**tutorial-0.md**

**Xây Dựng Ứng Dụng Phân Tích Log Server**

**Giới Thiệu**

Các vấn đề về An Ninh Mạng là mối lo ngại phổ biến đối với doanh nghiệp với câu hỏi là khi nào nó sẽ xảy ra? Một trong những đường phòng ngự đầu tiên để phát hiện các lỗ hổng tiềm ẩn trong mạng là điều tra các logs từ máy chủ của bạn. Bạn đã được đưa vào để áp dụng kỹ năng của mình trong lĩnh vực Kỹ thuật Dữ liệu và Phân Tích Dữ liệu để thu thập dữ liệu log máy chủ, tiền xử lý dữ liệu và lưu trữ nó vào hệ thống lưu trữ phân tán đáng tin cậy HDFS bằng cách sử dụng framework quản lý dữ liệu Apache NiFi. Bạn sẽ cần tiếp tục làm sạch và tinh chỉnh dữ liệu bằng Apache Spark để có cái nhìn cụ thể về những hoạt động đang diễn ra trên máy chủ của bạn, như là máy chủ nào thường xuyên truy cập máy chủ và quốc gia hoặc thành phố nào gây ra nhiều lưu lượng mạng nhất với máy chủ của bạn. Sau đó, bạn sẽ hóa và mô tả những sự kiện này bằng cách sử dụng cuốn sổ nghệ thuật dữ liệu Apache Zeppelin để có thể kể một câu chuyện về những hoạt động đang diễn ra trên máy chủ và xem liệu có điều gì đội của bạn nên cảnh báo.

**Công Nghệ Big Data Sử Dụng Để Phát Triển Ứng Dụng:**

Tập dữ liệu Logs máy chủ của NASA

Hộp cát HDF

* Apache Ambari
* Apache NiFi

Hộp cát HDP

* Apache Hadoop - HDFS
* Apache Spark
* Apache Zeppelin

**Mục Tiêu và Đối Tượng**

Tìm hiểu về dữ liệu log máy chủ, phân tích dữ liệu log, cách nó hoạt động, các ứng dụng khác nhau và các quy ước tốt nhất có thể được sử dụng trong phân tích log máy chủ.

Tìm hiểu cách xây dựng một luồng dữ liệu NiFi để thu thập dữ liệu log máy chủ.

Tìm hiểu cách làm sạch dữ liệu để lọc xuống các thông điệp có thể thông báo cho người dùng về các hoạt động diễn ra trên máy chủ của họ bằng Spark.

Tìm hiểu cách trực quan hóa kết quả của bạn sau khi làm sạch dữ liệu bằng cách sử dụng Zeppelin.

**Yêu Cầu Tiên Quyết**

Tải và triển khai Hortonworks Data Platform (HDP) Sandbox.

Đọc qua Hướng dẫn Sử dụng Hộp Cát HDP để thiết lập ánh xạ tên máy chủ thành địa chỉ IP.

Nếu bạn không có ít nhất 20GB RAM cho HDP Sandbox và 4GB RAM cho máy của bạn, thì tham khảo Hướng dẫn Triển khai Hộp Cát Hortonworks trên Microsoft Azure.

Kích hoạt Kiến Trúc Dữ Liệu Kết Nối:

Kích hoạt CDA cho VirtualBox

Kích hoạt CDA cho VMware

Kích hoạt CDA cho Docker

**Khung Chương Trình**

Dự án hướng dẫn bao gồm các mô-đun hướng dẫn sau đây:

Khái Niệm Phát Triển Ứng Dụng: Bao gồm thông tin về dữ liệu log máy chủ, phân tích dữ liệu log, cách nó hoạt động, các ứng dụng khác nhau và một số quy ước tốt nhất có thể sử dụng trong phân tích log máy chủ.

Thiết Lập Môi Trường Phát Triển: Bạn sẽ thực hiện cấu hình các dịch vụ phần mềm và/hoặc cài đặt các phụ thuộc cho các dịch vụ phần mềm cần thiết để phát triển ứng dụng.

Thu Thập Dữ Liệu Log Máy Chủ NASA: Bạn sẽ học cách xây dựng một luồng dữ liệu NiFi để thu thập 2 tháng dữ liệu log NASA, tiền xử lý dữ liệu và lưu trữ nó vào HDFS.

Làm Sạch Dữ Liệu Log Máy Chủ NASA: Bạn sẽ học cách tạo một Cuốn sổ Zeppelin để làm sạch dữ liệu log NASA và sử dụng Trình thông dịch Spark của Zeppelin để làm sạch dữ liệu và thu thập bất kỳ thông tin giá trị nào về các hoạt động đang diễn ra trên máy chủ.

Trực Quan Hóa Dữ Liệu Log NASA: Bạn sẽ tạo một Cuốn sổ Zeppelin khác có mục tiêu là trực quan hóa các điểm chính bạn tìm thấy khi làm sạch dữ liệu bằng Spark. Trực quan hóa của bạn sẽ mô tả từ dữ liệu log NASA, Các Máy Chủ Thường Xuyên Nhất - số lần truy cập mỗi địa chỉ IP máy chủ, Mã Phản Hồi - số lần mỗi mã phản hồi kèm theo máy chủ, Loại Phần Mở Rộng - số lượng loại định dạng tệp được chuyển giữa các thiết bị tương tác với máy chủ, Lưu Lượng Mạng Theo Vị Trí - vị trí nơi các lượt truy cập máy chủ đang đến.

**tutorial-1.md**

**Khái Niệm Phát Triển Ứng Dụng**

**Giới Thiệu**

Các sự cố an ninh xảy ra. Và khi chúng xảy ra, logs máy chủ có thể là đường phòng ngự tốt nhất của bạn. Hadoop đưa phân tích log máy chủ lên một tầm cao mới bằng cách tăng tốc và cải thiện pháp truy cứu an ninh và cung cấp một nền tảng chi phí thấp để thể hiện sự tuân thủ.

**Yêu Cầu Tiên Quyết**

* Đọc tổng quan về loạt bài hướng dẫn
* Chương trình

**Dữ Liệu Log Máy Chủ**

Logs máy chủ là các tệp log do máy tính tạo ra để ghi lại dữ liệu về hoạt động mạng và máy chủ. Chúng hữu ích để quản lý các hoạt động mạng, đặc biệt là đối với an ninh và tuân thủ quy định.

**Bộ Dữ Liệu Logs Máy Chủ NASA**

Bộ dữ liệu mà chúng ta sẽ sử dụng trong thí nghiệm này là của NASA-HTTP. Nó chứa các yêu cầu HTTP đến máy chủ WWW Trung tâm Vũ trụ Kennedy của NASA ở Florida. Logs này là tệp ASCII với một dòng cho mỗi yêu cầu, với các cột sau:

* Máy chủ thực hiện yêu cầu. Một tên máy chủ khi có thể, nếu không sẽ là địa chỉ Internet nếu tên không thể được tra cứu được.
* Dấu thời gian theo định dạng "DAY MON DD HH:MM:SS YYYY", với DAY là ngày trong tuần, MON là tên tháng, DD là ngày trong tháng, HH:MM:SS là thời gian trong ngày sử dụng đồng hồ 24 giờ, và YYYY là năm. Múi giờ là -0400.
* Yêu cầu được đặt trong dấu ngoặc kép.
* Mã phản hồi HTTP.
* Số byte trong phản hồi.

**Log Analysis là Gì?**

Phân Tích Log Máy Chủ là một đánh giá về các bản ghi được tạo ra bởi máy tính, mạng hoặc các hệ thống IT khác. Phân tích log được sử dụng bởi tổ chức để giảm nhiều rủi ro và tuân thủ các tiêu chuẩn quy định.

**Làm Thế Nào Phân Tích Log Hoạt Động?**

Logs được tạo thành từ nhiều tin nhắn theo thứ tự thời gian và được lưu trữ trên đĩa, tệp, hoặc một ứng dụng như một bộ thu log. Người phân tích dữ liệu chịu trách nhiệm đảm bảo các logs chứa một loạt tin nhắn và được giải thích theo ngữ cảnh. Chuẩn hóa thường được thực hiện để giải quyết trường hợp khi có những không nhất quán nhỏ giữa các logs, như một tệp log chứa WARN và một tệp khác chứa CRITICAL, nhưng có thể trong ngữ cảnh chúng có nghĩa giống nhau. Sau khi dữ liệu log được thu thập và làm sạch, nó có thể được phân tích để phát hiện các mẫu và bất thường như xâm nhập mạng.

**Các Ứng Dụng Cho Phân Tích Log**

Các tổ chức IT sử dụng phân tích log máy chủ để trả lời các câu hỏi về:

* Tuân Thủ – Các tổ chức lớn bị ràng buộc bởi các quy định như HIPAA và Sarbanes-Oxley. Làm thế nào các quản trị viên IT có thể chuẩn bị cho các kiểm toán hệ thống? Trong bài thực hành này, chúng ta sẽ tập trung vào một trường hợp sử dụng an ninh mạng. Cụ thể, chúng ta sẽ xem cách Apache Hadoop có thể giúp quản trị viên của một mạng doanh nghiệp lớn chẩn đoán và phản ứng với cuộc tấn công từ chối dịch vụ phân tán.
* An Ninhs – Ví dụ, nếu chúng ta nghi ngờ có một sự xâm phạm an ninh, làm thế nào chúng ta có thể sử dụng dữ liệu log máy chủ để xác định và khắc phục lỗ hổng?
* Sửa Chữa - Sửa lỗi hệ thống, máy tính và vấn đề mạng.
* Hành Vi Người Dùng - Hiểu rõ hơn về người dùng của bạn.
* Phân Tích Hồ Sơ Máy Tính - Tiến hành trong trường hợp điều tra.

**Quy Ước Tốt Nhất Cho Phân Tích Log**

Nhận Diện và Nhận Biết Mẫu

* Giúp phát hiện các bất thường bằng cách lọc thông điệp qua việc hiểu biết về các mẫu trong dữ liệu.

Chuẩn Hóa

* Thiết lập một định dạng chung giữa các yếu tố log, như thiết lập một dấu thời gian vào cùng một định dạng.

Đánh Dấu và Phân Loại

* Lọc và điều chỉnh cách bạn muốn hiển thị dữ liệu của bạn bằng cách đánh dấu các yếu tố log với từ khóa và phân loại chúng vào một số lớp mà bạn có thể thực hiện lọc.

Phân Tích Tương Quan

* Giúp khám phá mối quan hệ giữa dữ liệu không thể nhìn thấy trong một log duy nhất. Trong trường hợp tấn công mạng gần đây, phân tích tương quan có thể tìm thấy các thông điệp liên quan đến cuộc tấn công cụ thể bằng cách kết hợp các logs được tạo ra bởi các máy chủ, tường lửa, thiết bị mạng và nguồn khác. Dữ liệu thu thập được từ phân tích tương quan có thể hỗ trợ trong việc tạo ra cảnh báo khi các mẫu cụ thể xảy ra trong các log.

Sự Phớt Tay

* Là quy trình học máy không quan tâm đến các thông điệp log thường xuyên không hữu ích và cho phép các thông điệp bất thường được phát hiện và đánh dấu để điều tra, có thể hóa ra là những bất thường.

**Tóm Tắt**

Chúc mừng! Bây giờ bạn đã quen với khái niệm về dữ liệu log máy chủ, phân tích dữ liệu log, cách phân tích dữ liệu hoạt động, một số ứng dụng của phân tích log máy chủ và một số thực hành đề xuất để áp dụng khi phân tích logs máy chủ của bạn. Hãy bắt đầu xây dựng ứng dụng phân tích log máy chủ bằng cách thiết lập môi trường phát triển trong bài hướng dẫn tiếp theo.

**Đọc Thêm**

* Log Analysis là Gì? Use Cases, Best Practices, và Nhiều Hơn Nữa
* Làm thế nào để phát hiện sự xâm phạm an ninh mạng?
* Vấn đề với Học Máy là gì?
* Lockheed Martin. Các Kiến Trúc Phòng Vệ
* Gartner về Sự Lừa Dối
* Gartner về Phân Tích Hành Vi Người Dùng và Thực Thể
* Nền Tảng An Ninh Mạng của Hortonworks

**tutorial-2.md**

**Thiết lập môi trường phát triển**

**Giới Thiệu**

Trong phần này của dự án với vai trò là một Chuyên viên Dữ liệu, bạn có trách nhiệm thiết lập môi trường phát triển như sau: đảm bảo cả HDP và HDF CentOS7 có thể giải quyết tên miền, trên HDF tải tệp dữ liệu GeoLite2 Database, trên HDF tải Nhật ký NASA, trên HDF làm sạch bảng NiFi nếu có bất kỳ luồng nào từ một dự án cũ và trên HDP đảm bảo rằng chế độ bảo trì của Spark đã tắt. Sau khi hoàn thành những mục này, chúng ta sẽ sẵn sàng bắt đầu xây dựng ống dữ liệu.

**Điều Kiện Tiên Quyết**

* Đã kích hoạt CDA cho hệ thống tương ứng của bạn.
* Tổng Quan về Mã Shell Sử Dụng Ở Cả Hai Phương Pháp
* Phương pháp 1: Thiết Lập Môi Trường Phát Triển Thủ Công
* Phương pháp 2: Thiết Lập Môi Trường Phát Triển Tự Động
* Tóm Tắt
* Đọc Thêm

**Xác Nhận Đã Bao Gồm Điều Kiện Tiên Quyết**

Ánh Xạ địa chỉ IP của sandbox thành tên máy chủ mong muốn trong tệp hosts

* Nếu bạn cần trợ giúp về cách ánh xạ địa chỉ IP của Sandbox thành tên máy chủ, tham khảo Môi Trường Thiết Lập -> Ánh Xạ Địa Chỉ IP Của Sandbox Thành Tên Máy Chủ Mong Muốn Trong Tệp Hosts trong Khám Phá HDP Sandbox

Thiết lập mật khẩu quản trị Ambari cho "HDF" và "HDP"

Nếu bạn cần trợ giúp thiết lập mật khẩu quản trị Ambari,

* Đối với HDP, tham khảo Đặt Lại Mật Khẩu Quản Trị trong Khám Phá HDP Sandbox
* Đối với HDF, tham khảo Đặt Lại Mật Khẩu Quản Trị trong Khám Phá HDF Sandbox

Bắt đầu tất cả các dịch vụ cần thiết cho "HDF" và "HDP"

Nếu không chắc chắn, đăng nhập vào Bảng điều khiển quản trị Ambari

* đối với HDF tại http://sandbox-hdf.hortonworks.com:8080 và xác minh rằng NiFi đã khởi động, nếu không, hãy bắt đầu nó.
* đối với HDP tại http://sandbox-hdp.hortonworks.com:8080 và xác minh rằng HDFS, Spark2 và Zeppelin đã khởi động, nếu không, hãy khởi động chúng.

**Tổng Quan về Mã Shell Sử Dụng Ở Cả Hai Phương Pháp**

**Mã Shell HDF**

**setup-hdf.sh**

* Chức năng đợi đến khi trạng thái dịch vụ là STARTED hoặc INSTALLED(STOPPED)
* Thêm DNS công cộng của Google để giải quyết tên miền thành địa chỉ IP
* Tạo thư mục cho DB GeoLite và Nhật ký NASA
* Tải và giải nén DB GeoLite và Nhật ký NASA vào các thư mục thích hợp
* Dừng NiFi, sao lưu và xóa luồng NiFi hiện tại, khởi động NiFi để cập nhật thay đổi
* Mã Shell HDP

**setup-hdp.sh**

* Thêm DNS công cộng của Google để giải quyết tên miền thành địa chỉ IP
* Tắt chế độ bảo trì của Spark nếu nó đang bật.

**Phương Pháp 1: Thiết Lập Môi Trường Phát Triển Thủ Công**

**Thiết lập HDF**

Chúng ta sẽ sử dụng các lệnh shell để thiết lập các dịch vụ cần thiết trong nền tảng dữ liệu trong chuyển động và dữ liệu tại nơi nghỉ ngơi từ các máy chủ web shell của sandbox.

Mở HDF Sandbox Web Shell Client tại http://sandbox-hdf.hortonworks.com:4200.

Trước khi thực hiện mã shell, thay thế chuỗi "<Your-Ambari-Admin-Password>" trong dòng mã sau của code setup\_nifi "admin" "<Your-Ambari-Admin-Password>" ở dòng cuối với mật khẩu bạn đã tạo cho người dùng quản trị Ambari. Ví dụ, nếu mật khẩu quản trị Ambari của chúng ta được đặt thành yellowHadoop, thì dòng mã sẽ trông như sau: AMBARI\_USER\_PASSWORD="yellowHadoop"

Sao chép và dán mã từng dòng:

|  |
| --- |
| ##  # Sets up HDF Dev Environment, so User can focus on NiFi Flow Dev  # 1. Creates GeoFile directory and download in GeoFile DB  # 2. Backup existing NiFi flow on canvas  # 3. Uploads and Imports New NiFi flow onto canvas via NiFi Rest API  ##  DATE=`date '+%Y-%m-%d %H:%M:%S'`  LOG\_DIR\_BASE="/var/log/cda-sb/200"  mkdir -p $LOG\_DIR\_BASE/hdf  setup\_public\_dns()  {  echo "$DATE INFO: Adding Google Public DNS to /etc/resolve.conf"  echo "# Google Public DNS" | tee -a /etc/resolve.conf  echo "nameserver 8.8.8.8" | tee -a /etc/resolve.conf  echo "$DATE INFO: Checking Google Public DNS added to /etc/resolve.conf"  cat /etc/resolve.conf  # Log everything, but also output to stdout  echo "$DATE INFO: Executing setup\_public\_dns() bash function, logging to $LOG\_DIR\_BASE/hdf/setup-public-dns.log"  }  setup\_nifi()  {  echo "$DATE INFO: Setting Up HDF Dev Environment for Server Log Analysis App"  echo "$DATE INFO: Setting HDF\_AMBARI\_USER based on user input"  HDF\_AMBARI\_USER="$1" # $1: Expects user to pass "Ambari User" into the file  echo "$DATE INFO: Setting HDF\_AMBARI\_PASS based on user input"  HDF\_AMBARI\_PASS="$2" # $2: Expects user to pass "Ambari Admin Password" into the file  HDF\_CLUSTER\_NAME="Sandbox"  HDF\_HOST="sandbox-hdf.hortonworks.com"  HDF="hdf-sandbox"  AMBARI\_CREDENTIALS=$HDF\_AMBARI\_USER:$HDF\_AMBARI\_PASS  # Ambari REST Call Function: waits on service action completing  # Start Service in Ambari Stack and wait for it  # $1: HDF or HDP  # $2: Service  # $3: Status - STARTED or INSTALLED, but OFF  wait()  {  if [[ $1 == "hdp-sandbox" ]]  then  finished=0  while [ $finished -ne 1 ]  do  echo "$DATE INFO: Waiting for $1 $2 service action to finish"  ENDPOINT="http://$HDP\_HOST:8080/api/v1/clusters/$HDP\_CLUSTER\_NAME/services/$2"  AMBARI\_CREDENTIALS="$HDP\_AMBARI\_USER:$HDP\_AMBARI\_PASS"  str=$(curl -s -u $AMBARI\_CREDENTIALS $ENDPOINT)  if [[ $str == \*"$3"\* ]] || [[ $str == \*"Service not found"\* ]]  then  echo "$DATE INFO: $1 $2 service state is now $3"  finished=1  fi  echo "$DATE INFO: Still waiting on $1 $2 service action to finish"  sleep 3  done  elif [[ $1 == "hdf-sandbox" ]]  then  finished=0  while [ $finished -ne 1 ]  do  echo "$DATE INFO: Waiting for $1 $2 service action to finish"  ENDPOINT="http://$HDF\_HOST:8080/api/v1/clusters/$HDF\_CLUSTER\_NAME/services/$2"  AMBARI\_CREDENTIALS="$HDF\_AMBARI\_USER:$HDF\_AMBARI\_PASS"  str=$(curl -s -u $AMBARI\_CREDENTIALS $ENDPOINT)  if [[ $str == \*"$3"\* ]] || [[ $str == \*"Service not found"\* ]]  then  echo "$DATE INFO: $1 $2 service state is now $3"  finished=1  fi  echo "$DATE INFO: Still waiting on $1 $2 service action to finish"  sleep 3  done  else  echo "$DATE ERROR: Didn't Wait for Service, need to choose appropriate sandbox HDF or HDP"  fi  }  echo "$DATE INFO: Creating File Path to GeoLite DB and NASALogs for HDF NiFi"  GEODB\_NIFI\_DIR="/sandbox/tutorial-files/200/nifi"  mkdir -p $GEODB\_NIFI\_DIR/input/GeoFile  mkdir -p $GEODB\_NIFI\_DIR/input/NASALogs  mkdir -p $GEODB\_NIFI\_DIR/templates  chmod 777 -R $GEODB\_NIFI\_DIR  echo "$DATE INFO: Downloading and Extracting GeoLite DB for NiFi to /sandbox/tutorial-files/200/nifi/input/GeoFile/"  wget http://geolite.maxmind.com/download/geoip/database/GeoLite2-City.tar.gz \  -O $GEODB\_NIFI\_DIR/input/GeoFile/GeoLite2-City.tar.gz  tar -zxvf $GEODB\_NIFI\_DIR/input/GeoFile/GeoLite2-City.tar.gz \  -C $GEODB\_NIFI\_DIR/input/GeoFile/  echo "$DATE INFO: Removing GeoLite DB tar.gz file from /sandbox/tutorial-files/200/nifi/input/GeoFile/"  rm -rf $GEODB\_NIFI\_DIR/input/GeoFile/GeoLite2-City.tar.gz  echo "$DATE INFO: Downloading and Extracting NASALogs Aug1995 to /sandbox/tutorial-files/200/nifi/input/NASALogs/"  wget ftp://ita.ee.lbl.gov/traces/NASA\_access\_log\_Aug95.gz \  -O $GEODB\_NIFI\_DIR/input/NASALogs/NASA\_access\_log\_Aug95.gz  gunzip -c $GEODB\_NIFI\_DIR/input/NASALogs/NASA\_access\_log\_Aug95.gz \  > $GEODB\_NIFI\_DIR/input/NASALogs/NASA\_access\_log\_Aug95  echo "$DATE INFO: Removing NASALogs gz file from /sandbox/tutorial-files/200/nifi/input/NASALogs/"  rm -rf $GEODB\_NIFI\_DIR/input/NASALogs/NASA\_access\_log\_Aug95.gz  echo "$DATE INFO: Cleaning Up NiFi Canvas for NiFi Developer to build Cybersecurity Breach Analysis Flow..."  echo "$DATE INFO: Stopping HDF NiFi Service via Ambari REST Call"  #TODO: Check for status code for 400, then resolve issue  # List Services in HDF Stack  # curl -u $AMBARI\_CREDENTIALS -H "X-Requested-By: ambari" -X GET http://$HDF\_HOST:8080/api/v1/clusters/$HDF\_CLUSTER\_NAME/services/  curl -u $AMBARI\_CREDENTIALS -H "X-Requested-By: ambari" -X PUT -d '{"RequestInfo":  {"context": "Stop NiFi"}, "ServiceInfo": {"state": "INSTALLED"}}' \  http://$HDF\_HOST:8080/api/v1/clusters/$HDF\_CLUSTER\_NAME/services/NIFI  echo "$DATE INFO: Waiting on HDF NiFi Service to STOP RUNNING via Ambari REST Call"  wait $HDF NIFI "INSTALLED"  echo "$DATE INFO: HDF NiFi Service STOPPED via Ambari REST Call"  echo "$DATE INFO: Prebuilt HDF NiFi Flow removed from NiFi UI, but backed up"  mv /var/lib/nifi/conf/flow.xml.gz /var/lib/nifi/conf/flow.xml.gz.bak  echo "$DATE INFO: Starting HDF NiFi Service via Ambari REST Call"  curl -u $AMBARI\_CREDENTIALS -H "X-Requested-By: ambari" -X PUT -d '{"RequestInfo":  {"context": "Start NiFi"}, "ServiceInfo": {"state": "STARTED"}}' \  http://$HDF\_HOST:8080/api/v1/clusters/$HDF\_CLUSTER\_NAME/services/NIFI  echo "$DATE INFO: Waiting on HDF NiFi Service to START RUNNING via Ambari REST Call"  wait $HDF NIFI "STARTED"  echo "$DATE INFO: HDF NiFi Service STARTED via Ambari REST Call"  # Log everything, but also output to stdout  echo "$DATE INFO: Executing setup\_nifi() bash function, logging to $LOG\_DIR\_BASE/hdf/setup-nifi.log"  }  setup\_public\_dns | tee -a $LOG\_DIR\_BASE/hdf/setup-public-dns.log  setup\_nifi "admin" "<Your-Ambari-Admin-Password>" | tee -a $LOG\_DIR\_BASE/hdf/setup-nifi.log |

Mở HDF Sandbox Web Shell Client tại http://sandbox-hdp.hortonworks.com:4200.

Sao chép và dán mã từng dòng:

|  |
| --- |
| ##  # Sets up HDP Dev Environment, so User can focus on Spark Data Analysis  # 1. Add Google Public DNS to /etc/resolve.conf  # 2. Created Directory for Zeppelin Notebook, can be referenced later when auto importing Zeppelin Notebooks via Script  # 3. Created HDFS Directory for NiFi to have permission to write data  ##  DATE=`date '+%Y-%m-%d %H:%M:%S'`  LOG\_DIR\_BASE="/var/log/cda-sb/200"  echo "Setting Up HDP Dev Environment for Server Log Analysis App"  mkdir -p $LOG\_DIR\_BASE/hdp  setup\_public\_dns()  {  echo "$DATE INFO: Adding Google Public DNS to /etc/resolve.conf"  echo "# Google Public DNS" | tee -a /etc/resolve.conf  echo "nameserver 8.8.8.8" | tee -a /etc/resolve.conf  echo "$DATE INFO: Checking Google Public DNS added to /etc/resolve.conf"  cat /etc/resolve.conf  # Log everything, but also output to stdout  echo "$DATE INFO: Executing setup\_public\_dns() bash function, logging to $LOG\_DIR\_BASE/hdp/setup-public-dns.log"  }  setup\_zeppelin()  {  echo "$DATE INFO: Creating Directory for Zeppelin Notebooks"  mkdir -p /sandbox/tutorial-files/200/zeppelin/notebooks/  echo "$DATE INFO: Allowing read-write-execute permissions to any user, for zeppelin REST Call"  chmod -R 777 /sandbox/tutorial-files/200/zeppelin/notebooks/  # Log everything, but also output to stdout  echo "$DATE INFO: Executing setup\_zeppelin() bash function, logging to $LOG\_DIR\_BASE/hdp/setup-zeppelin.log"  }  setup\_hdfs()  {  # Creates /sandbox directory in HDFS  # allow read-write-execute permissions for the owner, group, and any other users  echo "$DATE INFO: Creating HDFS dir /sandbox/tutorial-files/200/nifi/ for HDF NiFi to write data"  sudo -u hdfs hdfs dfs -mkdir -p /sandbox/tutorial-files/200/nifi/  echo "$DATE INFO: Allowing read-write-execute permissions to any user, so NiFi has write access"  sudo -u hdfs hdfs dfs -chmod -R 777 /sandbox/tutorial-files/200/nifi/  echo "$DATE INFO: Checking directory was created and permissions were set"  sudo -u hdfs hdfs dfs -ls /sandbox/tutorial-files/200/  # Log everything, but also output to stdout  echo "$DATE INFO: Executing setup\_hdfs() bash function, logging to $LOG\_DIR\_BASE/hdp/setup-hdfs.log"  }  setup\_public\_dns | tee -a $LOG\_DIR\_BASE/hdp/setup-public-dns.log  setup\_zeppelin | tee -a $LOG\_DIR\_BASE/hdp/setup-zeppelin.log  setup\_hdfs | tee -a $LOG\_DIR\_BASE/hdp/setup-hdfs.log |

**Phương pháp 2: Tự Động Thiết Lập Môi Trường Phát Triển**

Chúng ta sẽ tải và thực thi một kịch bản shell để tự động hóa quá trình thiết lập nền tảng dữ liệu trong chuyển động và dữ liệu tại nơi nghỉ ngơi từ các máy chủ web shell của sandbox.

**Tự Động Thiết Lập HDF**

Mở HDF web shell client tại <http://sandbox-hdf.hortonworks.com:4200>.

Trước khi thực hiện kịch bản shell, hãy thay thế dòng mã shell sau AMBARI\_USER\_PASSWORD="<Your-Ambari-Admin-Password>" bằng mật khẩu bạn đã tạo cho người dùng quản trị Ambari. Ví dụ, nếu mật khẩu quản trị Ambari của chúng ta được đặt thành yellowHadoop, thì dòng mã sẽ trông như sau: AMBARI\_USER\_PASSWORD="yellowHadoop"

|  |
| --- |
| AMBARI\_USER="admin"  AMBARI\_USER\_PASSWORD="<Your-Ambari-Admin-Password>"  wget https://github.com/hortonworks/data-tutorials/raw/master/tutorials/cda/building-a-server-log-analysis-application/application/setup/shell/setup-hdf.sh  bash setup-hdf.sh $AMBARI\_USER $AMBARI\_USER\_PASSWORD |

Tự Động Thiết Lập HDP

Mở HDF web shell client tại <http://sandbox-hdp.hortonworks.com:4200>.

|  |
| --- |
| wget https://github.com/hortonworks/data-tutorials/raw/master/tutorials/cda/building-a-server-log-analysis-application/application/setup/shell/setup-hdp.sh  bash setup-hdp.sh |

**Tóm Tắt**

Chúc mừng bạn! Bạn đã đảm bảo rằng cả HDF và HDP CentOS7 đều có thể giải quyết tên miền. Do đó, bạn đã có thể tải xuống dữ liệu GeoLite2 DB và Nhật ký máy chủ NASA. Các phụ thuộc nền tảng để xây dựng ống dữ liệu đã được giải quyết và chúng ta hiện có thể tiến xa hơn bằng cách thu thập dữ liệu nhật ký máy chủ NASA với NiFi.

**Đọc Thêm**

* API REST của Ambari
* Nhật ký Máy chủ HTTP của NASA
* Cơ sở dữ liệu có thể tải miễn phí GeoLite2

**tutorial-3.md**

|  |
| --- |
| title |
| Acquiring NASA Server Log Data |

**Thu Thập Dữ Liệu Nhật Ký Máy Chủ NASA**

**Giới Thiệu**

Bạn đã được đưa vào dự án với vai trò là một Chuyên gia Dữ liệu với các trách nhiệm sau: thu thập nguồn dữ liệu Nhật ký máy chủ NASA, tiền xử lý dữ liệu và lưu trữ nó vào một hệ thống lưu trữ.

**Điều Kiện Tiên Quyết**

* Đã kích hoạt CDA cho hệ thống tương ứng của bạn.
* Thiết Lập Môi Trường Phát Triển

**Tổng Quan**

* Phương Pháp 1: Xây Dựng Luồng NiFi để Thu Thập Nhật Ký Máy Chủ NASA
* Phương Pháp 2: Nhập Luồng NiFi AcquireNASALogs qua Giao Diện Người Dùng
* Phương Pháp 3: Triển Khai Tự Động Luồng NiFi qua Cuộc Gọi REST
* Tóm Tắt
* Đọc Thêm

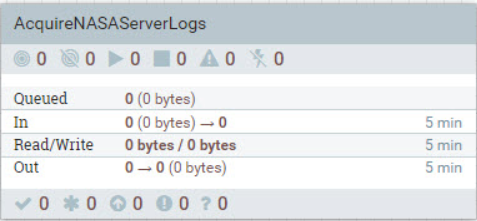
**Phương Pháp 1: Xây Dựng Luồng NiFi để Thu Thập Nhật Ký Máy Chủ NASA**

Sau khi bắt đầu sandbox, mở giao diện người dùng NiFi của HDF tại http://sandbox-hdf.hortonworks.com:9090/nifi.

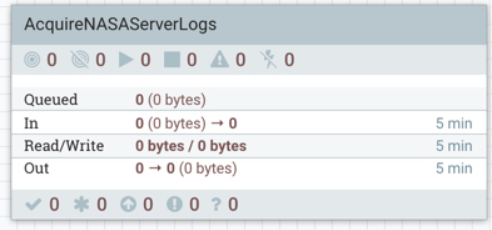
**Tạo Nhóm Quy Trình AcquireNASALogData**

Nhóm quy trình này thu thập dữ liệu nhật ký máy chủ của NASA từ tháng 8 năm 1995, tiền xử lý dữ liệu, lưu trữ nó vào Kafka, HDFS và hệ thống tệp cục bộ để phân tích sau này.

Kéo biểu tượng nhóm quy trình (process group) và thả nó lên bảng NiFi.



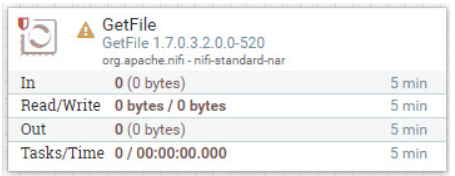
Nhập tên Nhóm Quy Trình: AcquireNASAServerLogs hoặc tùy chọn khác của bạn, sau đó nhấp vào THÊM.



Nhấp đúp vào nhóm quy trình để vào bên trong. Ở dưới cùng của bảng, bạn sẽ thấy dòng NiFi Flow >> AcquireNASAServerLogs. Hãy bắt đầu kết nối các bộ xử lý cho việc nhập dữ liệu, tiền xử lý và lưu trữ.

Nhập Nguồn Dữ Liệu Nhật Ký Máy Chủ NASA

Kéo biểu tượng bộ xử lý và thả nó lên bảng NiFi. Thêm bộ xử lý GetFile.



Giữ phím Control và nhấp chuột vào GetFile để cấu hình bộ xử lý:

| **Setting** | **Value** |
| --- | --- |
| Name | GrabNASALogs |

**Bảng 1: Tab Cài Đặt**

|  |  |
| --- | --- |
| **Setting** | **Value** |
| Name | GrabNASALogs |

**Bảng 2: Tab Lên Lịch**

|  |  |
| --- | --- |
| Scheduling | Value |
| Run Schedule | 60 sec |

**Bảng 3: Tab Thuộc Tính**

|  |  |
| --- | --- |
| **Property** | **Value** |
| **Input Directory** | /sandbox/tutorial-files/200/nifi/input/NASALogs |
| **Keep Source File** | true |

Click ÁP DỤNG.

**Chia tệp văn bản thành mỗi tệp 100,000 dòng**

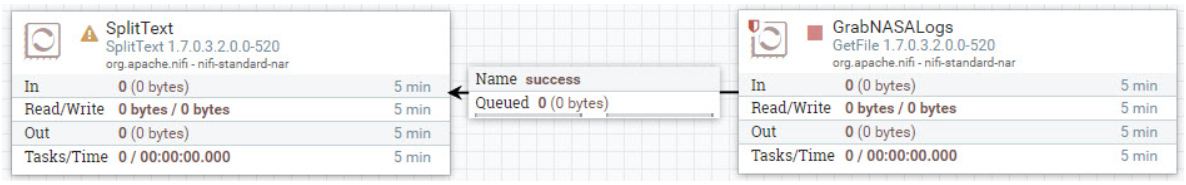
Kéo biểu tượng bộ xử lý và thả nó lên bảng NiFi. Thêm bộ xử lý SplitText.

Tạo kết nối giữa các bộ xử lý GetFile và SplitText. Di chuột qua GetFile để xem biểu tượng mũi tên, nhấn vào bộ xử lý và kết nối nó với SplitText.

Cấu hình Tạo Kết Nối:

|  |  |
| --- | --- |
| Connection | Value |
| For Relationships | success (checked) |

Click THÊM



Nhấn giữ Control + nhấp chuột lên SplitText để cấu hình bộ xử lý:

Bảng 1: Tab Cài Đặt

| **Setting** | **Value** |
| --- | --- |
| Name | Split100000LinesOfText |
| Automatically Terminate Relationships | failure (**checked**) |
| Automatically Terminate Relationships | original (**checked**) |

Bảng 2: Tab Lập Lịch

| **Scheduling** | **Value** |
| --- | --- |
| Run Schedule | 60 sec |

Bảng 3: Tab Thuộc Tính

| **Property** | **Value** |
| --- | --- |
| **Line Split Count** | 100000 |

Nhấp vào ÁP DỤNG.

Chia tệp văn bản thành các đoạn, mỗi đoạn chứa 10,000 dòng.

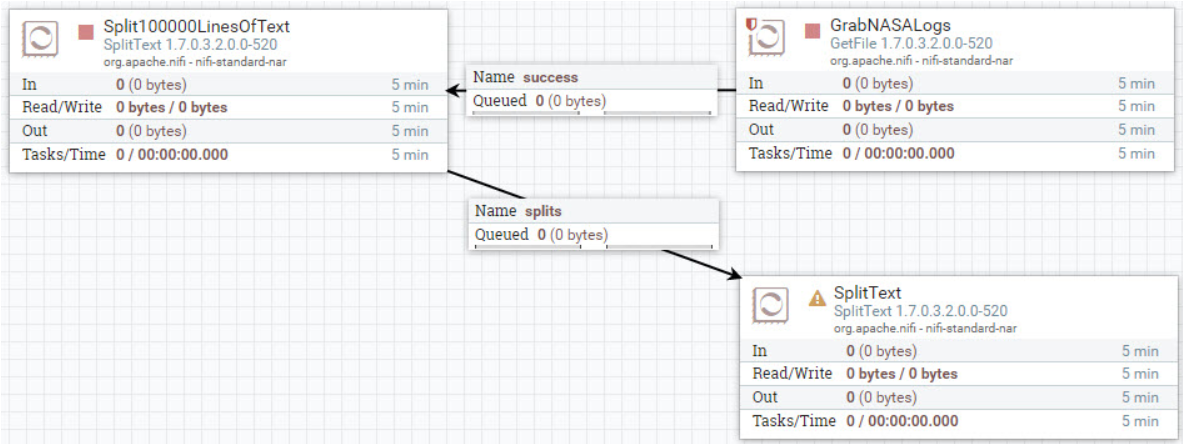
Kéo biểu tượng bộ xử lý và đặt nó lên bảng NiFi. Thêm bộ xử lý SplitText.

Tạo kết nối giữa các bộ xử lý Split100000LinesOfText và SplitText. Di chuột qua Split100000LinesOfText để xem biểu tượng mũi tên, nhấn vào bộ xử lý và kết nối nó với SplitText.

Cấu hình Tạo Kết Nối:

| **Connection** | **Value** |
| --- | --- |
| For Relationships | splits (**checked**) |

Click THÊM.



Nhấn giữ Control + nhấp chuột lên SplitText để cấu hình bộ xử lý:

Bảng 1: Tab Cài Đặt

| **Setting** | **Value** |
| --- | --- |
| Name | Split10000LinesOfText |
| Automatically Terminate Relationships | failure (**checked**) |
| Automatically Terminate Relationships | original (**checked**) |

Bảng 2: Tab Lập Lịch

| **Scheduling** | **Value** |
| --- | --- |
| Run Schedule | 60 sec |

Bảng 3: Tab Thuộc Tính

| **Property** | **Value** |
| --- | --- |
| **Line Split Count** | 10000 |

Click THÊM.

Chia tệp văn bản thành các đoạn, mỗi đoạn chứa 1,000 dòng.

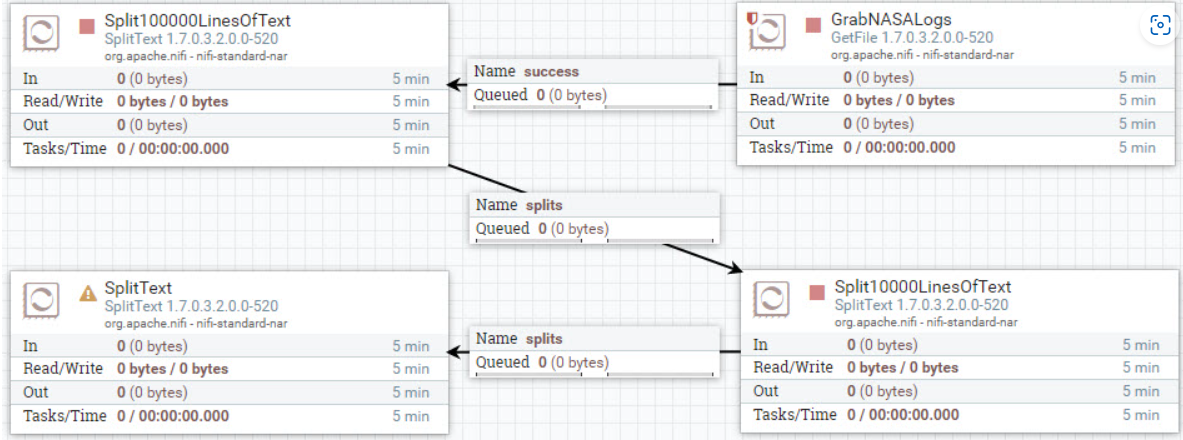
Kéo biểu tượng bộ xử lý và đặt nó lên bảng NiFi. Thêm bộ xử lý SplitText.

Tạo kết nối giữa các bộ xử lý Split10000LinesOfText và SplitText. Di chuột qua Split10000LinesOfText để xem biểu tượng mũi tên, nhấn vào bộ xử lý và kết nối nó với SplitText.

Cấu hình Tạo Kết Nối:

| **Connection** | **Value** |
| --- | --- |
| For Relationships | splits (**checked**) |

Click THÊM.



Nhấn giữ Control + nhấp chuột lên SplitText để cấu hình bộ xử lý:

Bảng 1: Tab Cài Đặt

| **Setting** | **Value** |
| --- | --- |
| Name | Split1000LinesOfText |
| Automatically Terminate Relationships | failure (**checked**) |
| Automatically Terminate Relationships | original (**checked**) |

Bảng 2: Tab Lập Lịch

| **Scheduling** | **Value** |
| --- | --- |
| Run Schedule | 10 sec |

Bảng 3: Tab Thuộc Tính

| **Property** | **Value** |
| --- | --- |
| **Line Split Count** | 1000 |

Nhấp vào ÁP DỤNG.

Chia tệp văn bản thành các đoạn, mỗi đoạn chứa 100 dòng.

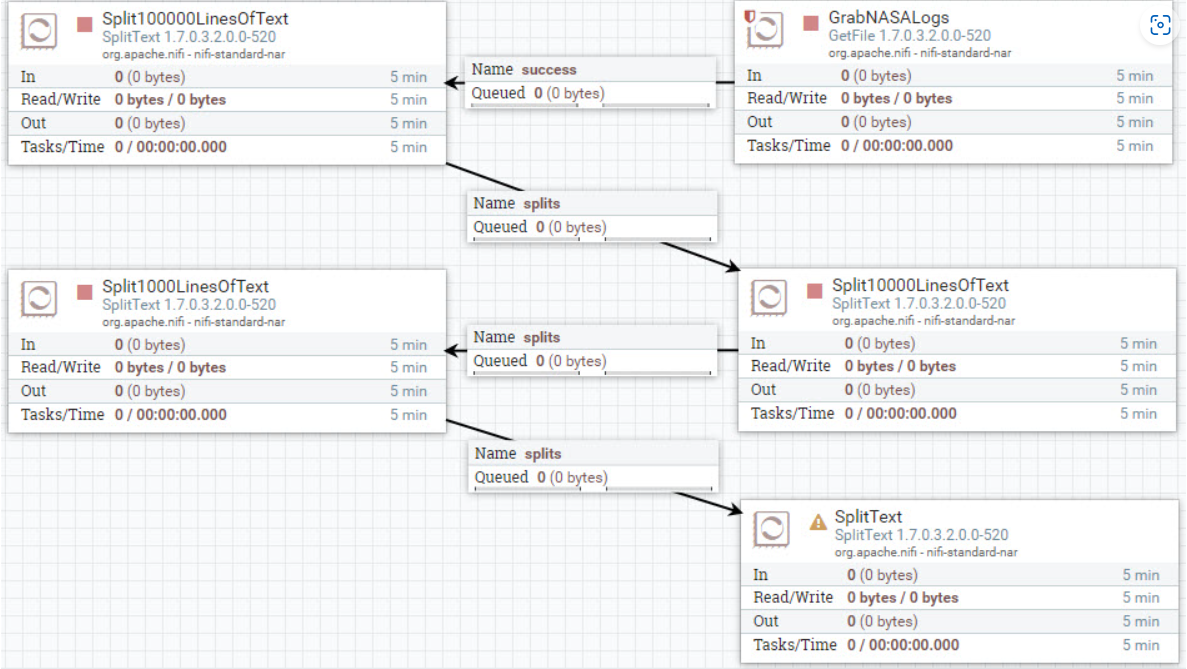
Kéo biểu tượng bộ xử lý và đặt nó lên bảng NiFi. Thêm bộ xử lý SplitText.

Tạo kết nối giữa các bộ xử lý Split1000LinesOfText và SplitText. Di chuột qua Split1000LinesOfText để xem biểu tượng mũi tên, nhấn vào bộ xử lý và kết nối nó với SplitText.

Cấu hình Tạo Kết Nối:

| **Connection** | **Value** |
| --- | --- |
| For Relationships | splits (**checked**) |

Click THÊM.



Nhấn giữ Control + nhấp chuột lên SplitText để cấu hình bộ xử lý:

Bảng 1: Tab Cài Đặt

| **Setting** | **Value** |
| --- | --- |
| Name | Split100LinesOfText |
| Automatically Terminate Relationships | failure (**checked**) |
| Automatically Terminate Relationships | original (**checked**) |

Bảng 2: Tab Lập Lịch

| **Scheduling** | **Value** |
| --- | --- |
| Scheduling Strategy | Event Driven |

Bảng 3: Tab Thuộc Tính

| **Property** | **Value** |
| --- | --- |
| **Line Split Count** | 100 |

Nhấp vào ÁP DỤNG.

**Chia tệp văn bản thành các đoạn, mỗi đoạn chứa 10 dòng.**

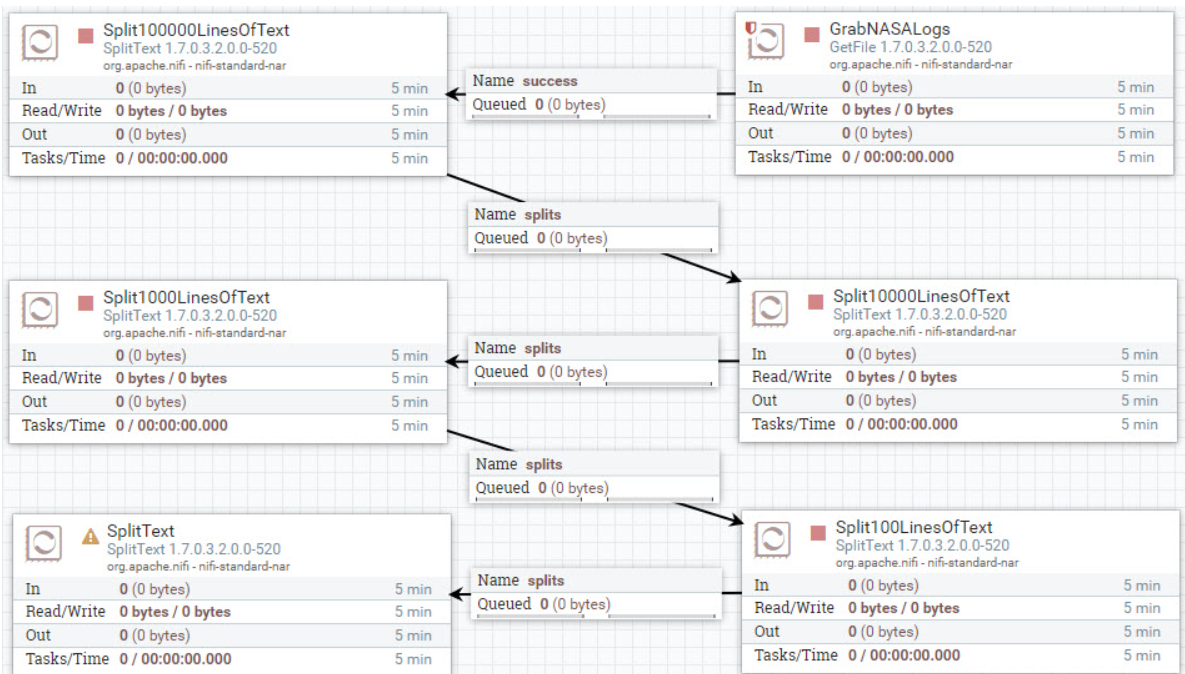
Kéo biểu tượng bộ xử lý và đặt nó lên bảng NiFi. Thêm bộ xử lý SplitText.

Tạo kết nối giữa các bộ xử lý Split100LinesOfText và SplitText. Di chuột qua Split100LinesOfText để xem biểu tượng mũi tên, nhấn vào bộ xử lý và kết nối nó với SplitText.

Cấu hình Tạo Kết Nối:

| **Connection** | **Value** |
| --- | --- |
| For Relationships | splits (**checked**) |

Click THÊM.



Nhấn giữ Control + nhấp chuột lên SplitText để cấu hình bộ xử lý:

Bảng 1: Tab Cài Đặt

| **Setting** | **Value** |
| --- | --- |
| Name | Split100LinesOfText |
| Automatically Terminate Relationships | failure (**checked**) |
| Automatically Terminate Relationships | original (**checked**) |

Bảng 2: Tab Lập Lịch

| **Scheduling** | **Value** |
| --- | --- |
| Scheduling Strategy | Event Driven |

Bảng 3: Tab Thuộc Tính

| **Property** | **Value** |
| --- | --- |
| **Line Split Count** | 100 |

Nhấp vào ÁP DỤNG.

Chia tệp văn bản thành các đoạn, mỗi đoạn chứa 1 dòng.

Kéo biểu tượng bộ xử lý và đặt nó lên bảng NiFi. Thêm bộ xử lý SplitText.

Tạo kết nối giữa các bộ xử lý Split10LinesOfText và SplitText. Di chuột qua Split10LinesOfText để xem biểu tượng mũi tên, nhấn vào bộ xử lý và kết nối nó với SplitText.

Cấu hình Tạo Kết Nối:

| **Connection** | **Value** |
| --- | --- |
| For Relationships | splits (**checked**) |

<https://github.com/xvanausloos/hdp_data_tutorials/blob/master/tutorials/cda/building-a-server-log-analysis-application/assets/images/acquire-nasa-log-data/splittext100_splittext10.jpg>

Nhấn giữ Control + nhấp chuột lên SplitText để cấu hình bộ xử lý:

Bảng 1: Tab Cài Đặt

Cài Đặt Giá Trị

Tên Split10LinesOfText

Tự động chấm dứt mối quan hệ failure (đã kiểm tra)

Tự động chấm dứt mối quan hệ original (đã kiểm tra)

Bảng 2: Tab Lập Lịch

Lịch Chạy Giá Trị

Ghi chú Chiến lược lập lịch này là thử nghiệm

Chiến lược lập lịch Điều Khiển Sự Kiện

Bảng 3: Tab Thuộc Tính

Thuộc Tính Giá Trị

Số Dòng Chia 10

Nhấp vào ÁP DỤNG.

Chia tệp văn bản thành các đoạn, mỗi đoạn chứa 1 dòng.

Kéo biểu tượng bộ xử lý và đặt nó lên bảng NiFi. Thêm bộ xử lý SplitText.

Tạo kết nối giữa các bộ xử lý Split10LinesOfText và SplitText. Di chuột qua Split10LinesOfText để xem biểu tượng mũi tên, nhấn vào bộ xử lý và kết nối nó với SplitText.

Cấu Hình Tạo Kết Nối:

Kết Nối Giá Trị

Đối Với Mối Quan Hệ splits (đã kiểm tra)

Nhấp vào THÊM.

<https://github.com/xvanausloos/hdp_data_tutorials/blob/master/tutorials/cda/building-a-server-log-analysis-application/assets/images/acquire-nasa-log-data/splittext10_splittext1.jpg>

Nhấn giữ Control + nhấp chuột lên SplitText để cấu hình bộ xử lý:

Bảng 1: Tab Cài Đặt

Cài Đặt Giá Trị

Tên Split1LinesOfText

Tự động chấm dứt mối quan hệ failure (đã kiểm tra)

Tự động chấm dứt mối quan hệ original (đã kiểm tra)

Bảng 2: Tab Lập Lịch

Lịch Chạy Giá Trị

Ghi chú Chiến lược lập lịch này là thử nghiệm

Chiến lược lập lịch Điều Khiển Sự Kiện

Bảng 3: Tab Thuộc Tính

Thuộc Tính Giá Trị

Số Dòng Chia 1

Nhấp vào ÁP DỤNG.

Rút trích Mẫu Regex từ Nội dung FlowFile thành Thuộc tính

Kéo biểu tượng bộ xử lý và đặt nó lên bảng NiFi. Thêm bộ xử lý ExtractText.

Tạo kết nối giữa các bộ xử lý Split1LinesOfText và ExtractText. Di chuột qua Split1LinesOfText để xem biểu tượng mũi tên, nhấn vào bộ xử lý và kết nối nó với ExtractText.

Cấu Hình Tạo Kết Nối:

Kết Nối Giá Trị

Đối Với Mối Quan Hệ splits (đã kiểm tra)

Nhấp vào THÊM.

<https://github.com/xvanausloos/hdp_data_tutorials/blob/master/tutorials/cda/building-a-server-log-analysis-application/assets/images/acquire-nasa-log-data/splittext1_to_extracttext.jpg>

Nhấn giữ Control + nhấp chuột lên ExtractText để cấu hình bộ xử lý:

Bảng 1: Tab Cài Đặt

Cài Đặt Giá Trị

Tự động chấm dứt mối quan hệ unmatched (đã kiểm tra)

Bảng 2: Tab Lập Lịch

Lịch Chạy Giá Trị

Ghi chú Chiến lược lập lịch này là thử nghiệm

Chiến lược lập lịch Điều Khiển Sự Kiện

Bảng 3: Tab Thuộc Tính

Thêm thuộc tính do người dùng xác định bằng cách nhấn vào nút +.

Thuộc Tính Giá Trị

IP (\b(?:\d{1,3}.){3}\d{1,3}\b)

Request\_Type "(.\*?)"

Response\_Code HTTP/\d.\d" (\d{3})

Time

(.∗?)

(.∗?)

Nhấp vào ÁP DỤNG.

Chuyển hướng FlowFiles Chứa Địa chỉ IP Hợp lệ

Kéo biểu tượng bộ xử lý và đặt nó lên bảng NiFi. Thêm bộ xử lý RouteOnAttribute.

Tạo kết nối giữa các bộ xử lý ExtractText và RouteOnAttribute. Di chuột qua ExtractText để xem biểu tượng mũi tên, nhấn vào bộ xử lý và kết nối nó với RouteOnAttribute.

Cấu Hình Tạo Kết Nối:

Kết Nối Giá Trị

Đối Với Mối Quan Hệ matched (đã kiểm tra)

Nhấp vào THÊM.

<https://github.com/xvanausloos/hdp_data_tutorials/blob/master/tutorials/cda/building-a-server-log-analysis-application/assets/images/acquire-nasa-log-data/routeonattribute_to_geoenrichip.jpg>

Nhấn giữ Control + nhấp chuột lên GeoEnrichIP để cấu hình bộ xử lý:

Bảng 1: Tab Cài Đặt

Cài Đặt Giá Trị

Tự động chấm dứt mối quan hệ not found (đã kiểm tra)

Bảng 2: Tab Lập Lịch

Lịch Chạy Giá Trị

Ghi chú Chiến lược lập lịch này là thử nghiệm

Chiến lược lập lịch Điều Khiển Sự Kiện

Bảng 3: Tab Thuộc Tính

Thêm thuộc tính do người dùng xác định bằng cách nhấn vào nút +.

Thuộc Tính Giá Trị

MaxMind Database File /sandbox/tutorial-files/200/nifi/input/GeoFile/GeoLite2-City\_<timestamp>/GeoLite2-City.mmdb

IP Address Attribute IP

Nhấp vào ÁP DỤNG.

Chuyển hướng FlowFiles Chứa Thông tin Thành Phố Không Rỗng

Kéo biểu tượng bộ xử lý và đặt nó lên bảng NiFi. Thêm bộ xử lý RouteOnAttribute.

Tạo kết nối giữa các bộ xử lý GeoEnrichIP và RouteOnAttribute. Di chuột qua GeoEnrichIP để xem biểu tượng mũi tên, nhấn vào bộ xử lý và kết nối nó với RouteOnAttribute.

Cấu Hình Tạo Kết Nối:

Kết Nối Giá Trị

Đối Với Mối Quan Hệ found (đã kiểm tra)

Nhấp vào THÊM.

<https://github.com/xvanausloos/hdp_data_tutorials/blob/master/tutorials/cda/building-a-server-log-analysis-application/assets/images/acquire-nasa-log-data/geoenrichip_to_routeonattribute.jpg>

Nhấn giữ Control + nhấp chuột lên RouteOnAttribute để cấu hình bộ xử lý:

Bảng 1: Tab Cài Đặt

Cài Đặt Giá Trị

Tự động chấm dứt mối quan hệ unmatched (đã kiểm tra)

Bảng 2: Tab Lập Lịch

Lịch Chạy Giá Trị

Ghi chú Chiến lược lập lịch này là thử nghiệm

Chiến lược lập lịch Điều Khiển Sự Kiện

Bảng 3: Tab Thuộc Tính

Thêm thuộc tính do người dùng xác định bằng cách nhấn vào nút +.

Thuộc Tính Giá Trị

Routing Strategy Route to 'matched' if all match

Check\_city ${IP.geo.city:isEmpty():not()}

Nhấp vào ÁP DỤNG.

Tìm kiếm và Thay Thế Nội Dung của FlowFile qua Regex

Kéo biểu tượng bộ xử lý và đặt nó lên bảng NiFi. Thêm bộ xử lý ReplaceText.

Tạo kết nối giữa các bộ xử lý RouteOnAttribute và ReplaceText. Di chuột qua RouteOnAttribute để xem biểu tượng mũi tên, nhấn vào bộ xử lý và kết nối nó với ReplaceText.

Cấu Hình Tạo Kết Nối:

Kết Nối Giá Trị

Đối Với Mối Quan Hệ matched (đã kiểm tra)

Nhấp vào THÊM.

<https://github.com/xvanausloos/hdp_data_tutorials/blob/master/tutorials/cda/building-a-server-log-analysis-application/assets/images/acquire-nasa-log-data/routeonattribute_to_replacetext.jpg>

Nhấn giữ Control + nhấp chuột lên ReplaceText để cấu hình bộ xử lý:

Bảng 1: Tab Cài Đặt

Cài Đặt Giá Trị

Tự động chấm dứt mối quan hệ failure (đã kiểm tra)

Bảng 2: Tab Lập Lịch

Lịch Chạy Giá Trị

Ghi chú Chiến lược lập lịch này là thử nghiệm

Chiến lược lập lịch Điều Khiển Sự Kiện

Bảng 3: Tab Thuộc Tính

Sao chép và dán các giá trị thuộc tính giữa các dấu ngoặc kép:

Thuộc Tính Giá Trị

Search Value "(?s)(^.\*$)"

Replacement Value "${IP}

Nhấp vào ÁP DỤNG.

Hợp nhất FlowFiles Khi Đã Tích Lũy Đủ Số Lượng

Kéo biểu tượng bộ xử lý và đặt nó lên bảng NiFi. Thêm bộ xử lý MergeContent.

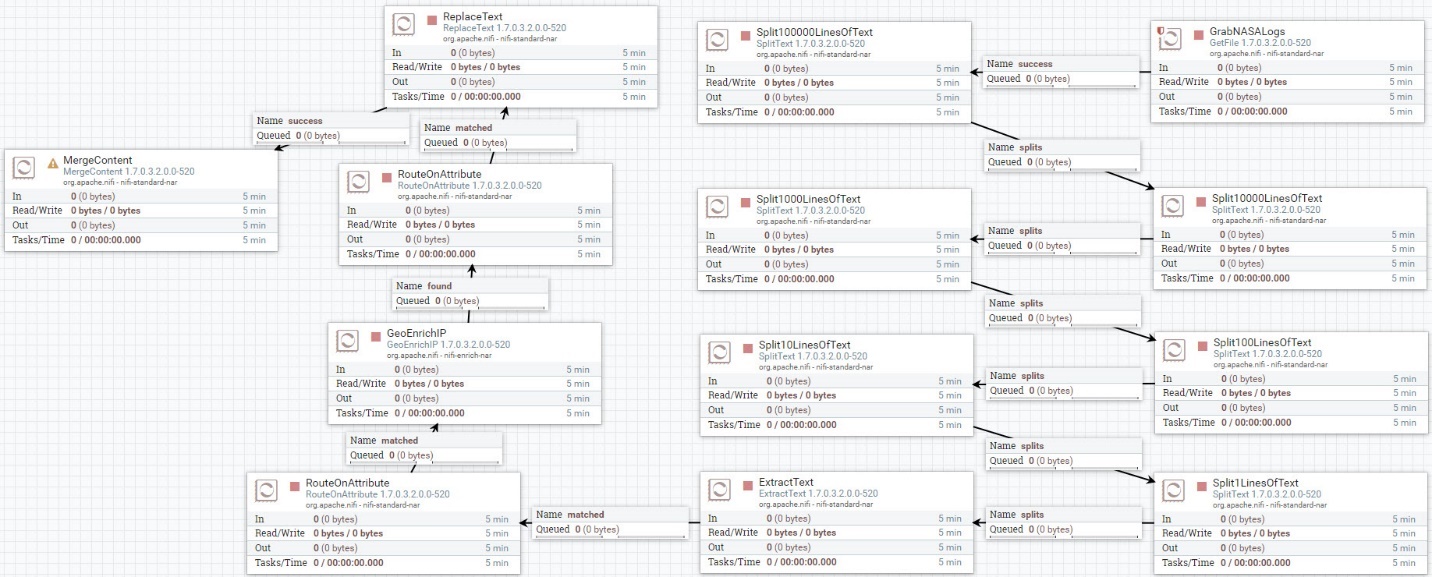
Tạo kết nối giữa các bộ xử lý ReplaceText và MergeContent. Di chuột qua ReplaceText để xem biểu tượng mũi tên, nhấn vào bộ xử lý và kết nối nó với MergeContent.

Cấu Hình Tạo Kết Nối:

Kết Nối Giá Trị

Đối Với Mối Quan Hệ success (đã kiểm tra)

Nhấp vào THÊM.



Nhấn giữ Control + nhấp chuột lên MergeContent để cấu hình bộ xử lý:

Bảng 1: Tab Cài Đặt

Cài Đặt Giá Trị

Tự động chấm dứt mối quan hệ failure (đã kiểm tra), original (đã kiểm tra)

Bảng 2: Tab Lập Lịch

Lịch Chạy Giá Trị

Ghi chú Chiến lược lập lịch này là thử nghiệm

Chiến lược lập lịch Điều Khiển Sự Kiện

Bảng 3: Tab Thuộc Tính

Ghi chú: nhấn shift + enter để xuống dòng cho demarcator

Thuộc Tính Giá Trị

Minimum Number of Entries 20

Maximum Number of Entries 40

Maximum Number of Bins 100

Delimiter Strategy Text

Demarcator nhấn {shift + enter}

Nhấp vào ÁP DỤNG.

Cập Nhật Tên Thuộc Tính của FlowFile Đã Hợp Nhất

Kéo biểu tượng bộ xử lý và đặt nó lên bảng NiFi. Thêm bộ xử lý UpdateAttribute.

Tạo kết nối giữa các bộ xử lý MergeContent và UpdateAttribute. Di chuột qua MergeContent để xem biểu tượng mũi tên, nhấn vào bộ xử lý và kết nối nó với UpdateAttribute.

Cấu Hình Tạo Kết Nối:

Kết Nối Giá Trị

Đối Với Mối Quan Hệ merged (đã kiểm tra)

Nhấp vào THÊM.

<https://github.com/xvanausloos/hdp_data_tutorials/blob/master/tutorials/cda/building-a-server-log-analysis-application/assets/images/acquire-nasa-log-data/mergecontent_to_updateattribute.jpg>

Nhấn giữ Control + nhấp chuột lên UpdateAttribute để cấu hình bộ xử lý:

Bảng 2: Tab Lập Lịch

Lịch Chạy Giá Trị

Ghi chú Chiến lược lập lịch này là thử nghiệm

Chiến lược lập lịch Điều Khiển Sự Kiện

Bảng 3: Tab Thuộc Tính

Thuộc Tính Giá Trị

filename logsample-${now():format("HHmmssSSS")}-${UUID()}-.txt

Nhấp vào ÁP DỤNG.

Ghi Nội Dung của FlowFile vào Hệ Thống Tệp Phân Tán Hadoop (HDFS)

Kéo biểu tượng bộ xử lý và đặt nó lên bảng NiFi. Thêm bộ xử lý PutHDFS.

Tạo kết nối giữa các bộ xử lý UpdateAttribute và PutHDFS. Di chuột qua UpdateAttribute để xem biểu tượng mũi tên, nhấn vào bộ xử lý và kết nối nó với PutHDFS.

Cấu Hình Tạo Kết Nối:

Kết Nối Giá Trị

Đối Với Mối Quan Hệ success (đã kiểm tra)

Nhấp vào THÊM.

<https://github.com/xvanausloos/hdp_data_tutorials/blob/master/tutorials/cda/building-a-server-log-analysis-application/assets/images/acquire-nasa-log-data/updateattribute_to_puthdfs.jpg>

Nhấn giữ Control + nhấp chuột lên PutHDFS để cấu hình bộ xử lý:

Bảng 1: Tab Thiết Lập

Thiết Lập Giá Trị

Tự động Chấm Dứt Mối Quan Hệ failure (đã kiểm tra)

Tự động Chấm Dứt Mối Quan Hệ success (đã kiểm tra)

Bảng 2: Tab Lập Lịch

Lập Lịch Giá Trị

Lịch Chạy 0 sec

Bảng 3: Tab Thuộc Tính

Thuộc Tính Giá Trị

Hadoop Configuration Resources /etc/hadoop/conf/core-site.xml,/etc/hadoop/conf/hdfs-site.xml

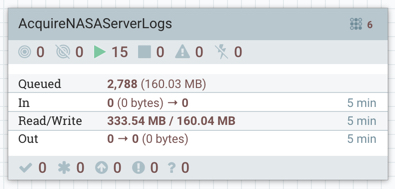
Thư mục /sandbox/tutorial-files/200/nifi/output/NASALogsAug1995

Nhấp vào ÁP DỤNG.

Biểu tượng hình nón màu vàng của bộ xử lý PutHDFS sẽ thay đổi thành biểu tượng biển cấm màu đỏ.

Bắt đầu Luồng Nhóm Tiến Trình để Thu Thập Dữ Liệu

Tại bước đường dẫn, chọn cấp độ NiFi Flow. Nhấn giữ Control + nhấp chuột lên tiến trình nhóm AcquireNASAServerLogs, sau đó chọn tùy chọn khởi động.



started\_acquirenasaserverlogs\_pg

Khi NiFi ghi dữ liệu máy chủ của bạn vào HDFS, bạn có thể kiểm tra bằng cách xem thông tin chứng minh dữ liệu, sau đó bạn có thể tắt nhóm tiến trình bằng cách nhấn giữ Control + nhấp chuột lên tiến trình nhóm AcquireNASAServerLogs, sau đó chọn tùy chọn dừng.

Cách Tiếp Cận 2: Nhập Dữ Liệu NiFi AcquireNASALogs qua Giao Diện Người Dùng

Tải mẫu NiFi AcquireNASAServerLogs.xml về máy tính cục bộ của bạn.

Sau khi bắt đầu sandbox của bạn, mở giao diện người dùng HDF NiFi tại http://sandbox-hdf.hortonworks.com:9090/nifi.

Mở bảng điều khiển Nếu chưa được mở, sau đó nhấn vào biểu tượng Tải Lên Mẫu tải lên.

<https://github.com/xvanausloos/hdp_data_tutorials/blob/master/tutorials/cda/building-a-server-log-analysis-application/assets/images/acquire-nasa-log-data/started_acquirenasaserverlogs_pg.jpg>

Nhấn vào biểu tượng Chọn Mẫu tìm kiếm.

Trình duyệt tệp trên máy tính cục bộ của bạn sẽ xuất hiện, tìm mẫu AcquireNASAServerLogs.xml bạn vừa tải về, sau đó nhấn Mở, sau đó nhấn Tải Lên.

Bạn sẽ nhận được thông báo rằng Mẫu đã được nhập thành công. Nhấn OK để xác nhận.

Kéo biểu tượng Mẫu vào bảng NiFi.

Thêm Mẫu có tên là AcquireNASAServerLogs.

Bắt đầu luồng NiFi. Nhấn giữ Control + nhấp chuột lên tiến trình nhóm AcquireNASAServerLogs, sau đó chọn tùy chọn khởi động.

started\_acquirenasaserverlogs\_pg

Khi NiFi ghi dữ liệu máy chủ của bạn vào HDFS, bạn có thể kiểm tra bằng cách xem thông tin chứng minh dữ liệu, sau đó bạn có thể tắt nhóm tiến trình bằng cách nhấn giữ Control + nhấp chuột lên tiến trình nhóm AcquireNASAServerLogs, sau đó chọn tùy chọn dừng.

Cách Tiếp Cận 3: Triển Khai Tự Động Luồng NiFi qua Cuộc Gọi REST

Mở HDF Sandbox Web Shell Client tại http://sandbox-hdf.hortonworks.com:4200. Sao chép và dán đoạn mã shell sau:

bash

Copy code

NIFI\_TEMPLATE="AcquireNASAServerLogs"

wget https://github.com/hortonworks/data-tutorials/raw/master/tutorials/cda/building-a-server-log-analysis-application/application/development/shell/nifi-auto-deploy.sh

bash nifi-auto-deploy.sh $NIFI\_TEMPLATE

Mở giao diện người dùng HDF NiFi tại http://sandbox-hdf.hortonworks.com:9090/nifi. NiFi của bạn vừa được tải lên, nhập và khởi động. Bạn chỉ cần đảm bảo bật bộ xử lý

Nhấn giữ Control + nhấp chuột lên PutHDFS để cấu hình bộ xử lý:

Bảng 1: Tab Thiết Lập

Thiết Lập Giá Trị

Tự động Chấm Dứt Mối Quan Hệ failure (đã kiểm tra)

Tự động Chấm Dứt Mối Quan Hệ success (đã kiểm tra)

Bảng 2: Tab Lập Lịch

Lập Lịch Giá Trị

Lịch Chạy 0 sec

Bảng 3: Tab Thuộc Tính

Thuộc Tính Giá Trị

Hadoop Configuration Resources /etc/hadoop/conf/core-site.xml,/etc/hadoop/conf/hdfs-site.xml

Thư mục /sandbox/tutorial-files/200/nifi/output/NASALogsAug1995

Nhấp vào ÁP DỤNG.

Biểu tượng hình nón màu vàng của bộ xử lý PutHDFS sẽ thay đổi thành biểu tượng biển cấm màu đỏ.

Bắt đầu Luồng Nhóm Tiến Trình để Thu Thập Dữ Liệu

Tại bước đường dẫn, chọn cấp độ NiFi Flow. Nhấn giữ Control + nhấp chuột lên tiến trình nhóm AcquireNASAServerLogs, sau đó chọn tùy chọn khởi động.

started\_acquirenasaserverlogs\_pg

Khi NiFi ghi dữ liệu máy chủ của bạn vào HDFS, bạn có thể kiểm tra bằng cách xem thông tin chứng minh dữ liệu, sau đó bạn có thể tắt nhóm tiến trình bằng cách nhấn giữ Control + nhấp chuột lên tiến trình nhóm AcquireNASAServerLogs, sau đó chọn tùy chọn dừng.

Cách Tiếp Cận 2: Nhập Dữ Liệu NiFi AcquireNASALogs qua Giao Diện Người Dùng

Tải mẫu NiFi AcquireNASAServerLogs.xml về máy tính cục bộ của bạn.

Sau khi bắt đầu sandbox của bạn, mở giao diện người dùng HDF NiFi tại http://sandbox-hdf.hortonworks.com:9090/nifi.

Mở bảng điều khiển Nếu chưa được mở, sau đó nhấn vào biểu tượng Tải Lên Mẫu tải lên.

Nhấn vào biểu tượng Chọn Mẫu tìm kiếm.

Trình duyệt tệp trên máy tính cục bộ của bạn sẽ xuất hiện, tìm mẫu AcquireNASAServerLogs.xml bạn vừa tải về, sau đó nhấn Mở, sau đó nhấn Tải Lên.

Bạn sẽ nhận được thông báo rằng Mẫu đã được nhập thành công. Nhấn OK để xác nhận.

Kéo biểu tượng Mẫu vào bảng NiFi.

Thêm Mẫu có tên là AcquireNASAServerLogs.

Bắt đầu luồng NiFi. Nhấn giữ Control + nhấp chuột lên tiến trình nhóm AcquireNASAServerLogs, sau đó chọn tùy chọn khởi động.

started\_acquirenasaserverlogs\_pg

Khi NiFi ghi dữ liệu máy chủ của bạn vào HDFS, bạn có thể kiểm tra bằng cách xem thông tin chứng minh dữ liệu, sau đó bạn có thể tắt nhóm tiến trình bằng cách nhấn giữ Control + nhấp chuột lên tiến trình nhóm AcquireNASAServerLogs, sau đó chọn tùy chọn dừng.

Cách Tiếp Cận 3: Triển Khai Tự Động Luồng NiFi qua Cuộc Gọi REST

Mở HDF Sandbox Web Shell Client tại http://sandbox-hdf.hortonworks.com:4200. Sao chép và dán đoạn mã shell sau:

bash

Copy code

NIFI\_TEMPLATE="AcquireNASAServerLogs"

wget https://github.com/hortonworks/data-tutorials/raw/master/tutorials/cda/building-a-server-log-analysis-application/application/development/shell/nifi-auto-deploy.sh

bash nifi-auto-deploy.sh $NIFI\_TEMPLATE

Mở giao diện người dùng HDF NiFi tại http://sandbox-hdf.hortonworks.com:9090/nifi. NiFi của bạn vừa được tải lên, nhập và khởi động. Bạn chỉ cần đảm bảo bật bộ xử lý PutHDFS, để NiFi có thể lưu trữ dữ liệu vào HDFS.

User

Cleaning the Raw NASA Log Data

Introduction

You have been brought onto the project as a Data Engineer with the following responsibilities: load in HDFS data into Spark DataFrame, analyze the various columns of the data to discover what needs cleansing, each time you hit checkpoints in cleaning up the data, you will register the DataFrame as a temporary table for later visualization in a different notebook and when the cleaning is complete, you will create a Hive table for ability to visualize the data with external visualization tools.

Prerequisites

Enabled CDA for your appropriate system.

Setup Development Environment

Acquired NASA Server Log Data

Outline

Approach 1: Clean Raw NASA Log Data with Spark Zeppelin Interpreter

Approach 2: Import Zeppelin Notebook to Clean NASA Log Data via UI

Summary

Further Reading

Approach 1: Clean Raw NASA Log Data with Spark Zeppelin Interpreter

Open HDP Zeppelin UI at http://sandbox-hdp.hortonworks.com:9995.

1. Click Create new note. Insert Note Name Cleaning-Raw-NASA-Log-Data, then press Create Note.

We are taken to the new Zeppelin Notebook.

ChatGPT

Clean Raw NASA Log Data with Spark Zeppelin Interpreter

Mở giao diện người dùng Zeppelin trên HDP tại http://sandbox-hdp.hortonworks.com:9995.

Nhấp vào Tạo ghi chú mới. Chèn Tên Ghi chú Làm-Sạch-Dữ liệu-Log-NASA-Gốc, sau đó nhấn Tạo ghi chú.

Chúng ta sẽ được chuyển đến sổ tay Zeppelin mới.

Implementation Steps:

Chèn mã sau vào ô lệnh Zeppelin:

scala

Copy code

import org.apache.spark.sql.SparkSession

// Load the raw data into a Spark DataFrame

val rawLogData = spark.read.text("/sandbox/tutorial-files/200/nifi/output/NASALogsAug1995")

// Display the first few rows of the DataFrame

rawLogData.show(5)

Nhấp vào biểu tượng Play (▶️) để chạy mã. Kết quả sẽ hiển thị một số hàng đầu tiên của DataFrame.

Tiếp theo, thực hiện kiểm tra cột để xác định những cột cần được làm sạch:

scala

Copy code

// Display the schema of the DataFrame

rawLogData.printSchema()

Nhấp vào biểu tượng Play (▶️) để chạy mã. Kết quả sẽ hiển thị schema của DataFrame.

Dựa vào thông tin từ schema, quyết định những cột cần được làm sạch và thực hiện các bước làm sạch cụ thể tương ứng với mỗi cột.

Mỗi khi bạn đạt đến một điểm kiểm tra trong việc làm sạch dữ liệu, đăng ký DataFrame như một bảng tạm thời để sau này có thể xem trực quan trong một sổ tay khác.

scala

Copy code

// Register the DataFrame as a temporary table

rawLogData.createOrReplaceTempView("temp\_log\_data")

Nhấp vào biểu tượng Play (▶️) để chạy mã và đăng ký DataFrame.

Tiếp tục làm sạch dữ liệu cho đến khi hoàn thành.

Khi làm sạch dữ liệu hoàn tất, tạo một bảng Hive để có khả năng xem dữ liệu bằng các công cụ trực quan hóa bên ngoài.

scala

Copy code

// Create a Hive table from the cleaned DataFrame

spark.sql("CREATE TABLE cleaned\_nasa\_logs AS SELECT \* FROM temp\_log\_data")

Nhấp vào biểu tượng Play (▶️) để chạy mã và tạo bảng Hive từ DataFrame đã làm sạch.

Kết thúc quá trình làm sạch dữ liệu.

Cách Tiếp Cận 2: Nhập Sổ Tay Zeppelin để Làm Sạch Dữ Liệu NASA Log qua Giao Diện Người Dùng

Tải sổ tay Zeppelin để làm sạch Dữ liệu Log NASA tại đây: Cleaning-Raw-NASA-Log-Data.json

Sau khi tải xong, bạn có thể nhúng sổ tay vào Zeppelin UI bằng cách tải lên tệp JSON.

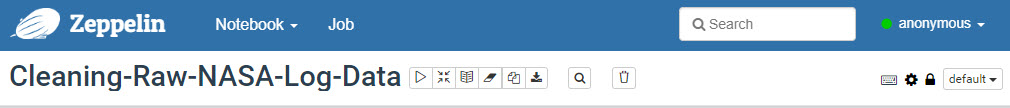
Tóm Tắt

Chúc mừng! Bây giờ bạn đã biết cách làm sạch dữ liệu log NASA sử dụng Zeppelin trên HDP. Bạn đã tải dữ liệu vào DataFrame của Spark, phân tích cột và thực hiện các bước làm sạch. DataFrame đã được đăng ký làm bảng tạm thời và sau đó tạo một bảng Hive để có thể xem dữ liệu bằng các công cụ trực quan hóa bên ngoài.

Đọc Thêm

Hướng Dẫn Người Dùng Zeppelin

API REST Zeppelin



Tải Thư Viện Bên Ngoài

Chúng ta sẽ sử dụng lệnh %dep của Zeppelin để nhập thư viện bên ngoài com.databricks:spark-csv\_2.11:1.4.0. Hãy sao chép và dán đoạn mã sau vào ghi chú Zeppelin:

scala

Copy code

%dep

z.reset()

z.load("com.databricks:spark-csv\_2.11:1.4.0")

Nhấn {shift + enter} hoặc biểu tượng Play (▶️) để biên dịch ghi chú trong sổ tay Zeppelin.

<https://github.com/xvanausloos/hdp_data_tutorials/blob/master/tutorials/cda/building-a-server-log-analysis-application/assets/images/cleaning-raw-nasa-log-data/loading_external_lib.jpg>

Loading the DataFrame from HDFS

We will load the dataframe from HDFS directory /sandbox/tutorial-files/200/nifi/output/NASALogsAug1995 using PySpark's sqlContext.read.format() function. Then we will use .show() function to display the content of the dataframe. Copy and paste the following pyspark code into the next available note in the notebook:

|  |
| --- |
| %pyspark  from pyspark.sql.types import StructType, StructField, DoubleType, StringType  schema = StructType([  # Represents a field in a StructType  StructField("IP", StringType()),  StructField("Time", StringType()),  StructField("Request\_Type", StringType()),  StructField("Response\_Code",StringType()),  StructField("City", StringType()),  StructField("Country", StringType()),  StructField("Isocode", StringType()),  StructField("Latitude", StringType()),  StructField("Longitude", StringType())  ])  logs\_df = sqlContext.read \  .format("com.databricks.spark.csv") \  .schema(schema) \  .option("header", "false") \  .option("delimiter", "|") \  .load("/sandbox/tutorial-files/200/nifi/output/NASALogsAug1995")  logs\_df.show(truncate=False) |

Khi chúng ta kiểm thử bản demo, có 20 hàng được hiển thị.

<https://github.com/xvanausloos/hdp_data_tutorials/blob/master/tutorials/cda/building-a-server-log-analysis-application/assets/images/cleaning-raw-nasa-log-data/loading_dataframe_from_hdfs.jpg>

Phân Tích Thời Gian (Timestamp) từ Cột Thời Gian (Time)

Chúng ta sẽ phân tích thời gian từ cột "Time" và loại bỏ cột cũ để thay thế bằng cột mới "Timestamp".

|  |
| --- |
| %pyspark  from pyspark.sql.functions import udf  months = {  'Jan': 1, 'Feb': 2, 'Mar':3, 'Apr':4, 'May':5, 'Jun':6, 'Jul':7, 'Aug':8, 'Sep': 9, 'Oct':10, 'Nov': 11, 'Dec': 12  }  def parse\_timestamp(time):  """ This function takes a Time string parameter of logs\_df dataframe  Returns a string suitable for passing to CAST('timestamp') in the format YYYY-MM-DD hh:mm:ss  """  return "{0:04d}-{1:02d}-{2:02d} {3:02d}:{4:02d}:{5:02d}".format(  int(time[7:11]),  months[time[3:6]],  int(time[0:2]),  int(time[12:14]),  int(time[15:17]),  int(time[18:20])  )  udf\_parse\_timestamp = udf(parse\_timestamp)  # Assigning the Timestamp name to the new column and dropping the old Time column  parsed\_df = logs\_df.select('\*',  udf\_parse\_timestamp(logs\_df['Time'])  .cast('timestamp')  .alias('Timestamp')).drop('Time')  # Stores the dataframe in cache for the future use  parsed\_df.cache()  # Displays the results  parsed\_df.show() |

<https://github.com/xvanausloos/hdp_data_tutorials/blob/master/tutorials/cda/building-a-server-log-analysis-application/assets/images/cleaning-raw-nasa-log-data/parsing_timestamp.jpg>

Làm Sạch Cột Request\_Type

Hiện tại, cột Request\_Type chứa các từ như GET và HTTP/1.0 bao quanh dữ liệu thực sự được yêu cầu. Chúng ta sẽ loại bỏ hai từ này khỏi mỗi dòng.

python

Copy code

%pyspark

from pyspark.sql.functions import split, regexp\_extract

# Tạo một cột mới là 'Request\_Path' bằng cách loại bỏ các từ GET và HTTP/1.0 từ cột 'Request\_Type'

path\_df = parsed\_df.select('\*', regexp\_extract('Request\_Type', r'^.\*\w+\s+([^\s]+)\s+HTTP.\*', 1)

.alias('Request\_Path')).drop('Request\_Type')

# Cache the dataframe

path\_df.cache()

# Hiển thị kết quả

path\_df.show(truncate=False)

Lọc Các Host Thường Xuyên Truy Cập Server NASA Nhất

Chúng ta muốn lọc ra các máy chủ (host) thường xuyên truy cập máy chủ của NASA và sau đó lưu dữ liệu vào một bảng tạm thời.

python

Copy code

%pyspark

# Nhóm dữ liệu theo cột 'IP' và sau đó đếm

most\_frequent\_hosts = parsed\_df.groupBy("IP").count()

# Hiển thị kết quả

most\_frequent\_hosts.show()

# Đăng ký biến 'most\_frequent\_hosts' như một bảng tạm thời

most\_frequent\_hosts.registerTempTable("most\_frequent\_hosts")

Lọc Đếm Cho Mỗi Mã Phản Hồi

Mục tiêu của chúng ta là tìm ra số lần mà mỗi mã phản hồi đã xuất hiện và lưu kết quả vào một bảng tạm thời để sử dụng sau này.

python

Copy code

%pyspark

# Nhóm dữ liệu theo cột 'Response\_Code' và sau đó đếm

status\_count = path\_df.groupBy('Response\_Code').count()

# Hiển thị kết quả

status\_count.show()

# Đăng ký biến 'status\_count' như một bảng tạm thời

status\_count.registerTempTable("status\_count")

tutorial-05

Trực Quan Hóa Dữ Liệu Log Server NASA

Tiếp Cận 1: Trực Quan Hóa Dữ Liệu Log NASA với Trình Biên Dịch Spark Zeppelin

Mở HDP Zeppelin UI tại http://sandbox-hdp.hortonworks.com:9995.

Nhấp vào Tạo ghi chú mới. Chèn Tên ghi chú Visualizing-NASA-Log-Data, sau đó nhấn Tạo Ghi chú.

Chúng ta được chuyển đến Trình Biên Dịch Zeppelin mới.

Trực Quan Hóa Các Host Thường Xuyên Truy Cập Server NASA Thông Qua Biểu Đồ Bánh

Chúng ta sẽ hiển thị số lượng cho mỗi máy chủ thường xuyên truy cập máy chủ của NASA. Dữ liệu này sẽ minh họa ai là những máy chủ thường xuyên kết nối đến máy chủ của NASA.

Chạy mã SQL sau:

sql

Copy code

%sql

SELECT \* FROM most\_frequent\_hosts

ORDER BY count

DESC LIMIT 20

Sau đó chọn biểu đồ bánh.

Trực Quan Hóa Sự Xuất Hiện Của Mã Phản Hồi Thông Qua Biểu Đồ Cột

Biểu đồ cột sẽ tạo ra một hình ảnh rõ ràng về mã phản hồi nào xuất hiện thường xuyên nhất.

Chạy mã SQL sau:

sql

Copy code

%sql

SELECT \* FROM status\_count

ORDER BY Response\_Code

Sau đó chọn biểu đồ cột.

Trực Quan Hóa Các Bản Ghi Với Mã Phản Hồi 200

Chúng ta sẽ trực quan hóa các bản ghi với mã phản hồi 200, tức là các bản ghi có thành công mỗi giờ.

Chạy mã SQL sau:

sql

Copy code

%sql

SELECT \* FROM success\_logs\_by\_hours\_df

ORDER BY Hour

Sau đó chọn biểu đồ cột.

Trực Quan Hóa Các Phần Mở Rộng

Mục tiêu của chúng ta là tìm số lượng phần mở rộng khác nhau có sẵn trong bộ dữ liệu của chúng ta. Chúng ta sẽ nhóm cột và sau đó đếm số lượng bản ghi trong mỗi nhóm.

Chạy mã SQL sau:

sql

Copy code

%sql

SELECT \* FROM extension\_df\_count

ORDER BY count DESC

Sau đó chọn biểu đồ cột.

Trực Quan Hóa Địa Điểm Giao Thông Mạng

Chúng ta sẽ xem xét lưu lượng mạng cho mỗi quốc gia và cho các thành phố trong Hoa Kỳ.

Lưu Lượng Mạng Theo Quốc Gia

Chạy mã SQL sau:

sql

Copy code

%sql

SELECT country, count(country) AS count FROM path\_df GROUP BY country ORDER BY count

Sau đó chọn biểu đồ cột.

https://github.com/xvanausloos/hdp\_data\_tutorials/blob/master/tutorials/cda/building-a-server-log-analysis-application/assets/images/visualizing-nasa-log-data/most\_frequent\_hosts\_pie\_chart.jpg

Lưu Lượng Mạng Theo Thành Phố Của Một Quốc Gia

Chạy mã SQL sau:

sql

Copy code

%sql

SELECT city, count(city) AS count FROM path\_df WHERE country='United States' GROUP BY city ORDER BY count

<https://github.com/xvanausloos/hdp_data_tutorials/blob/master/tutorials/cda/building-a-server-log-analysis-application/assets/images/visualizing-nasa-log-data/net_traffic_loc_city.jpg>

Tiếp Cận 2: Nhập Ghi Chú Zeppelin Để Trực Quan Hóa Dữ Liệu Log NASA Qua Giao Diện Người Dùng

Mở HDP Zeppelin UI tại http://sandbox-hdp.hortonworks.com:9995.

<https://github.com/xvanausloos/hdp_data_tutorials/blob/master/tutorials/cda/building-a-server-log-analysis-application/assets/images/visualizing-nasa-log-data/response_code_bar_chart.jpg>

Nhấp vào Nhập ghi chú. Chọn Thêm từ URL.

Chèn URL sau vì chúng ta sẽ nhập ghi chú Visualizing-NASA-Log-Data:

https://raw.githubusercontent.com/hortonworks/data-tutorials/master/tutorials/cda/building-a-server-log-analysis-application/application/development/zeppelin-notebook/Visualizing-NASA-Log-Data.json

Nhấp vào Nhập ghi chú.

Ghi chú của bạn Visualizing-NASA-Log-Data nên là một phần của danh sách ghi chú bây giờ.

<https://github.com/xvanausloos/hdp_data_tutorials/blob/master/tutorials/cda/building-a-server-log-analysis-application/assets/images/visualizing-nasa-log-data/visualizing-nasa-log-data-added-list.jpg>

Nhấp vào ghi chú Visualizing-NASA-Log-Data. Sau đó nhấn nút play cho tất cả các đoạn văn được thực hiện. Nút play nằm gần tiêu đề của ghi chú này ở đầu trang trang web.

Kết luận

Chúc mừng! Bạn vừa trực quan hóa các khía cạnh quan trọng của dữ liệu từ các log server của NASA, chẳng hạn như số lần máy chủ cụ thể nào đó đánh vào máy chủ, các mã phản hồi chỉ ra tỷ lệ thành công của truyền dữ liệu và nơi lưu lượng mạng nặng nhất.

Đọc Thêm

Trực Quan Hóa Dữ Liệu