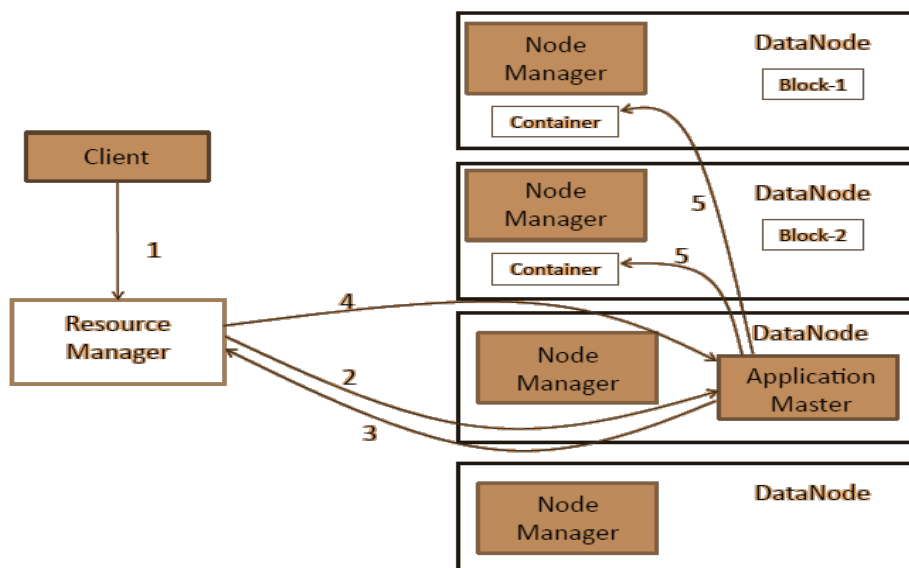


YARN

- YARN = Yet Another Resource Negotiator ou YARN Application Resource Negotiator
- La motivation initiale de YARN est d'améliorer encore l'évolutivité, la fiabilité et l'utilisation des clusters.
- L'idée de base de YARN est de diviser les deux principales fonctionnalités du **JobTracker** (gestion des ressources et planification / surveillance des Jobs) en processus démon séparés.
- On distingue donc un **ResourceManager** (RM) global et un **ApplicationMaster** (AM) pour chaque application qui s'exécute dans le cluster
 - ✓ ex: MapReduce Job ou Oozie Job-Netz.
- De plus, il existe désormais un **NodeManager** (NM) sur chaque nœud worker, qui surveille la consommation de ressources sur le nœud.
 - ✓ Les ressources sont divisées en conteneurs.
 - ✓ MapReduce n'est qu'une application du YARN
- Un avantage de YARN est que différentes applications dans différentes versions peuvent fonctionner en parallèle sur le même cluster.
 - ✓ Pour les développeurs d'applications, rien ne change lors du passage de MRv1 à MRv2: le modèle de programmation MapReduce et son API restent inchangés

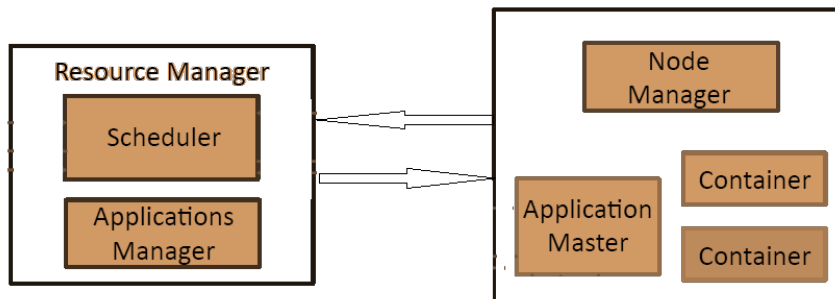
- Une application est exécutée dans le cluster selon les étapes suivantes:



1. Le **client** envoie la demande au **ResourceManager**.
2. **ResourceManager** lance l'**ApplicationMaster**
3. L'**ApplicationMaster** négocie les conteneurs requis avec **ResourceManager**
4. Allocation des conteneurs nécessaires à l'exécution des applications
5. L'**ApplicationMaster** contacte le **NodeManager** responsable pour démarrer le conteneur.

Le ResourceManager

- Le **ResourceManager** est un planificateur chargé de la répartition des ressources disponibles entre les applications concurrentes et d'optimiser l'utilisation du cluster. Il comporte deux composants principaux: **Scheduler** et **ApplicationsManager**



Le Scheduler

- S'occupe de la distribution de ressources des conteneurs aux différentes applications exécutées dans le cluster.
- L'allocation des ressources nécessaires à l'exécution des applications est basée sur les informations fournies par les applicationsMaster.

L'ApplicationManager

- Identifié l'ApplicationMaster.
- Accepte les demandes de Job et négocie le premier conteneur sur lequel l'ApplicationMaster peut s'exécuter.
- Surveille, démarre et, le cas échéant, redémarre l'ApplicationsMaster sur d'autres nœuds en cas d'erreurs.

L'ApplicationMaster

Il existe un ApplicationMaster (AM) pour chaque application démarrée dans le cluster (par exemple, MapReduce Job, Oozie jobNetz).

- Négocie avec le ResourceManager des conteneurs de ressources, suivi l'état et surveille la progression des conteneurs affectés à l'exécution de l'application.
- Est également responsable de la division d'une application en plusieurs tâches.
- Collabore avec les NodeManagers, dès que les ressources sont allouées, pour exécuter et surveiller les tâches individuelles.

Le NodeManager

S'exécute sur chaque nœud du cluster.

- Reçoit ses instructions du gestionnaire de ressources
- Est responsable des conteneurs présents sur le nœud, surveille l'utilisation des ressources (CPU, mémoire, etc.) et rapporte ces informations au ResourceManager.

