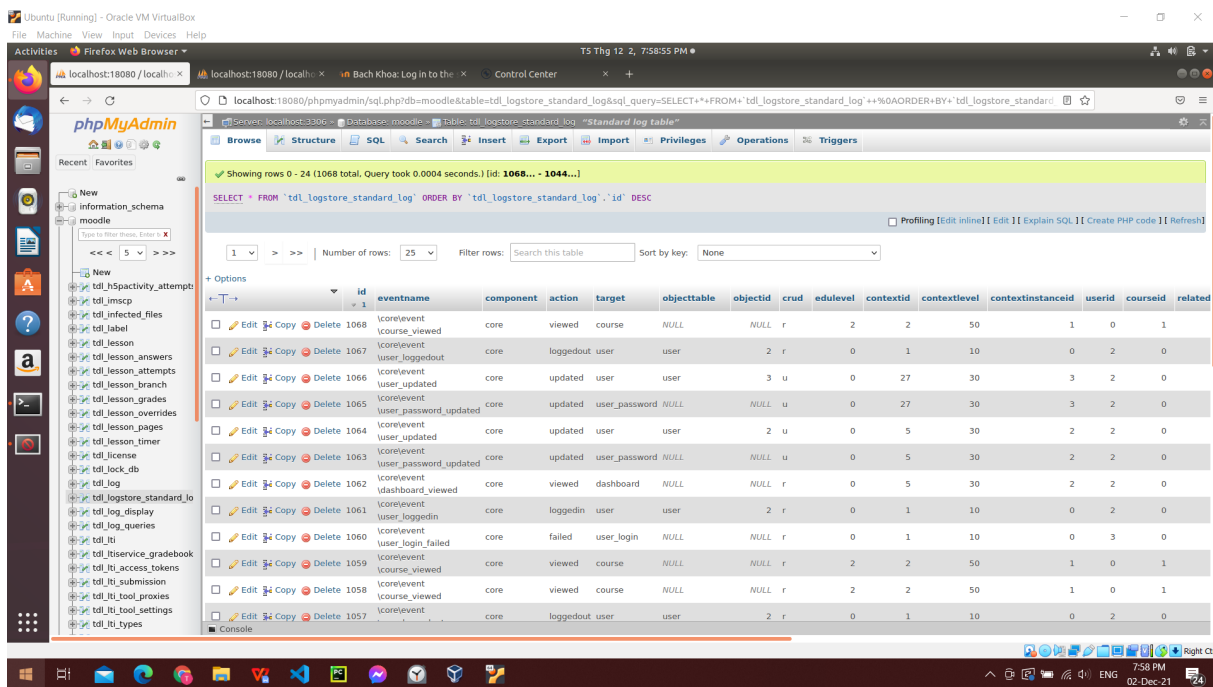
Ta lần lượt thử một số tính năng trong Moodle, dữ liệu về các hoạt động này sẽ được lưu trong bảng tdl_logstore_standard_log (hoặc mdl_logstore_standard_log, tùy vào Tables prefix lúc cài đặt Moodle) của cơ sở dữ liệu MySQL, đồng thời dữ liệu cũng được realtime streaming sang Confluent.



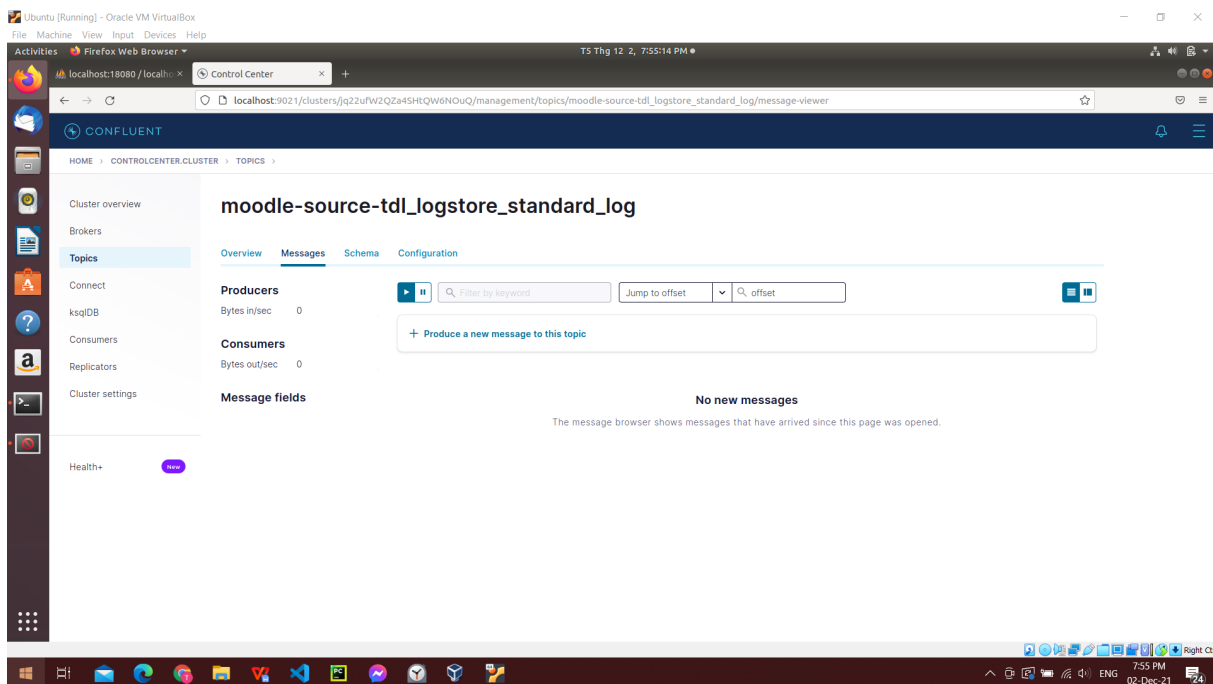
Hình 6.1: Bảng tdl_user

6.1 Login

Trước khi login:



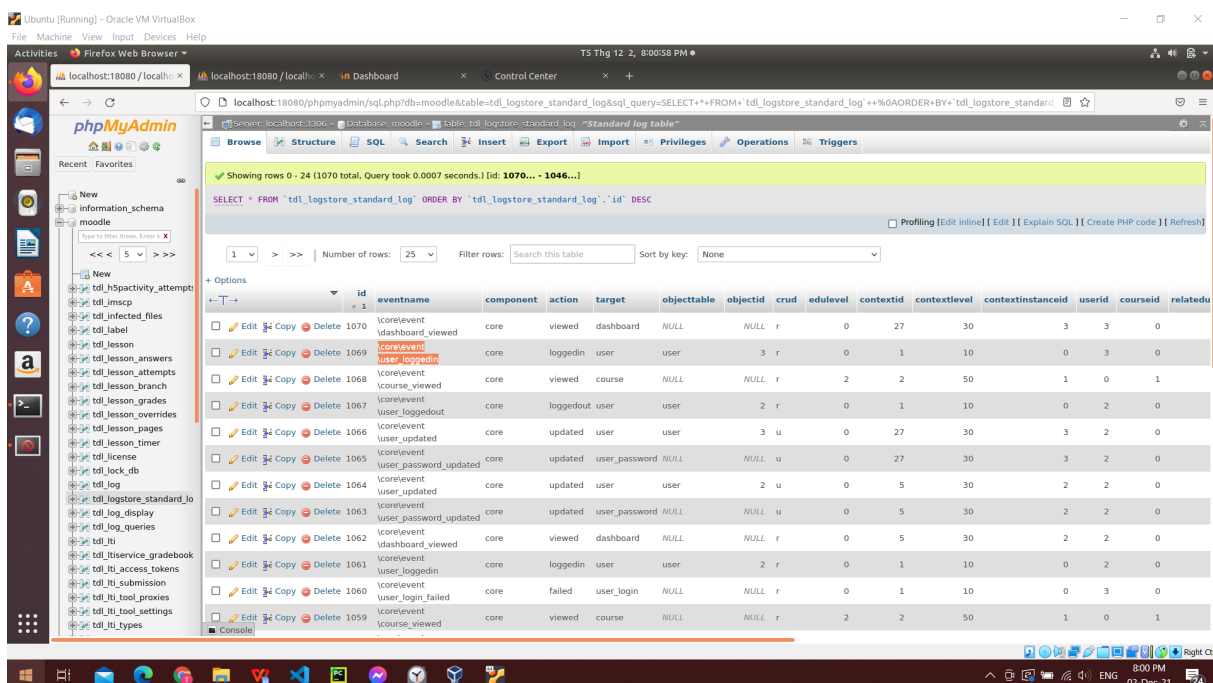
Hình 6.2: Bảng tdl_logstore_standard_log trước khi login



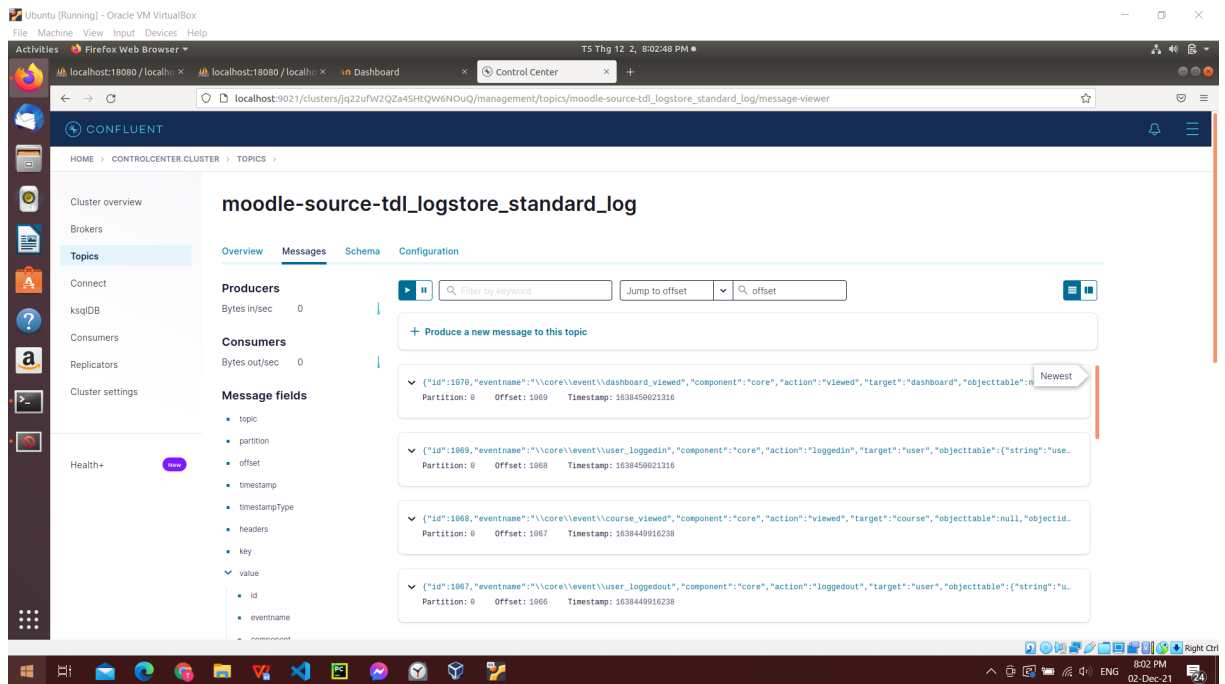
Hình 6.3: Confluent trước khi login

Sau khi login:

- Bảng tdl_logstore_standard_log có thêm dòng id=1069, action=loggedin, userid=3.
- Confluent đã nhận được message có id=1069.



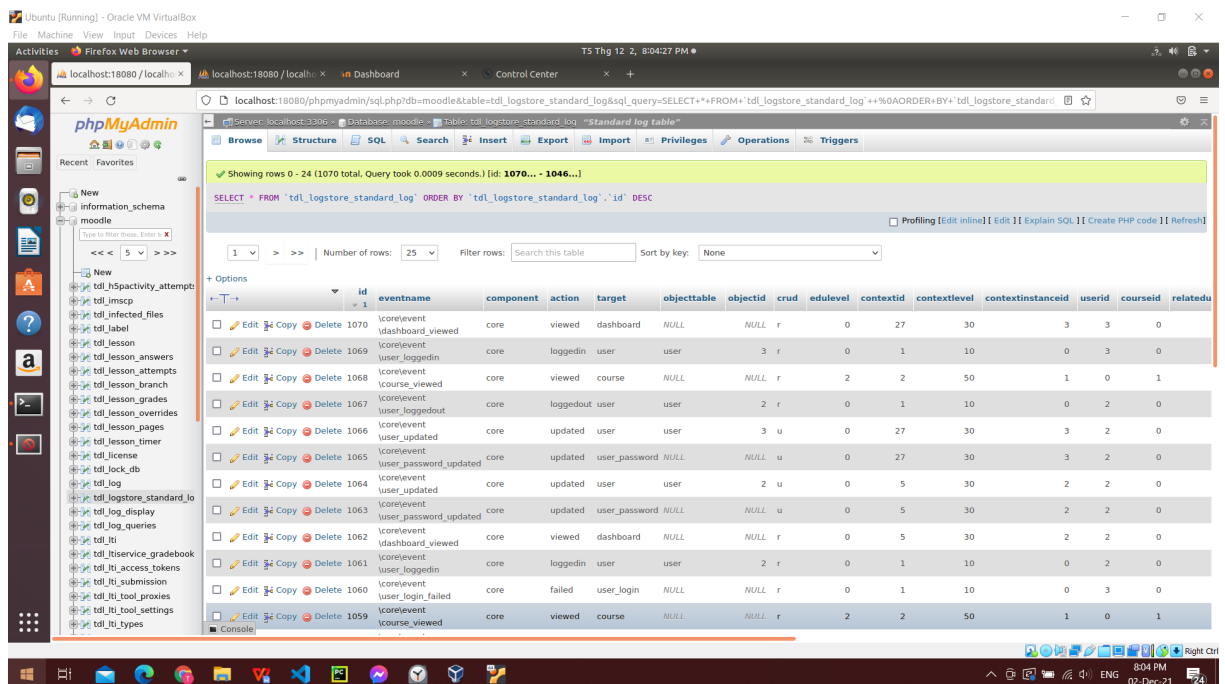
Hình 6.4: Bảng tdl_logstore_standard_log sau khi login



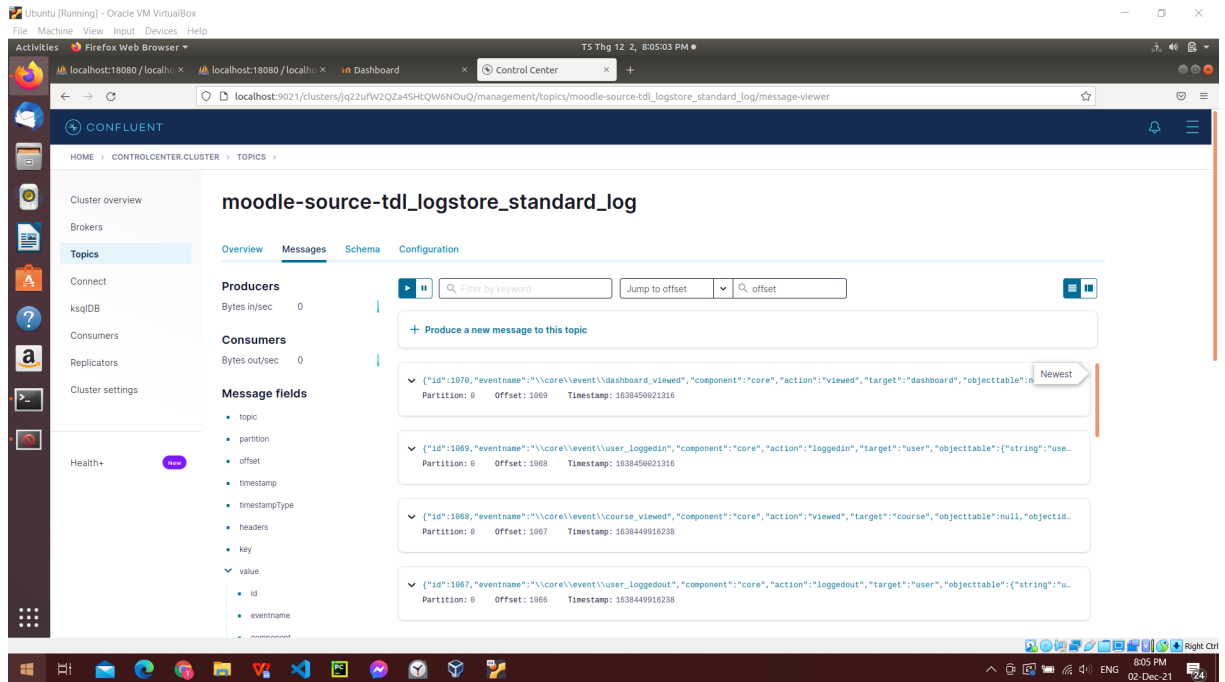
Hình 6.5: Confluent sau khi login

6.2 Logout

Trước khi logout:



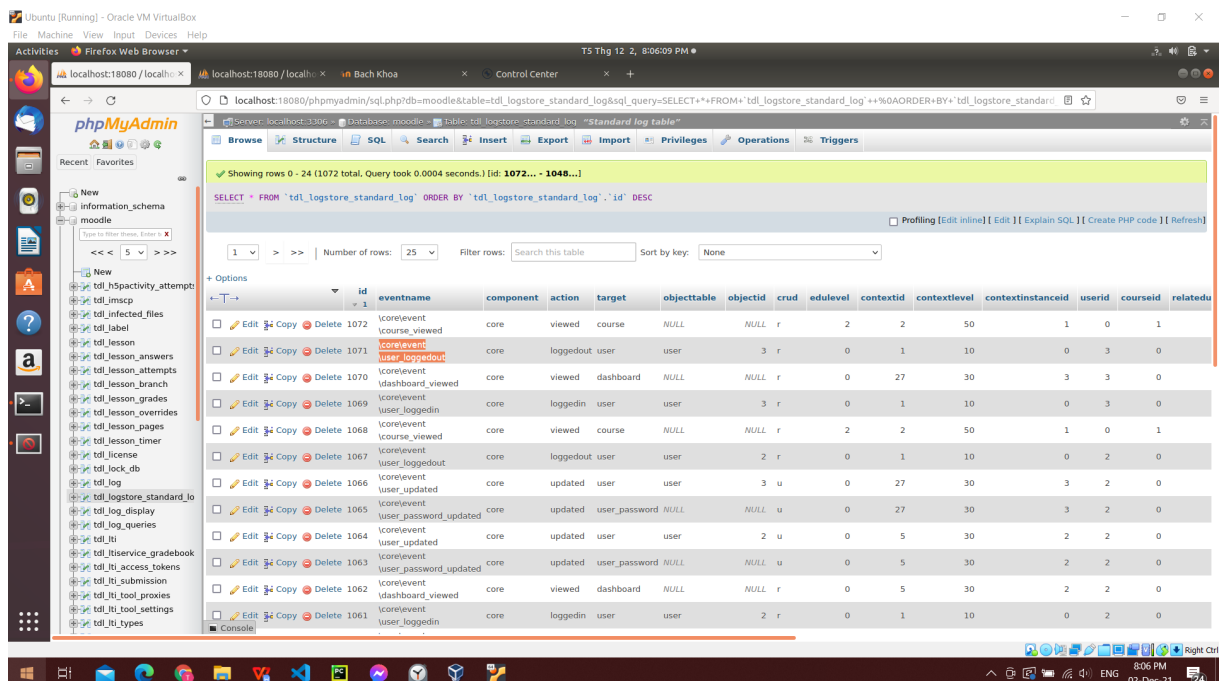
Hình 6.6: Bảng tdl_logstore_standard_log trước khi logout



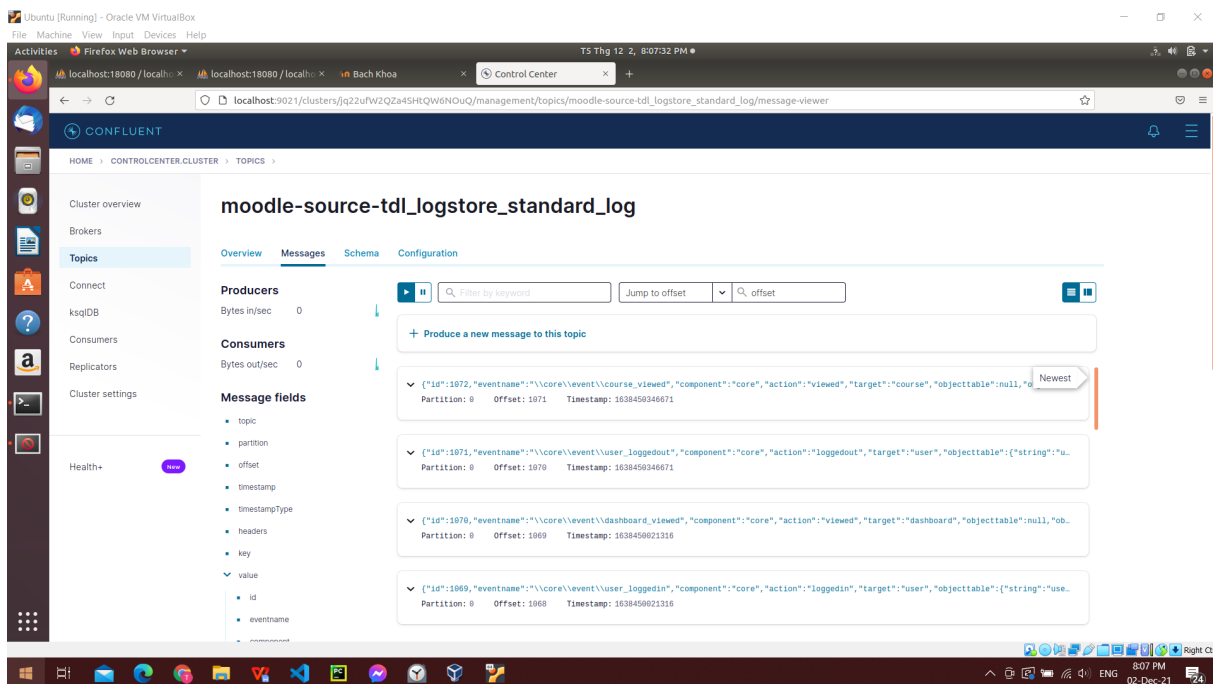
Hình 6.7: Confluent trước khi logout

Sau khi logout:

- Bảng tdl_logstore_standard_log có thêm dòng id=1071, action=loggedout, userid=3.
- Confluent đã nhận được message có id=1071.



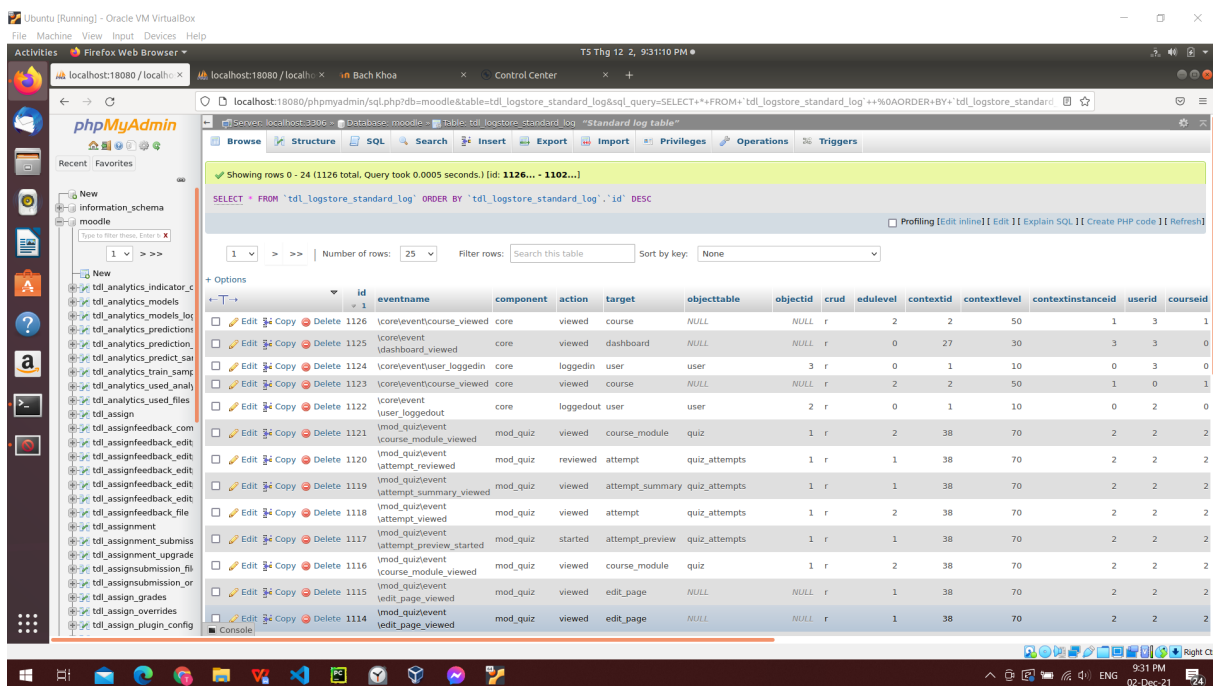
Hình 6.8: Bảng tdl_logstore_standard_log sau khi logout



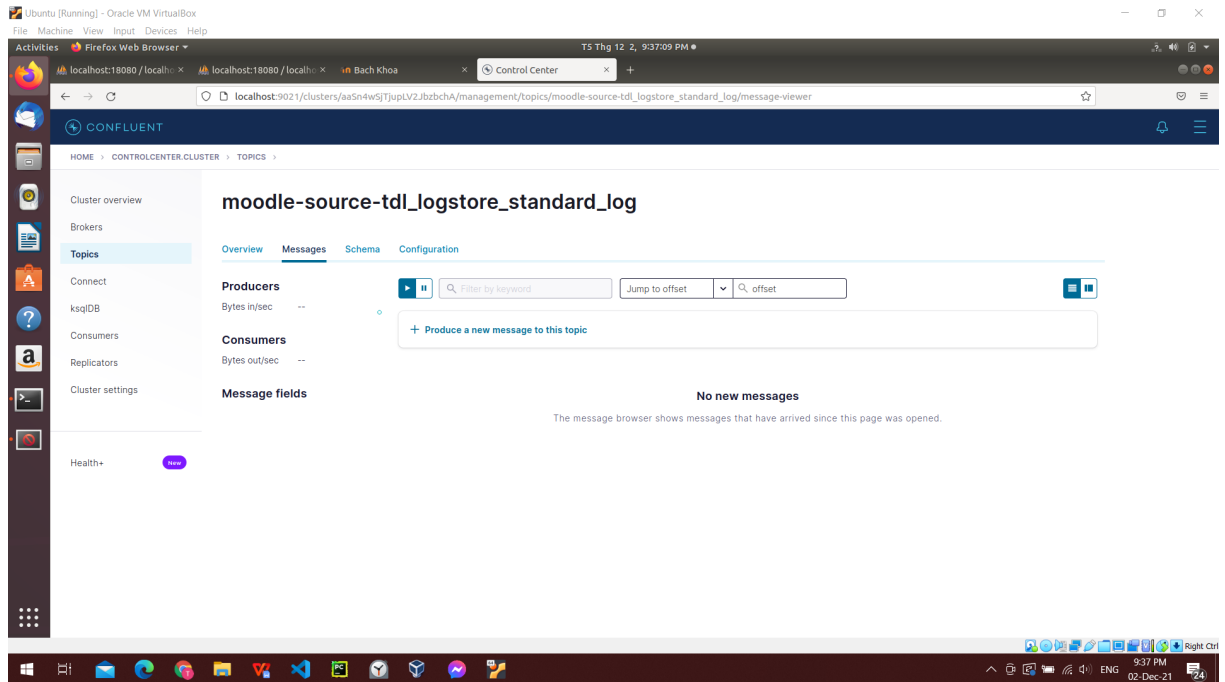
Hình 6.9: Confluent sau khi logout

6.3 View course

Trước khi view course:



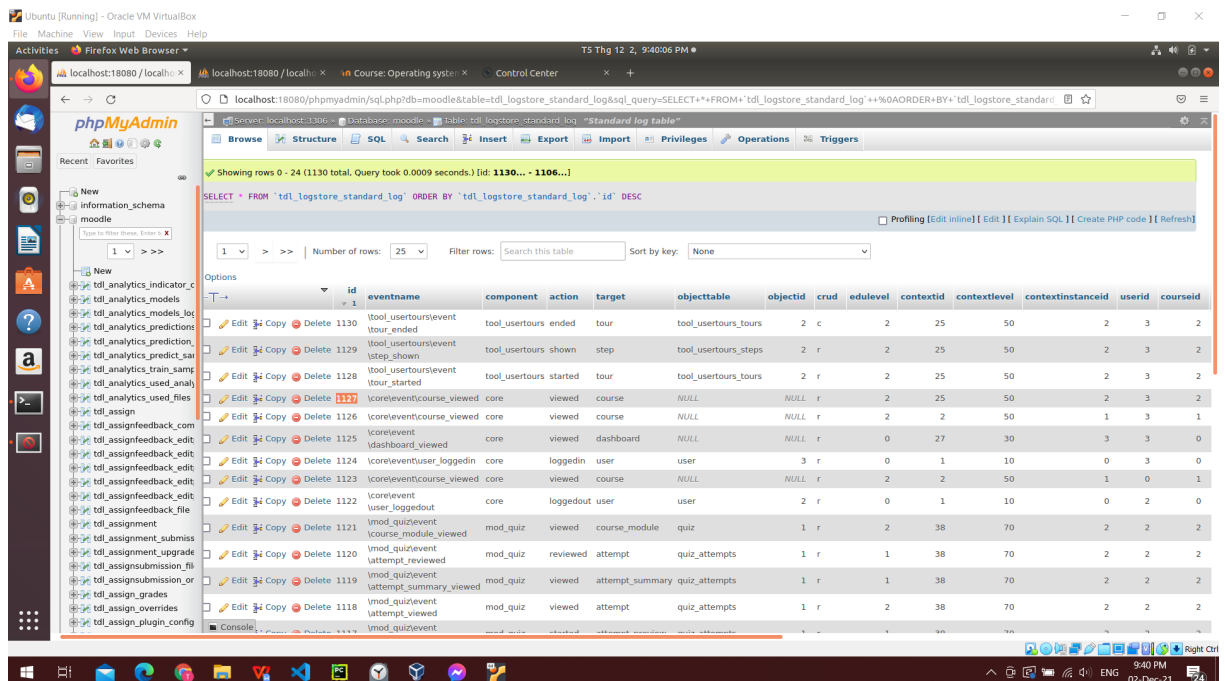
Hình 6.10: Bảng tdl_logstore_standard_log trước khi view course



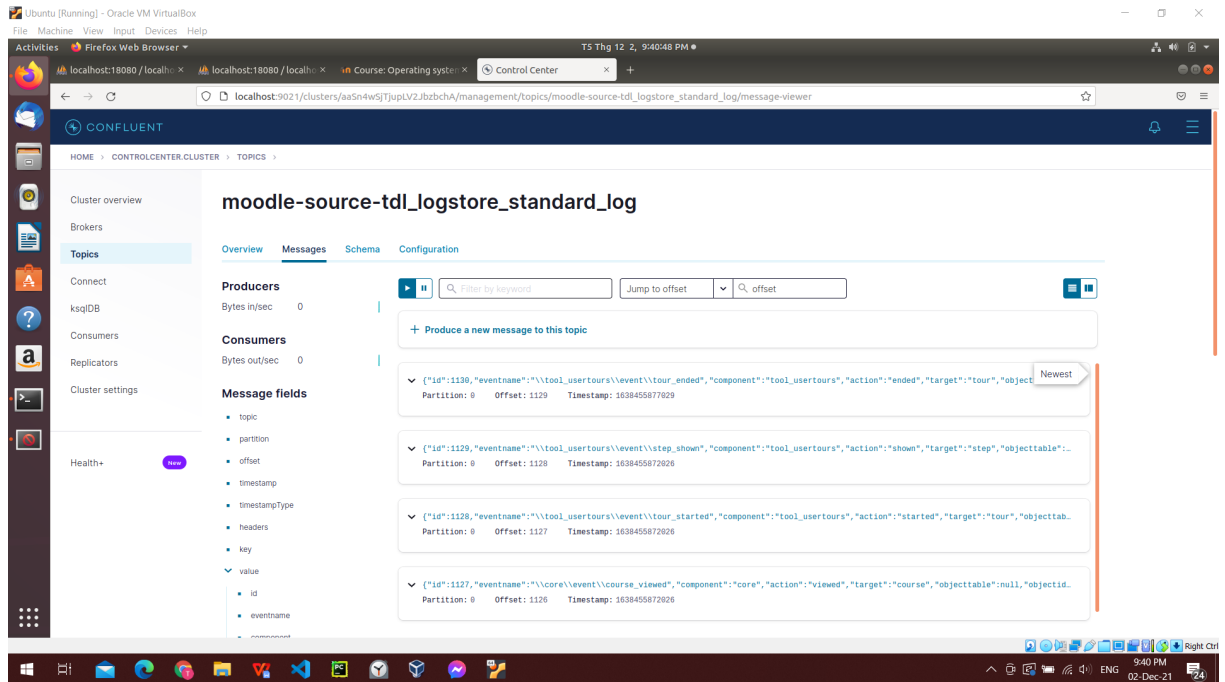
Hình 6.11: Confluent trước khi view course

Sau khi view course:

- Bảng tdl_logstore_standard_log có thêm dòng id=1127, action=viewed, target=course, userid=3, courseid=2.
- Confluent đã nhận được message có id=1127.



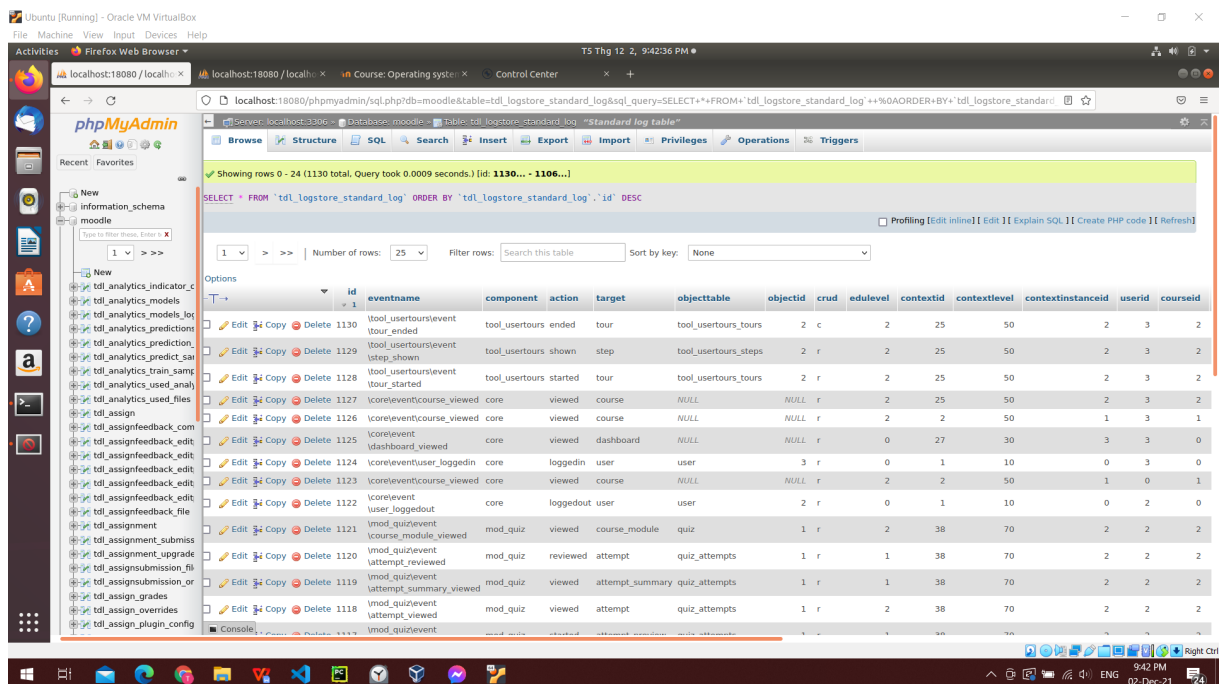
Hình 6.12: Bảng tdl_logstore_standard_log sau khi view course



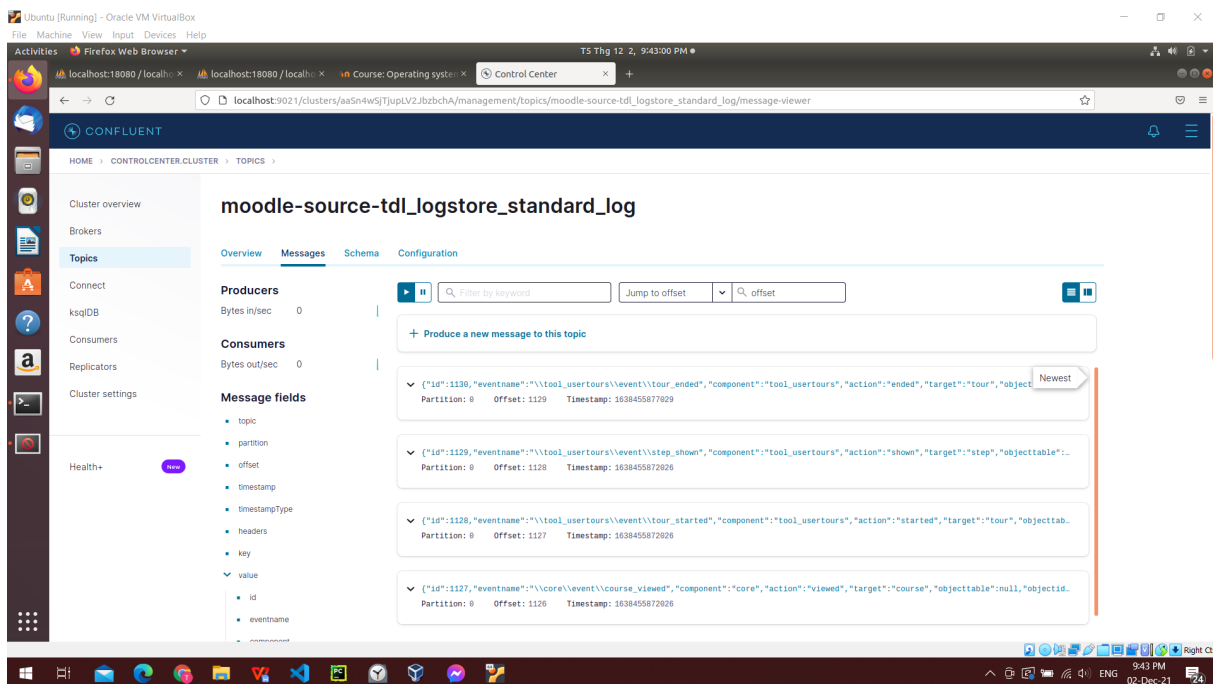
Hình 6.13: Confluent sau khi view course

6.4 View, start and submit quiz attempt

Trước khi view, start and submit quiz attempt:



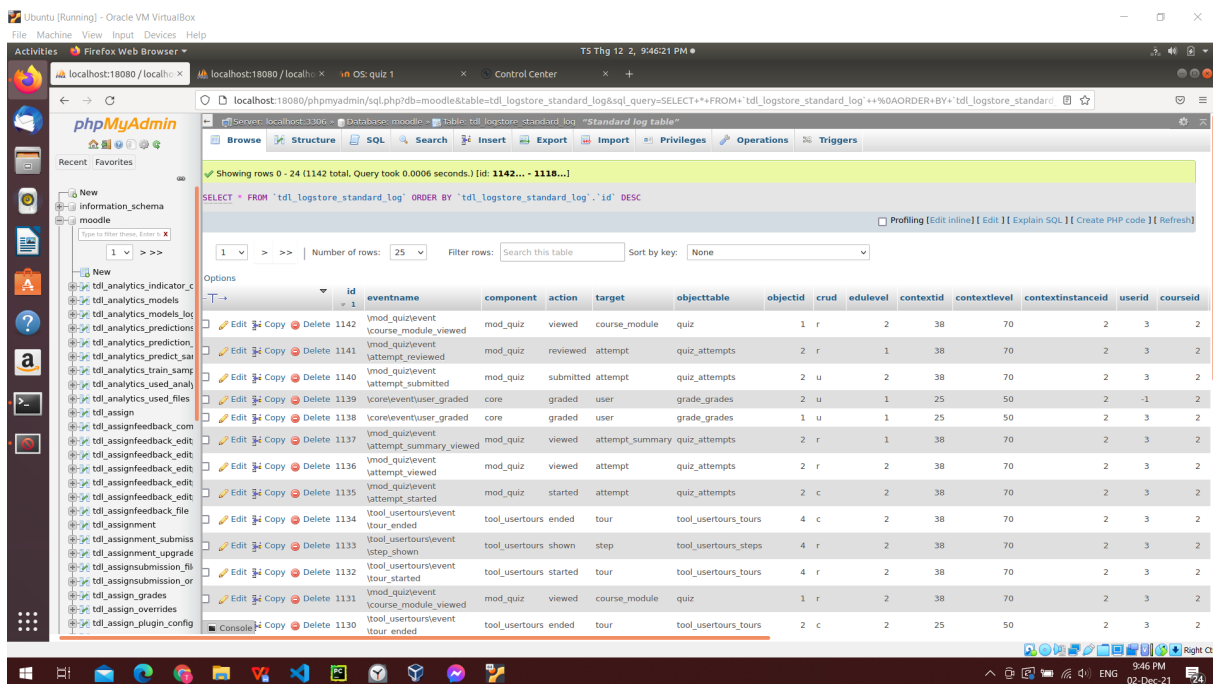
Hình 6.14: Bảng tdl_logstore_standard_log trước khi view, start and submit quiz attempt



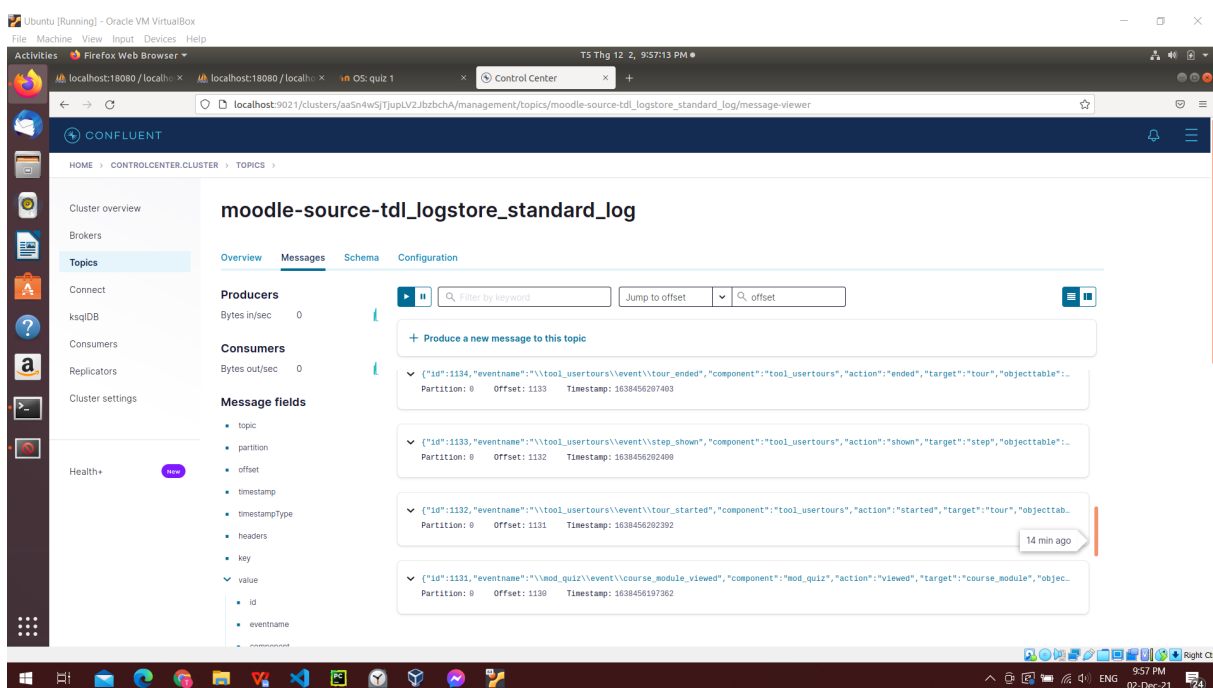
Hình 6.15: Confluent trước khi view, start and submit quiz attempt

Sau khi view, start and submit quiz attempt:

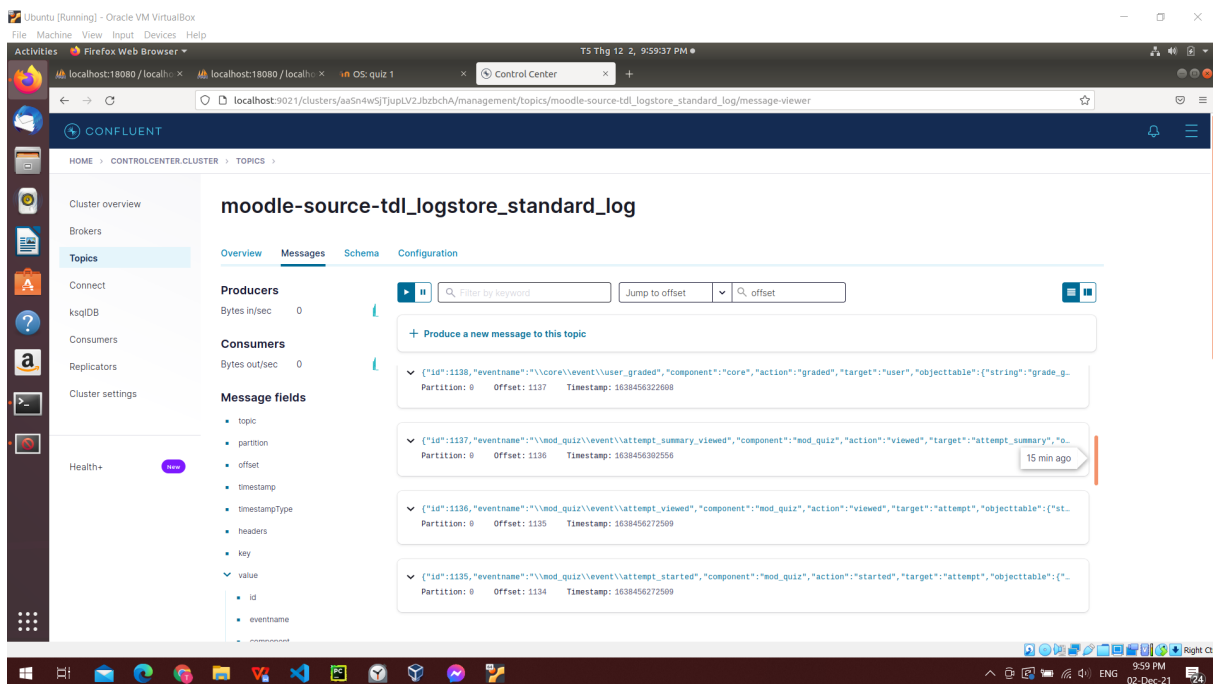
- Bảng tdl_logstore_standard_log có thêm các dòng:
 - id=1131, action=viewed, target=course_module, objecttable=quiz, userid=3, courseid=2.
 - id=1135, action=started, target=attempt, objecttable=quiz_attempts, userid=3, courseid=2
 - id=1136, action=viewed, target=attempt, objecttable=quiz_attempts, userid=3, courseid=2
 - id=1137, action=viewed, target=attempt_summary, objecttable=quiz_attempts, userid=3, courseid=2
 - id=1138, action=graded, target=user, objecttable=grade_grades, userid=3, courseid=2
 - id=1140, action=submitted, target=attempt, objecttable=quiz_attempts, userid=3, courseid=2
 - id=1141, action=reviewed, target=attempt, objecttable=quiz_attempts, userid=3, courseid=2
- Confluent đã nhận được các message có id lần lượt là 1131, 1135, 1136, 1137, 1138, 1140, 1141.



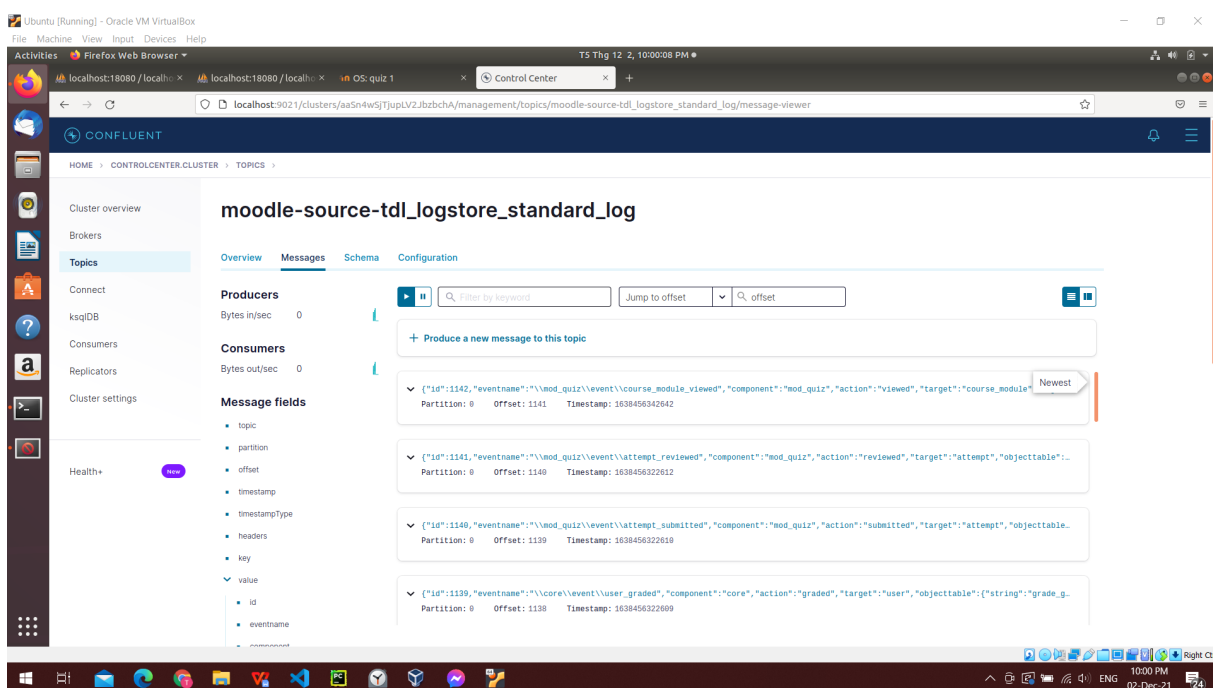
Hình 6.16: Bảng tdl_logstore_standard_log sau khi view, start and submit quiz attempt



Hình 6.17: Confluent sau khi view, start and submit quiz attempt (1)



Hình 6.18: Confluent sau khi view, start and submit quiz attempt (2)



Hình 6.19: Confluent sau khi view, start and submit quiz attempt (3)

7 Tổng kết

- Kết quả: Nhóm đã hiện thực được việc thu thập dữ liệu là các hoạt động của người học ở trang Moodle, dữ liệu lưu ở log của Moodle được realtime streaming đầy đủ và chính xác sang Kafka với Confluent Platform.
- Hướng phát triển của đề tài: Với kết quả trên, ta có thể triển khai cluster gồm nhiều node để thu



thập được dữ liệu lớn và phong phú hơn, từ đó ta xử lý dữ liệu và huấn luyện mô hình để dự đoán kết quả học tập của người học.

Tài liệu tham khảo

- [1] Martin Dougiamas. *Moodle*. URL: <https://moodle.org/>.
- [2] Sumyea Helal & Jiuyong Li & Lin Liu & Esmaeil Ebrahimie & Shane Dawson & Duncan J. Murray & Qi Long. *Predicting academic performance by considering student heterogeneity*. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0950705118303939>.
- [3] LinkedIn & Jay Kreps & Neha Narkhede & Jun Rao. *Kafka Connect*. URL: <https://docs.confluent.io/platform/current/connect/index.html>.
- [4] LinkedIn & Jay Kreps & Neha Narkhede & Jun Rao. *Kafka Stream*. URL: <https://kafka.apache.org/documentation/streams/>.
- [5] QandA Junction. *Cài đặt LAMP và Moodle*. URL: https://www.youtube.com/watch?v=3Xc0cF0Nh04&ab_channel=QandAJunction.
- [6] Jay Kreps & Neha Narkhede & Jun Rao. *Cài đặt Confluent*. URL: <https://www.confluent.io/get-started/?product=software>.
- [7] Confluent Inc. *Cài đặt JDBC Connector*. URL: <https://docs.confluent.io/kafka-connect-jdbc/current/>.