Configuring Intelligent Data Lake on Amazon Elastic MapReduce 5.0

© Copyright Informatica LLC 2017. Informatica, the Informatica logo, and Intelligent Data Lake are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at https://www.informatica.com/trademarks.html

Abstract

You can configure Intelligent Data Lake on Amazon Elastic MapReduce (EMR) 5.0. This article lists the steps required for this configuration.

Supported Versions

- Intelligent Data Lake 10.1.1

Table of Contents

Overview

You can deploy Intelligent Data Lake on Amazon AWS using Amazon EMR.

This article is intended for Intelligent Data Lake administrators who are responsible for installing Intelligent Data Lake on Amazon EMR. You must have knowledge of various Amazon Web Services components such as Amazon EMR, RDS, and EC2. You must also have knowledge of deployment practices for Informatica Big Data Management, Informatica Enterprise Information Catalog, and Intelligent Data Lake. For more information about Amazon Web Services, see the Amazon product documentation. For more information about Informatica products, see the appropriate product documentation.

To install and configure Intelligent Data Lake on Amazon EMR, perform the following steps:

1. Set up the Informatica domain in an Amazon AWS environment.
2. Set up the MySQL database.
3. Install and Configure Big Data Management for Amazon EMR.
4. Configure the Data Integration Service.
5. Configure Live Data Map.
6. Configure Intelligent Data Lake.

For more information about installing and configuring Big Data Management for Amazon EMR, see the *Informatica 10.1.1 Update 2 Big Data Management Installation and Configuration Guide*. For more information about installing and configuring Live Data Map, see the *Informatica 10.1.1 Update 2 Enterprise Information Catalog Installation and Configuration Guide*.

Limitations and Constraints

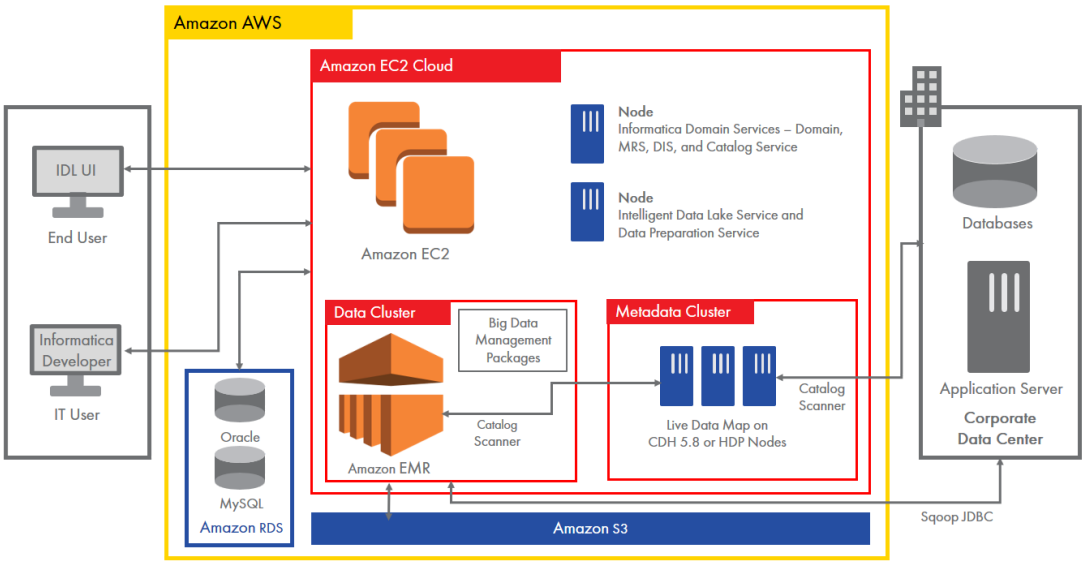
Note the following constraints while configuring Intelligent Data Lake on Amazon EMR.

- The Catalog Service can be deployed on an embedded cluster or an external supported cluster in Cloudera CDH or Hortonworks HDP.
- There is no support for ephemeral clusters.
- Intelligent Data Lake accesses the data on Amazon S3 through Hive by directing the Hive warehouse to Amazon S3. You cannot manage data outside the Hive warehouse.
- Amazon recommends that you use `S3://` URIs to access S3 files from Amazon EMR.
- The Data Preparation Service durable storage can either be pointed to the Catalog cluster's HDFS or the EMR cluster's HDFS.
- If you need audit functionality, the Audit Service will use EMR cluster libraries. The Audit Service will point to HBase on Amazon EMR only.

Intelligent Data Lake on Amazon EMR Architecture

Deploy Intelligent Data Lake in the AWS public cloud leveraging Amazon EMR and other leading Hadoop distributions such as Cloudera CDH and Hortonworks HDP. Host the Informatica domain in the cloud. In this mode of deployment, you install the Informatica domain, Big Data Management binaries, and Intelligent Data Lake binaries on an Amazon EMR cluster.

The following figure shows the deployment architecture for configuring Intelligent Data Lake on Amazon EMR:



Step 1. Set up the Domain

Set up the domain and configure the Catalog Service.

1. Set up a domain where Live Data Map is running on CDH 5.8 cluster on Amazon Elastic Cloud Compute (EC2) with the following configurations:
   - One node for the Informatica service (m4.2xsmall, 8 CPU core, 32GB RAM) - A small deployment with 32 GB memory, 8 CPU cores and one node.
   - Three nodes for Cloudera CDH cluster (3 x m4.2xlarge, total 24 CPU core, 96GB RAM) - A large deployment with 96 GB memory, 24 CPU cores, and three nodes.

   Note: You can set up Live Data Map either on an internal Hortonworks HDP cluster or on an external Cloudera CDH or Hortonworks HDP cluster.
2. Deploy Oracle on Amazon RDS.
3. Configure the Catalog Service to use CDH for running HBase, SOLR, and Spark.

   For more information about configuring the Catalog Service, see the *Informatica 10.1.1 Update 2 Enterprise Information Catalog Installation and Configuration Guide*.
4. Create a zip archive file **emr_5.0.zip** of all the files in this location:

   `<INFA_HOME>/services/shared/hadoop/amazon_emr5.0.0/lib`
5. Copy the **emr_5.0.zip** file to the following location:

   `<INFA_HOME>/services/CatalogService/ScannerBinaries`
6. Open the `<INFA_HOME>/services/CatalogService/ScannerBinaries/CustomDeployer/scannerDeployer.xml` file and add the path to the **emr_5.0.zip** file.
7. Copy the following files to Amazon S3 for installation:
   - Informatica installer
   - RPM installer
   - Informatica Hadoop installer

Step 2. Set up the Database

Set up the database for Amazon EMR configuration.

1. Create a MySQL instance in AWS RDS using the following commands:

```
aws rds create-db-instance
--db-name emr_hive
--db-instance-identifier emr-hive
--allocated-storage 5
--db-instance-class db.t2.micro
--engine mysql
--master-username hive
--master-user-password ******
--vpc-security-group-ids sg-c5d9e4a3
--availability-zone us-west-2b
--db-subnet-group-name default
--db-parameter-group-name default.mysql5.6
--no-multi-az
--engine-version 5.6.27
--license-model general-public-license
--option-group-name default:mysql-5-6
```

2. Retrieve the MySQL database instance hostname and port number by entering the following commands:

```
HIVE_DB_HOST=`aws rds describe-db-instances --db-instance-identifier emr-hive | grep
Address | awk -F":" '{print($2)}' | tr -d \ \"\,`
HIVE_DB_PORT=`aws rds describe-db-instances --db-instance-identifier emr-hive | grep
\"Port\" | awk -F":" '{print($2)}' | tr -d \ \"\,`
```

Step 3. Install and Configure Big Data Management for Amazon EMR

Create a Hadoop cluster. Then, install Big Data Management and configure Big Data Management for Amazon EMR.

1. Install Big Data Management for Amazon EMR.

   For more information, see the Installation for Amazon EMR chapter in the Informatica 10.1.1 Update 2 Big Data Management Installation and Configuration Guide.

2. Configure Big Data Management for Amazon EMR.

   For more information, see the Configuration for Amazon EMR chapter in the Informatica 10.1.1 Update 2 Big Data Management Installation and Configuration Guide.

3. Start the cluster and add a startup action to run the bootstrap script. The script copies the Informatica binaries on all nodes. The command line output appears as follows:

```
aws emr create-cluster
--termination-protected
--applications Name=Hadoop Name=Hive
--bootstrap-actions '[{"Path":"s3://infa-software/LDM/1010_364/
bootstrap_sample_rpm_installer.bash","Name":"Custom action"}]'
--ec2-attributes
'{"KeyName":"sonoma_preview_key","InstanceProfile":"EMR_EC2_DefaultRole","SubnetId":"sub
net-b98ff5dc","EmrManagedSlaveSecurityGroup":"sg-
c2d9e4a4","EmrManagedMasterSecurityGroup":"sg-c5d9e4a3"}'
--service-role EMR_DefaultRole
--enable-debugging
--release-label emr-4.6.0
--log-uri 's3n://aws-logs-964822751373-us-west-2/elasticmapreduce/'
--name 'My cluster'
--instance-groups '[{"InstanceCount":
1,"InstanceGroupType":"MASTER","InstanceType":"m3.xlarge","Name":"Master instance group
- 1"},{"InstanceCount":
2,"InstanceGroupType":"CORE","InstanceType":"m3.xlarge","Name":"Core instance group -
2"}]'
--configurations '[{"Classification":"core-site","Properties":
{"fs.s3.awsAccessKeyId":"${AWS_ACCESS_KEY}","fs.s3.awsSecretAccessKey":"$
{AWS_SECRET_KEY}"}, "Configurations":[]},
                  {"Classification":"hive-site","Properties":{
                        "hive.metastore.warehouse.dir": "s3://infa-datalake/user/hive/
warehouse",
 "javax.jdo.option.ConnectionURL" : "jdbc:mysql://${HIVE_DB_HOST}:${HIVE_DB_PORT}/
$database.name?createDatabaseIfNotExist=true",
 "javax.jdo.option.ConnectionPassword": "******"
 },"Configurations":[]}]'
--region us-west-2
```

Step 4. Configure the Data Integration Service

Update the hive-site.xml and yarn-site.xml files on the Data Integration Service node.

1. On the Data Integration Service node, copy the following properties into hive-site.xml and yarn-site.xml files:

```
<property>
<name>fs.s3n.endpoint</name>
<value>s3 endpoint region location. this is needed if s3 region is different from emr
region</value>
</property>
<property>
<name>hive.metastore.warehouse.dir</name>
<value><this is usually going to point to hdfs location but can be changed to point to
s3 location as well. this property is needed only in hive-site.xml></value>
</property>
<property>
<name>fs.s3.awsAccessKeyId</name>
```

```
<value>s3 access key</value>
</property>
<property>
<name>fs.s3.awsSecretAccessKey</name>
<value>s3 secret access key</value>
</property>
```

2. Find the following .jar files on the Amazon EMR cluster:
   - emrfs-hadoop-assembly-2.9.0.jar
   - s3-dist-cp.jar
   - hadoop-common-2.7.2-amzn-3.jar
3. Copy the .jar files to the following location in the Data Integration Service node:

   ```
   /<Informatica installation directory>/Informatica/services/shared/hadoop/<distribution name>_<version number>/lib
   ```
4. Create and configure the Data Integration Service to connect to use the cluster in pushdown mode.

Step 5. Configure Live Data Map and Intelligent Data Lake

Configure Live Data Map and log in to Intelligent Data Lake to work with the assets in Amazon EMR.

1. Create and configure a resource in Live Data Map to point to the Amazon EMR cluster.

   For more information, see the *Informatica 10.1.1 Update 2 Live Data Map Installation and Configuration Guide*.
2. Create and configure an Intelligent Data Lake Service pointing to the Amazon EMR resource.

   For more information, see the *Informatica 10.1.1 Update 2 Intelligent Data Lake Installation and Configuration Guide*.
3. Make sure you can access the data stored on Amazon S3 using Intelligent Data Lake through Amazon EMR:
   a. Log in to Amazon EMR.
   b. Create a Hive table pointing to Amazon S3 using the following command:

      ```
      CREATE TABLE mydata (key STRING, value INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY '=' LOCATION 's3://infa-datalake/users';
      ```
   c. Run the Live Data Map resource.
   d. Log in to Intelligent Data Lake.

      You'll find the newly created table in the list of data assets when you search for it. You can perform operations on this data asset.
4. Make sure you can upload and add data assets using Intelligent Data Lake:
   a. Log in to Intelligent Data Lake.
   b. Upload a file in Intelligent Data Lake.

      When you upload a file, a table is created in Hive and the data is stored in Amazon S3.

Tips and Tricks

Here are some tips, tricks, and best practices that you can follow while configuring Intelligent Data Lake on Amazon EMR.

- Configure the staging area for file upload in Intelligent Data Lake to use HDFS location through the HDFS connection. Intelligent Data Lake does not support Amazon S3 location for staging area.

- The schema name configured under Hive Pushdown Configuration in the Hadoop connection can point to a schema whose default storage location is Amazon S3 or HDFS. Informatica recommends using a schema configured with HDFS location.
- This configuration can only access a single Amazon S3 account located at a single endpoint. You can either use the default one or set it up by providing the fs.s3n.endpoint. You cannot use different Amazon S3 accounts for different schemas.
- As Amazon S3 access keys need to be included in yarn-site.xml for the Blaze engine to work, you can face issues when the Amazon EMR instance is configured to scale the number of worker nodes based on usage.
- In Amazon EMR 5.0, HBase is configured to use HDFS. So, you have to manually backup the HBase data to Amazon S3.

Authors

Aravind Kumar Arunachalam
Principal QA Engineer

Chakravarthy Tenneti
Lead Technical Writer