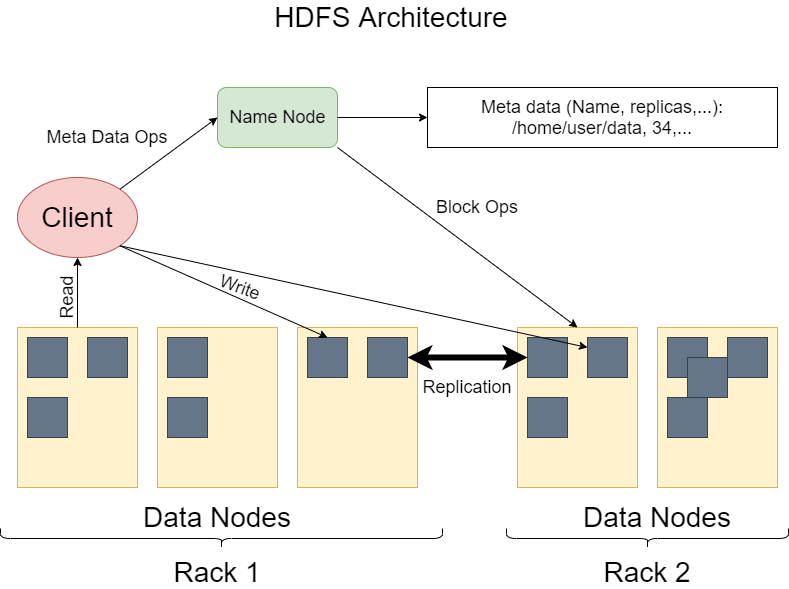
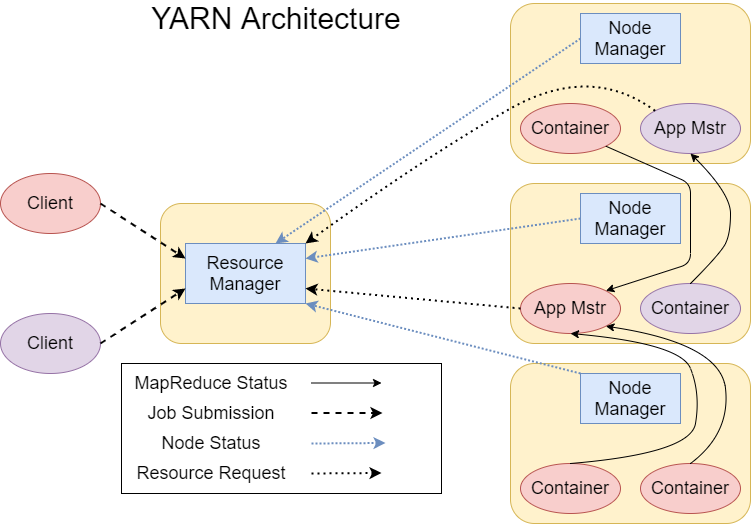
# **Kiến trúc của HDFS**



* Kiến trúc của HDFS là master/slave, một HDFS Cluster sẽ luôn bao gồm 1 NameNode, NameNode này là một master server và nó có nhiệm vụ quản lý cho hệ thống tập tin và điều chỉnh các truy cập đến những tập tin khác. Từ đó, bổ sung cho NameNode để có nhiều DataNodes. Sẽ luôn có 1 DataNode dành riêng cho các máy chủ dữ liệu. Trong một HDFS, 1 tập tin lớn sẽ được chia thành 1 hoặc nhiều khối, những khối này đều sẽ được lưu trữ trong 1 tập các DataNodes
* NameNode có tác vụ chính là đóng, mở và đổi tên cho các tập tin, thư mục. Ngoài ra, nó còn điều chỉnh cho các truy cập đến hệ thống tập tin.
* DataNode có tác vụ là ghi, đọc vào hệ thống tập tin, nó còn có nhiệm vụ tạo, xoá, nhân rộng các dữ liệu dựa trên chỉ dẫn của NameNode.
* Quá trình hoạt động của NameNode và DataNode:
  + NameNode: Có trách nhiệm điều phối cho các thao tác truy cập của client với hệ thống HDFS. Bởi vì các DataNode là nơi lưu trữ thật sự các block của các file trên HDFS nên chúng là nơi đáp ứng các truy cập này. NameNode sẽ thực hiện nhiệm vụ của nó thông qua daemon tên namenode chạy trên port 8021.
  + DataNode: DataNode server sẽ chạy một daemon datanode trên port 8022, theo định kỳ thì mỗi DataNode sẽ có nhiệm vụ báo cáo cho Namenode biết được danh sách tất cả các block mà nó đang lưu trữ. Để NameNode có thể dựa vào nó để cập nhật lại các metadata trong nó.

# **Kiến trúc của YARN**



* Trong cấu trúc của YARN, có 2 thành phần riêng biệt đảm nhiệm chức năng quản lý tài nguyên và quản lý job:
  + Resource Manager (RM): quản lý toàn bộ tài nguyên tính toán của cluster.
  + Application Master (AM): đơn vị nằm trên 1 ứng dụng quản lý vòng đời của Job.
* Các node tính toán trong cluster sẽ chạy NodeManager quản lý các tiến trình chạy trên máy đó. Resource Manager và Node Manager trở thành xương sống của tính toán phân tán trong YARN. Việc mỗi ứng dụng được tách ra riêng cho phép các tiến trình chạy lâu (long running process) cũng có thể được khởi động trên YARN.
* Application Master trên 1 ứng dụng là 1 thư viện cho phép yêu cầu tài nguyên từ Resource Manager và giao tiếp với Node Manager để chạy và thực thi các tasks. Application Master cho phép xây dựng các ứng dụng khác MR chạy trên YARN.
* Quy trình 1 ứng dụng chạy trên YARN:
  + Client giao 1 task cho Resource Manager
  + Resource Manager tính toán tài nguyên cần thiết theo yêu cầu của ứng dụng và tạo 1 Application Master (App Mstr). App Mstr được chuyển đến chạy 1 node tính toán. App Mstr sẽ liên lạc với các Node Manager ở các node khác để ra yêu cầu công việc cho node này.
  + Node Manager nhận yêu cầu và chạy các task trên container.
  + Các thông tin trạng thái sẽ được gửi đến Application Master.
* Các ứng dụng khác nhau sẽ có các Application Master khác nhau và sẽ được khởi động ở các máy trạm khác nhau. Khi một máy trạm bị lỗi phần cứng và sập, các ứng dụng khác không bị ảnh hưởng, chỉ có App Mstr chạy trên máy đó bị ảnh hưởng.

# **Các lệnh cơ bản thao tác với HDFS**

* **appendToFile**
  + Cú pháp: hadoop fs -appendToFile <localsrc> ... <dst>
  + Nối 1 hoặc nhiều file từ server vào file đích.
* **cat**
  + Cú pháp: hadoop fs -cat [-ignoreCrc] URI [URI ...]
  + Hiển thị nội dung file.
* **count**
  + Cú pháp: hadoop fs -count [-q] [-h] [-v] [-x] [-t [<storage type>]] [-u] [-e] [-s] <paths>
  + Đếm số thư mục, file.
* **ls**
  + Cú pháp: hadoop fs -ls [-C] [-d] [-h] [-q] [-R] [-t] [-S] [-r] [-u] [-e] <args>
  + Đối với file sẽ trả ra kết quả theo dạng: permissions number\_of\_replicas userid groupid filesize modification\_date modification\_time filename
  + Đối với folder sẽ trả ra kết quả theo dạng: permissions userid groupid modification\_date modification\_time dirname
* **mkdir**
  + Cú pháp: hadoop fs -mkdir [-p] <paths>
  + Tạo 1 thư mục như đường dẫn, -p tạo thêm cả thư mục cha trên đường dẫn nếu có.
* **mv**
  + Cú pháp: hadoop fs -mv URI [URI ...] <dest>
  + Chuyển file/folder từ nguồn đến địa chỉ đích.
* **put**
  + Cú pháp: hadoop fs -put [-f] [-p] [-l] [-d] [-t <thread count>] [ - | <localsrc1> .. ]. <dst>
  + Copy 1 hoặc nhiều file từ server đến đích.
* **rm**
  + Cú pháp: hadoop fs -rm [-f] [-r |-R] [-skipTrash] [-safely] URI [URI ...]
  + Xoá file hoặc thư mục ở đường dẫn.
* **tail**
  + Cú pháp: hadoop fs -tail [-f] URI
  + Hiển thị kilobyte cuối cùng của file.
* **get**
  + Cú pháp: hadoop fs -get [-ignorecrc] [-crc] [-p] [-f] <src> <localdst>
  + Copy file/folder về server.
* **help**
  + Cú pháp: hadoop fs -help
  + Danh sách các lệnh sử dụng trong hdfs.
* **dfsadmin**
  + Cú pháp: hdfs dfsadmin
  + Đưa ra thông tin hệ thống.
* **start-all.sh**
  + Chạy tất cả hệ thống.
* **stop-all.sh**
  + Dừng cả hệ thống.