

Implementing Informatica Big Data Management 10.2 in an Amazon Cloud Environment

Abstract

You can take advantage of cloud computing efficiencies and power by deploying a Big Data Management solution in the Amazon AWS environment. You can use a hybrid solution to offload or extend on-premises applications to the cloud. You can also use a lift-and-shift strategy to move an existing on-premises big data solution to the Amazon EMR environment to improve processing speed. This article describes the architecture and advantages of migrating your big data solution to Amazon AWS, how to implement the one-click Big Data Management deployment, and how to implement ephemeral clusters, and auto-scaling.

Supported Versions

- Informatica Big Data Management 10.2

Table of Contents

Overview	3
AWS Technology Overview.	3
AWS Building Blocks.	3
AWS Regions and Availability Zones.	4
Networking, Connectivity, and Security.	5
Understanding Amazon Cluster Types.	5
Amazon Node Types.	5
Amazon EMR Layers.	6
Options for Deploying Big Data Management on Amazon EMR.	7
Cloud Deployment.	7
Hybrid Deployment.	8
Deploying Big Data Management on Amazon EMR through the Amazon Marketplace.	9
Configuring Big Data Management in the Amazon Cloud Environment.	10
Pre-Implementation Tasks.	10
Provision Amazon EMR Cluster Resources and the Informatica Domain.	11
Monitoring Instance Provision and Informatica Domain Creation.	15
Functionality.	16
Connectivity.	16
Transformations.	16
Ephemeral Clusters.	17
Auto-scaling.	20
Data Lake Use Case.	23
Lift and Shift Use Case.	24
Best Practices.	25
Guidelines for Selecting EC2 Instances for the EMR Cluster.	25
Cluster Sizing Guidelines.	27
Guidelines and Recommendations for Utilizing Clusters and Storage.	28
Performance Best Practices.	29

Using VPC to Access Resources on Multiple Clusters.	29
Case Studies.	30
Hive on S3 Versus Hive on HDFS.	30
Spark Queries on Different EC2 Instance Sizes.	31
Spark Query Performance on Various EC2 Instance Types.	32
For More Information.	32
Appendix: Bootstrap Script Example.	33

Overview

Customers of Amazon Web Services (AWS) and Informatica can deploy Informatica Big Data Management in the AWS public cloud.

Using Big Data Management on Amazon EMR provides the following benefits:

- **Faster time to insight.** Dynamic big data integration delivers high throughput data ingestion and data delivery from nearly any source, leveraging Amazon EMR for high performance data processing at scale, and delivering the right analytical data to business stakeholders.
- **Faster time to deployment.** The simple One-Click automated deployment of Big Data Management on EMR from the AWS Marketplace allows organizations to quickly and efficiently deploy a big data integration solution on a high performance cloud infrastructure platform.
- **Accelerated data architecture modernization.** If you are planning to modernize your data strategy initiatives on AWS, Big Data Management provides rich functionality, such as metadata driven data integration, dynamic mappings, and SQL to mapping conversion to help shorten development cycles and reduce time to market.
- **Clean, complete, and trusted data.** Whether you are offloading or extending on-premises applications to the cloud or fully embracing the cloud, collaborative data quality ensures confidence in data fidelity while facilitating data sharing, empowering business stakeholders to curate data, audit data holistically, and relate data at scale. Big Data Management empowers organizations with complete, high-quality, actionable data.

AWS Technology Overview

Amazon Web Services (AWS) provides organizations with a cloud platform for business transactions and data archival, retrieval and processing. When you use AWS as the platform for your big data initiatives, you take advantage of cloud computing efficiencies, security, and functionality.

AWS Building Blocks

Amazon Web Services offers the basic building blocks of storage, networking, and computation, as well as services such as a managed database, big data, and messaging services.

A Big Data Management deployment on EMR can use the following service offerings:

Amazon EC2 instances

Amazon Elastic Compute Cloud (Amazon EC2) instances provide scalable computing capacity in the Amazon Web Services (AWS) cloud. You can launch as many or as few virtual servers as you need with zero investment on hardware. You can configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

Big Data Management can be deployed on Amazon EC2 with the ability to scale up and scale down the environment based on requirements. Big Data Management can be deployed in a mixed environment that contains on-premises machines and Amazon EC2 instances.

Amazon S3 storage

Amazon Simple Storage Service (S3) is easy-to-use object storage with a simple web service interface to store and retrieve any amount of data from anywhere on the web.

Big Data Management provides native, high-volume connectivity to Amazon S3 and support for Hive on S3. It is designed and optimized for big data integration between cloud and on-premise data sources to S3 as object stores.

Amazon Redshift

Amazon Redshift is a cloud-based, fast, fully managed, petabyte-scale data warehouse that makes it simple and cost-effective to analyze all data using existing business intelligence tools. Informatica's PowerExchange for Amazon Redshift connector allow users to securely read data from or write data to Amazon Redshift.

Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while managing time-consuming database administration tasks, freeing you up to focus on your applications and business. Amazon RDS provides you with several familiar database engines to choose from, including Amazon Aurora, Oracle, Microsoft SQL Server, PostgreSQL, MySQL, and MariaDB.

Amazon Aurora

Amazon Aurora is a MySQL-compatible relational database engine that combines the speed and availability of high-end commercial databases with the simplicity and cost-effectiveness of open source databases.

AWS Direct Connect

AWS Direct Connect makes it easy to establish a dedicated network connection from your premises to AWS. Using AWS Direct Connect, you can establish private connectivity between AWS and your datacenter, office, or colocation environment, which in many cases can reduce your network costs, increase bandwidth throughput, and provide a more consistent network experience than Internet-based connections.

Amazon Virtual Private Cloud

Amazon Virtual Private Cloud (Amazon VPC) lets you provision a logically isolated section of the Amazon Web Services (AWS) cloud where you can launch AWS resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.

You can easily customize the network configuration for your Amazon Virtual Private Cloud. For example, you can create a public-facing subnet for your web servers that has access to the Internet, and place your back end systems such as databases or application servers in a private facing subnet with no Internet access. You can leverage multiple layers of security, including security groups and network access control lists, to help control access to Amazon EC2 & EMR instances in each subnet.

AWS Regions and Availability Zones

Regions are self-contained geographical locations where AWS services are deployed. Regions have their own deployment of each service. Each service within a region has its own endpoint that you can interact with to use the service.

Regions contain availability zones, which are isolated fault domains within a general geographical location. Some regions have more availability zones than others. While provisioning, you can choose specific availability zones or let AWS select them for you.

Networking, Connectivity, and Security

Amazon AWS enables the following networking, connectivity, and security features:

Virtual Private Cloud (VPC)

VPC has several different configuration options. See the VPC documentation for a detailed explanation of the options and choose based on your networking requirements. You can deploy Big Data Management in either public or private subnets.

Connectivity to the Internet and Other AWS Services

Deploying the instances in a public subnet allows them to have access to the Internet for outgoing traffic as well as to other AWS services, such as S3 and RDS.

Private Data Center Connectivity

You can establish connectivity between your data center and the VPC hosting your Informatica services by using a VPN or Direct Connect. We recommend using Direct Connect so that there is a dedicated link between the two networks with lower latency, higher bandwidth, and enhanced security. You can also connect to EC2 through the Internet via VPN tunnel.

Security Groups

You can define rules for EC2 instances and define allowable traffic, IP addresses, and port ranges. Instances can belong to multiple security groups.

Understanding Amazon Cluster Types

The central component of Amazon EMR is the cluster. A cluster is a collection of Amazon Elastic Compute Cloud (Amazon EC2) instances. Each instance in the cluster is called a node. Each node has a role within the cluster, referred to as the node type. Amazon EMR also installs layers of software components on each node type, giving each node a role in a distributed application like Apache Hadoop.

Amazon Node Types

Amazon EMR nodes are of the following types:

Master node

Manages the cluster by running software components which coordinate the distribution of data and tasks among other nodes—so-called slave nodes—for processing. The master node tracks the status of tasks and monitors the health of the cluster.

Core node

A slave node that has software components which run tasks and store data in the Hadoop Distributed File System (HDFS) on the cluster.

Task node

An optional slave node that has software components which only run tasks.

Amazon EMR Layers

Amazon EMR service architecture consists of several layers, each of which provides certain capabilities and functionality to the cluster. An EMR cluster has the following layers:

Storage

The storage layer includes the different file systems that are used with your cluster. You can choose from among the following storage options:

- **Hadoop Distributed File System (HDFS)** is a distributed, scalable file system for Hadoop. HDFS distributes the data it stores across instances in the cluster, storing multiple copies of data on different instances to ensure that no data is lost if an individual instance fails. This ephemeral storage that is reclaimed when you terminate a cluster. HDFS is useful for caching intermediate results during MapReduce processing or for workloads which have significant random I/O.
- **EMR File System (EMRFS)**. Amazon EMR uses the EMR File System to enable Hadoop to access data stored in Amazon S3 as if it were a file system like HDFS. You can use either HDFS or Amazon S3 as the file system in your cluster. Most often, Amazon S3 is used to store input and output data, and intermediate results are stored in HDFS.
- **Local File System**. The local file system refers to a locally connected disk. When you create a Hadoop cluster, each node is created from an Amazon EC2 instance that comes with a pre-configured block of pre-attached disk storage called an instance store. Data on instance store volumes persists only during the life cycle of its Amazon EC2 instance.

Cluster resource management

The resource management layer is responsible for managing cluster resources and scheduling the jobs for processing data.

By default, Amazon EMR uses YARN (Yet Another Resource Negotiator), which is a component introduced in Apache Hadoop 2.0 to centrally manage cluster resources for multiple data-processing frameworks. However, there are other frameworks and applications that are offered in Amazon EMR that do not use YARN as a resource manager. Amazon EMR also has an agent on each node which administers YARN components, keeps the cluster healthy, and communicates with the Amazon EMR service.

Data processing frameworks

The data processing framework layer is the engine used to process and analyze data. Many frameworks run on YARN or have their own resource management.

Informatica documentation refers to these data processing frameworks as run-time engines. The engine you choose depends on processing needs, such as batch, interactive, in-memory, or streaming. Your choice of run-time engine affects the languages and interfaces on the application layer, which is the layer used to interact with the data you want to process.

The following main run-time engines are available for Amazon EMR:

- **Spark**. Apache Spark is a cluster framework and programming model for processing big data workloads. Like Hadoop MapReduce, Spark is an open-source, distributed processing system but uses directed acyclic graphs for execution plans and leverages in-memory caching for datasets. Spark supports multiple interactive query modules such as SparkSQL.
- **Blaze**. Informatica Blaze is the industry's unique data processing engine integrated with YARN to provide intelligent data pipelining, job partitioning, job recovery, and scalability, which is optimized to deliver high performance, scalable data processing leveraging Informatica's cluster aware data integration technology.

- **Hadoop MapReduce.** Hadoop MapReduce is an open-source programming model for distributed computing. It simplifies the process of writing parallel distributed applications by handling all of the logic, while you provide the Map and Reduce functions. The Map function maps data to sets of key value pairs called intermediate results. The Reduce function combines the intermediate results, applies additional algorithms, and produces the final output. There are multiple frameworks available for MapReduce, such as Hive, which automatically generate Map and Reduce programs.

Options for Deploying Big Data Management on Amazon EMR

Informatica and Amazon AWS make available a completely automated deployment of Big Data Management on Amazon EMR cluster through the Amazon Marketplace. By default, the deployment consists of a minimum recommended M3 instance type using the m3.xlarge model that provides a balance of compute, memory, and network resources required for Big Data Management services and Amazon EMR cluster.

Choose from the following options for deploying Big Data Management on Amazon EMR:

Cloud deployment

Choose one of these options for deploying Big Data Management on the AWS cloud:

- One-click deployment in the Amazon Marketplace is best suited for proof of concept and prototyping big data projects where Amazon AWS and Big Data Management services are deployed automatically on AWS infrastructure.
If you use this option, installation and configuration are automated, saving you time and effort.
- Manual deployment. Manually install and configure the Informatica domain and Big Data Management on an Amazon EMR cluster.
If you use this option, it takes longer to install and configure the software, but you have the ability to customize your deployment.

Hybrid deployment

When you choose the hybrid option, you install and configure the Informatica domain and Big Data Management on-premise, and configure them to push processing to the Amazon EMR cluster.

Advantages of this approach include:

- You can customize the deployment to meet your requirements.
- You have full control over the deployment.
- You can use leading Hadoop distributions, including Cloudera, Hortonworks, MapR, and IBM BigInsights as Hadoop environments on Amazon EMR.

Disadvantages of this approach include:

- Extra time and manual effort to install and configure software.
- Network latency when you push a mapping to the Amazon EMR cluster.
- Maintenance of the Informatica domain and Big Data Management software, as well as a Hadoop environment.

Cloud Deployment

You can deploy the Informatica domain and Big Data Management on the Amazon AWS cloud.

Choose from the following options for deploying Big Data Management on Amazon EMR:

One-Click Deployment in the Amazon Marketplace

Deploy a default configuration of Big Data Management, including the Informatica domain and application services, through the Amazon Marketplace.

One-click deployment provisions cluster nodes with the latest Big Data Management software. The automated process assigns the correct number of Big Data Management domain and Amazon EMR nodes and installs and configures Big Data Management on an Amazon EC2 instance during the provision of cluster nodes.

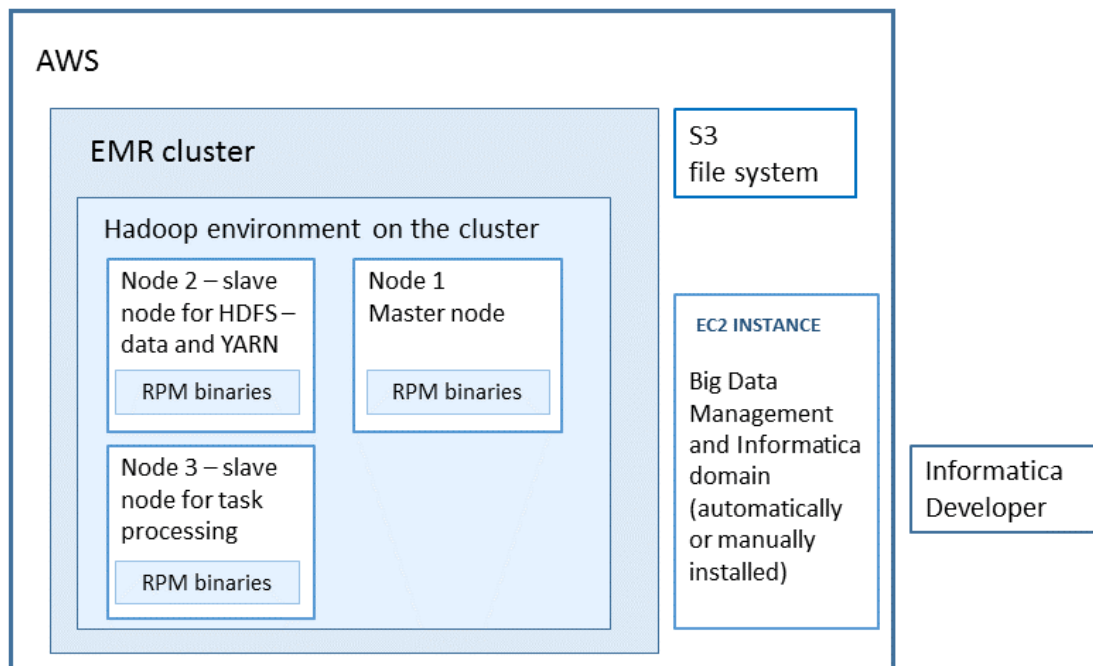
The deployment automatically creates the Informatica domain, the Model Repository Service, and the Data Integration service, and assigns the connection to the Amazon EMR cluster for HDFS and Hive. It provisions the Amazon EMR cluster with HDFS and EMRFS for storage, YARN for processing, and Hive and Spark run-time engines.

Manual Deployment in the Amazon AWS Cloud

Deploy Big Data Management in the AWS cloud, leveraging Amazon EMR and other leading Hadoop distributions, such as Cloudera, Hortonworks, MapR.

In this mode of deployment, you install the Informatica domain and Big Data Management on an Amazon EMR cluster. You use a local installation of the Developer tool to create and run mappings. You can use the native Amazon EMR Hadoop utilities, or access a different distribution of Hadoop that you installed in the Amazon EMR environment.

The following image shows the architecture of a cloud deployment, with either automatic or manual installation of Big Data Management RPM binaries on cluster nodes, and either automatic or manual installation of Informatica components on an EC2 instance:



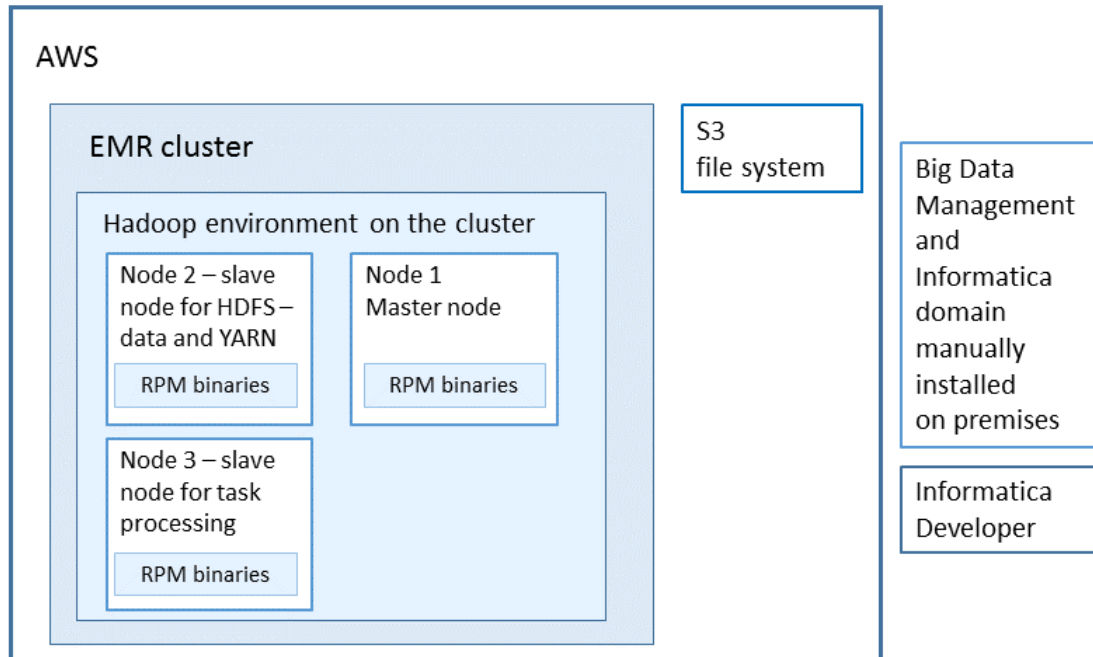
Note: For information about how to perform a manual deployment of Big Data Management in the AWS cloud, see [Big Data Management 10.2 user documentation on the Informatica Network](#).

Hybrid Deployment

When you choose a hybrid option, you install and configure the Informatica domain and Big Data Management on-premises, and configure them to run mappings on the Amazon EMR cluster. Run mappings in a Hadoop environment hosted on an Amazon EMR cluster where you manually install Big Data Management RPM binaries. You can use the

native Amazon EMR Hadoop utilities or access a different distribution of Hadoop that you installed in the Amazon EMR environment.

The following image shows the architecture with Big Data Management RPM binaries and the Hadoop distribution on Amazon EMR, and the domain and client on premises:



Note: For information about how to deploy Big Data Management 10.2 in the AWS cloud, see [Big Data Management 10.2 user documentation on the Informatica Network](#).

Deploying Big Data Management on Amazon EMR through the Amazon Marketplace

Informatica and Amazon AWS enable a completely automated deployment of Big Data Management on Amazon EMR cluster through the Amazon Marketplace.

The default one-click deployment consists of a minimum recommended EC2 instance type using the m3.xlarge model to provide a balance of compute, memory, and network resources required for Big Data Management services and the Amazon EMR cluster. You can choose other M3 instance types or C3 Compute-optimized instances for high performing processing. You can choose any number of core nodes for the Amazon EMR cluster.

One-click deployment provisions the nodes with the latest Big Data Management software. The automated process assigns the correct number of Big Data Management domain and Amazon EMR nodes and installs and configures Big Data Management on an Amazon EC2 instance during the provision of cluster nodes. The deployment automatically creates the Informatica domain, the Model Repository Service, and the Data Integration service, and assigns the connection to the Amazon EMR cluster for HDFS and Hive. It provisions the Amazon EMR cluster with HDFS and EMRFS for storage, YARN for processing, and Hive and Spark run-time engines.

The Informatica domain and repository database are hosted on Amazon RDS using MS SQL Server, which handles management tasks such as backups, patch management, and replication. To control and configure Informatica services, you use the browser-based Administrator tool, which accesses the domain instance. To configure and run mappings, you use the Developer tool client on a Windows machine.

Configuring Big Data Management in the Amazon Cloud Environment

You can choose to enable Big Data Management for Amazon EMR in the Amazon cloud environment. When you create an implementation of Big Data Management in the Amazon cloud, you bring online virtual machines where you install and run Big Data Management.

First, perform pre-implementation tasks. Then configure an Amazon EMR cluster with storage and S3 connectivity. Finally, monitor cluster provision and the creation of the Informatica domain.

Pre-Implementation Tasks

Before you configure Big Data Management in the Amazon EMR cloud environment, perform the tasks in this section.

Verify Prerequisites

Before you configure Big Data Management in the Amazon EMR cloud environment, verify the following prerequisites:

- You have purchased a license for Big Data Management and have uploaded the Big Data Management license file to an Amazon s3 bucket.
The license file has a name like `BDMLicense.key`.
- Your IAM user has permissions to create EMR clusters, Elastic IP addresses, EC2 instances, Amazon Relational Database Service (RDS), and the additional resources and services to support them.
- You have configured an Amazon private key (.pem file) to use for authentication during setup.
- You have configured a VPC infrastructure with DNS support enabled, and with at least two available subnets. One of these subnets must meet the following requirements:
 - Must be a member of the VPC.
 - Must have access to the internet via gateway. The Informatica domain must use this subnet.
 - Must have auto-assign public IPv4 enabled.The other two subnets must meet the following requirements:
 - Must be members of the VPC.
 - Must be in two separate regions, to enable database failover.
 - May be private or public. Any public subnet must have auto-assign public IPv4 enabled.

Open Ports

When you manually install Big Data Management in the AWS cloud, you must open ports to enable communication between the Informatica domain and the cluster.

Create a security group in the AWS account and specify the ports listed in the following table. Open these ports for both inbound and outbound communication. Make a note of the security group and include it in the list of parameters in the script to create the cluster.

For ports that are necessary to open for client access, also open the ports on the Developer tool machines.

Note: When you use the one-click method to deploy Big Data Management on AWS, the process configures these ports automatically.

The following list of ports to open presumes that the Informatica domain is on the AWS cluster:

Port configuration property name	Port number	Client Access
mapreduce.jobhistory.address	10020	No
mapreduce.jobhistory.webapp.address	19888	Yes
yarn.resourcemanager.scheduler.address	8030	No
yarn.resourcemanager.webapp.address	8088	Yes
yarn.resourcemanager.address	8032	Yes
yarn.resourcemanager.resourcetracker.address	8025	No
yarn.web-proxy.address	20888	Yes
yarn.container log	8040	Yes
yarn.nodemanager.address	8041	No
yarn.timeline-service.address	8188	Yes
dfs.encryption.key.provider.uri	9700	No
hdfs	8020	Yes
dfs.datanode.address	5100	No
hive.server2.thrift.port	10000	Yes
hive.metastore.port	9083	Yes
infagrid.blaze.console.jfSPORT	9090	No
infagrid.blaze.console.httpport	9080	Yes
blaze execution range Note: The values that you use for the Blaze port range depend on the values that the cluster connection uses. A total range of approximately 2500 ports is recommended.	10600-12300	No

Provision Amazon EMR Cluster Resources and the Informatica Domain

You can use the AWS marketplace to provision cluster resources and install Big Data Management in the cluster.

1. Go to the Amazon AWS marketplace (<https://aws.amazon.com/marketplace>).
2. Search for and select **Informatica Big Data Management 10.2.0**, and then click **Continue**.

The **Create Stack** screen opens. The following image shows part of the **Create Stack** screen:

Specify Details

Specify a stack name and parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. [Learn more](#).

Stack name

Parameters

Network Configuration

VPC

Search by ID, or Name tag value

Which VPC should this be deployed to?

KeyName

Search

Name of an existing EC2 KeyPair to enable SSH access to the Informatica Domain

Informatica Domain Subnet

Search by ID, or Name tag value

Select a publically accessible subnet ID for the Informatica Domain

Informatica Database

Search by ID, or Name tag value

Subnets

Select two subnet IDs each from a different region in the VPC chosen above (such as: us-west-1b, us-west-1c)

IP Address Range

0.0.0.0/0

The range of IP addresses to access the Informatica domain and EMR cluster

3. In the **Stack name** field, type the name of the stack instance to create.
4. In the **Parameters** section, enter the following information in the **Network Configuration** area:

Property	Description
VPC	Select the Virtual Private Cloud (VPC) location to install Big Data Management. The VPC is a provisioned computing instance on Amazon's AWS cloud. Amazon AWS provides one or more VPC with each account. Each VPC has a range of IP addresses. The VPC must meet the following requirements: <ul style="list-style-type: none">- Set up with public access through the internet via an attached internet gateway.- The DNS Resolution property of the VPC must be set to Yes.- The Edit DNS Hostnames property of the VPC must be set to Yes.
KeyName	Select an existing EC2 KeyPair name to enable SSH access for Informatica services to the EC2 instance. This might be the key pair that you created in the Prerequisites section.
Informatica Domain Subnet	Select a publically accessible subnet for the Informatica domain.

Property	Description
Informatica Database Subnets	<p>Specify the IDs of three different subnets.</p> <p>One of these subnets must meet the following requirements:</p> <ul style="list-style-type: none"> - Must be a member of the VPC. - Must have access to the internet via gateway. The Informatica domain must use this subnet. - Must have auto-assign public IPv4 enabled. <p>The other two subnets must meet the following requirements:</p> <ul style="list-style-type: none"> - Must be members of the VPC. - Must be in two separate regions, to enable database failover. - May be private or public. Any public subnet must have auto-assign public IPv4 enabled. <p>It is not necessary to choose subnets in the domain subnet.</p>
IP Address Range	<p>IP address range to use to limit SSH access from the Informatica domain to the EC2 instance.</p> <p>For example, to specify the range of 10.20.30.40 to 10.20.30.49, enter the following string:</p> <p>10.20.30.40/49</p>

5. In the **Amazon EC2 Configuration** section, enter the following information to configure the Informatica domain and the domain repository database:

Property	Description
Informatica Domain Instance Type	<p>Select the type for the instance to host the Informatica domain.</p> <p>Each type corresponds to a different size, in ascending order of size. Default is m4.large.</p> <p>Note: When you select an instance type here and in later steps, be aware that Amazon charges more when you select a larger instance type.</p>
Informatica Administrator User Name	<p>Enter the administrator user name for Big Data Management.</p> <p>In this field and the following field, you can specify any user name and password. Make a note of the user name and password, and use it later to log in to the Administrator tool to configure the Informatica domain.</p>
Informatica Administrator Password	<p>Enter the administrator password for Big Data Management.</p>
BDM License Key Location	<p>Enter the location of the Big Data Management license key file. The location is the name of the S3 bucket where the key was saved. For example:</p> <p>myBucketName</p>
BDM License Key Name	<p>Enter the path and filename of the Big Data Management license key file in the S3 bucket location.</p> <p>The path must include subdirectories <i>under</i> the bucket name.</p> <p>For example, where the entire path including the bucket name is myBucketName/SubDir1/SubDir2/BDMLicense.key, type the following:</p> <p>SubDir1/SubDir2/BDMLicense.key</p>

6. In the **Amazon RDS Configuration** section, enter the following information to enable Amazon RDS to access the Informatica domain database:

Property	Description
Informatica Database Username	User name for the domain and the Model repository database. In this field and the following field, you can specify any user name and password.
Informatica Database Password	Password for the domain and the Model repository database.

7. in the **Amazon EMR Configuration** section, enter the following information to provision the Amazon EMR cluster:

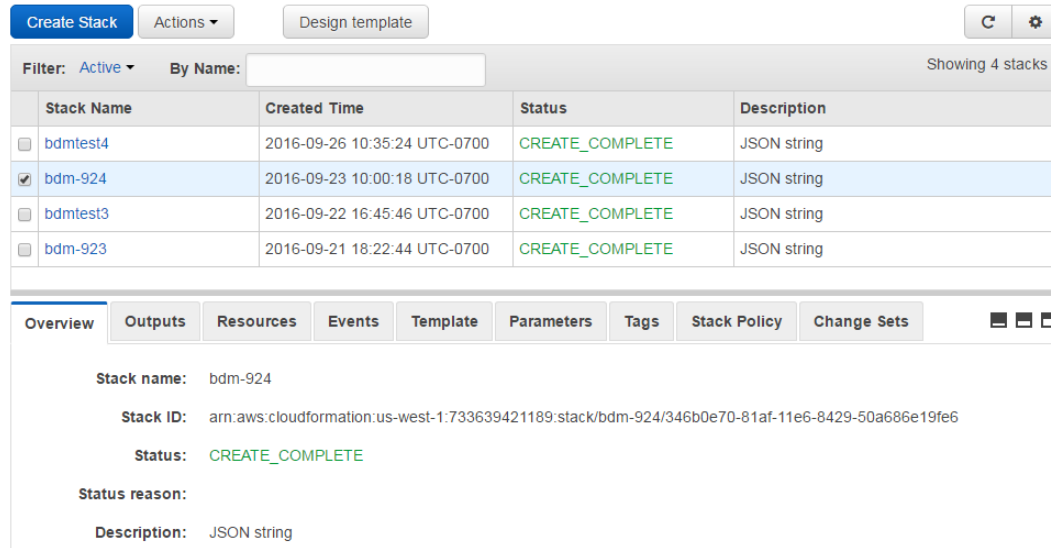
Property	Description
EMR Cluster Name	Enter a name for the Amazon EMR cluster where the BDM instance will be deployed.
EMR Master Node Instance Type	Select the instance type. Each type corresponds to a different size, in ascending order of size. Default is m3.xlarge.
EMR Core Nodes [instance type]	Select the instance type. Each type corresponds to a different size, in ascending order of size. Default is m4.large.
EMR Core Nodes [number of core nodes]	Enter an integer for the number of core nodes to support the Big Data Management deployment. Minimum is 1. Maximum is 500. Note: Informatica requires at least a minimum of 8 CPU vCores and 30 GB memory.
EMR Logs Bucket Name	Enter the name of the S3 bucket where EMR stores logs.

8. Click **Next**.
The **Options** page opens.
9. Optionally enter key labels and values for resources, and then click **Next**.
10. Review the configuration, then click the check box to acknowledge that you are about to create resources.

11. Click **Create**.

Amazon AWS begins to create the stack. Amazon AWS displays the **Cloud Formation** dashboard.

The following image shows the **Cloud Formation** dashboard:



Monitoring Instance Provision and Informatica Domain Creation

You can monitor the progress of creating the cluster instance and Informatica domain.

1. Select the stack that you are creating, then select the **Events** tab.
2. When the stack creation is complete, select the **Resources** tab.

The **Resources** tab displays information about the stack and Big Data Management instance. You can select the linked Physical ID properties of resources to get more information about them.

3. Click the **Outputs** tab.

When the Informatica domain setup is complete, the **Outputs** tab displays the following information:

Property	Description
ICSMultiNodeClusterURL	URL to access the Informatica Cluster Service Hadoop gateway node. If you had selected the cluster size as <i>small</i> , the Output tab displays the <i>ICSSingleNodeClusterURL</i> property name.
InstanceID	Hostname of the Informatica domain.
InstanceSetupLogs	Location of the setup log files for the Informatica domain EC2 instance.
InformaticaAdminConsoleURL	URL of Informatica Administrator. Use the Administrator tool to administer Informatica services.
InformaticaAdminConsoleServerLogs	Location of the Informatica Administrator log files.

Note: If the **Outputs** tab is not populated with this information, wait for domain setup to be complete.

4. Open the **Resources** tab. Click the **Physical ID** of the `AdministrationServer` property. The **Physical ID** corresponds to the name of the Informatica domain.

The **Instance Administration** screen opens.

You can use the **Instance Administration** screen to launch the Enterprise Information Catalog instance. You can also get information such as the public DNS and public IP address.

Functionality

When you deploy Big Data Management in the Amazon AWS cloud environment, you can run mappings in a cluster on Amazon EMR against sources and targets in Amazon S3 or other Amazon data repositories. This section explains the following functional features:

- **Connectivity.** Perform configuration tasks to enable connectivity between your data warehouse and AWS.
- **Transformations.** Transformations perform specific functions within big data mappings.
- **Ephemeral clusters.** Use an ephemeral cluster strategy to save AWS resources.
- **Auto-scaling.** Configure auto-scaling to scale out (increase) and scale in (decrease) core and task nodes for Spark mapping processing.

Connectivity

You can use the following methods to read and write data between your data warehouse and Amazon AWS:

PowerExchange Adapter for Amazon S3

You can use PowerExchange for Amazon S3 to read and write delimited flat file data and binary files as pass-through data from and to Amazon S3 buckets. Create an Amazon S3 connection to specify the location of Amazon S3 sources and targets you want to include in a data object. You can use the Amazon S3 connection in data object read and write operations.

For more information about the Amazon S3 Adapter, see the *Informatica PowerExchange for Amazon S3 User Guide*.

PowerExchange Adapter for Amazon Redshift

You can use PowerExchange for Amazon Redshift to read data from or write data to Amazon Redshift. You can also use PowerExchange for Amazon Redshift to read data from Amazon Redshift views. You can use Amazon Redshift objects as sources and targets in mappings.

For more information about the Amazon Redshift Adapter, see the *Informatica PowerExchange for Amazon Redshift User Guide*.

Hive on Amazon S3

When you configure properties on the cluster and on the Informatica domain to access Hive tables on Amazon S3, you can use a Hive connection to read data from and write data to Hive tables.

For more information about configuring access to Hive tables on Amazon S3, see the *Informatica Big Data Management Installation and Configuration Guide*.

Transformations

Informatica Developer provides a set of transformations that perform specific functions. For example, an Aggregator transformation performs calculations on groups of data. Transformations in a mapping represent the operations that

the Data Integration Service performs on the data. Data passes through transformation ports that you link in a mapping or mapplet.

For more information about transformations that you can use in big data mappings, see the *Informatica Developer Transformation Guide*.

Ephemeral Clusters

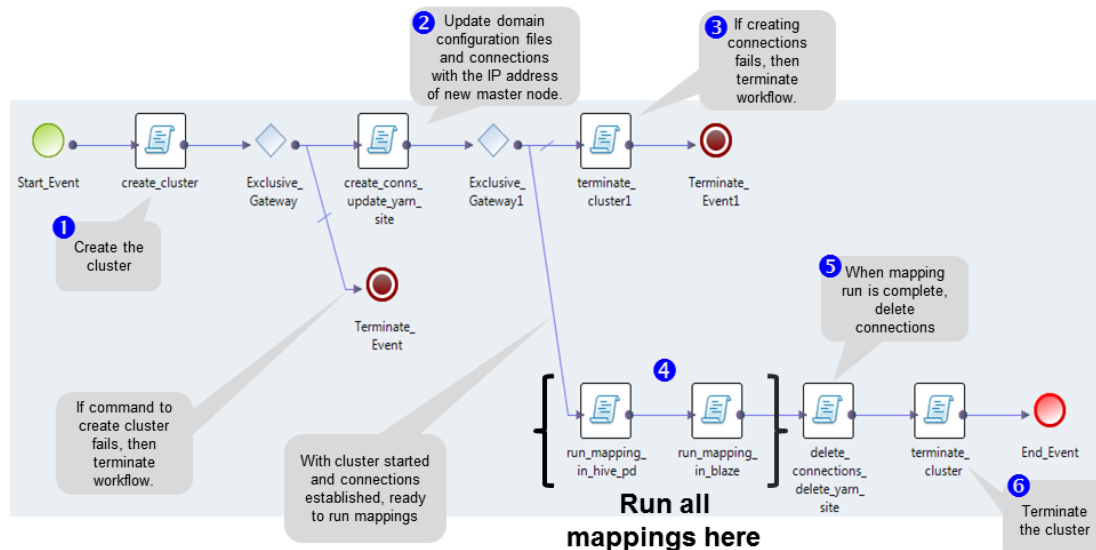
In an ephemeral cluster strategy, the clusters are created, exist for the time it takes for jobs to complete, and then cease to exist when they are brought down.

Create a workflow to implement the strategy. An Informatica workflow contains Command tasks that use scripts that spin up and configure an Amazon EMR cluster, and Mapping tasks that run mappings. When the mapping runs are complete, additional Command tasks can then terminate the cluster, so that you use Amazon EMR cluster resources only when you need them.

Note: Big Data Management 10.2.1 introduces new functionality for cloud platform clusters.

- For the Amazon AWS and Microsoft Azure cloud platforms, you can create a workflow using new Create Cluster and Delete Cluster workflow tasks. For more information about using these tasks in a workflow, see the article "[How to Create Cloud Platform Clusters Using a Workflow on Big Data Management](#)."
- You can use Command tasks in a workflow create a Cloudera Altus cluster on AWS. For information about how to create an ephemeral cluster on Cloudera Altus on AWS, see the article "[How to Create Cloudera Altus Clusters with a Cluster Workflow on Big Data Management](#)."

The following image shows a typical workflow for using an ephemeral cluster:



The workflow contains Command and Mapping tasks that run scripts, as described in the following steps:

Step 1. Creates the cluster.

The Command task contains a script that creates an EMR cluster using information from an exported list of cluster properties, and writes information to an output file.

The script runs on all nodes of the cluster to install Informatica binaries and configure the cluster to communicate with the Informatica domain.

Step 2. Updates configuration files and connections with cluster master node IP address.

The Command task contains a script that lists configuration files and domain connections and updates them with a variable that represents the IP address of the cluster master node.

Step 3. If the cluster fails to start, terminates the cluster

The Command task contains a script that deletes connections and terminates the cluster. The script is triggered by the failure of the cluster to start successfully.

Step 4. Runs mappings.

Mapping tasks in the workflow run mappings.

Step 5. Deletes connections and configuration files.

After the mapping run, the Command task contains a script that deletes connections and configurations files that are no longer necessary.

Step 6. Terminates the cluster.

This script for this Command task is similar to the script in step 3.

Step 1. Create the Cluster

In the console of the cluster where you want to implement Big Data Management, Click **AWS CLI export** to generate a file with information required by the Create Cluster command. Use the properties and values that it contains to populate a script that performs the following actions:

1. Launches a bootstrap script to install Informatica binaries in an s3 bucket.

The bootstrap section of the cluster creation script points to the location of Informatica RPM binary files that you uploaded to an s3 bucket. See the sample bootstrap script in the appendix of this article.

2. Creates a cluster on AWS with the properties from the file you generated with the **AWS CLI export** button, waits for the cluster creation to finish, and writes output information to a file with a name like `ClusterCreation.txt` in a temporary directory.

The following text is an example of the cluster creation command:

```
aws emr create-cluster --termination-protected --applications Name=Hadoop Name=Hive
Name=Pig Name=Hue --bootstrap-actions '[{"Path":"s3://s3-jj-demo/
installHadoopRPM1010.sh","Name":"Install RPM 1010"}]' --tags
'BUSINESSUNIT=DTMQAHQ' 'APPLICATIONENV=TEST' 'RUNNINGSCHEDULE=ENABLED' 'APPLICATIONTYPE=E
MRCLUSTER' 'NAME=xxx' --ec2-attributes
'{"KeyName":"xxx","InstanceProfile":"EMR_EC2_DefaultRole","SubnetId":"xxx","EmrManagedSla
veSecurityGroup":"sg-5a8d5b3f","EmrManagedMasterSecurityGroup":"sg-5b8d5b3e"}' --release-
label emr-5.0.0 --log-uri 's3n://xxx/elasticmapreduce/' --instance-groups
'[{ "InstanceCount":
1, "InstanceGroupType": "MASTER", "InstanceType": "m3.xlarge", "Name": "Master - 1"},
{ "InstanceCount": 3, "InstanceGroupType": "CORE", "InstanceType": "m3.xlarge", "Name": "Core -
2"}]' --configurations '[{"Classification": "yarn-site", "Properties":
{"yarn.nodemanager.resource.cpu-vcores": "24", "yarn.nodemanager.resource.memory-
mb": "16000", "yarn.scheduler.maximum-allocation-mb": "16384", "yarn.scheduler.minimum-
allocation-mb": "256", "yarn.nodemanager.vmem-check-enabled": "false"}, "Configurations": []},
{ "Classification": "core-site", "Properties": {"hadoop.proxyuser.hadoop.groups": "*",
"hadoop.proxyuser.hadoop.hosts": "*", "fs.s3.enableServerSideEncryption": "true", "hadoop.pro
xyuser.yarn.groups": "*", "hadoop.proxyuser.yarn.groups": "*", "fs.s3.awsAccessKeyId": "xxx", "f
s.s3.awsSecretAccessKey": "xxx"}, "Configurations": []}]' --auto-scaling-role
EMR_AutoScaling_DefaultRole --service-role EMR_DefaultRole --enable-debugging --name
'xxx' --scale-down-behavior TERMINATE_AT_TASK_COMPLETION --region us-west-2 > /
tmp/.ClusterCreation.txt
```

Notice the creation of an output file at the end of the command. Subsequent scripts reference this file to get information about the cluster.

3. Returns a message that the cluster creation is complete and the workflow can execute the next step.

The script runs on all nodes of the cluster to install Informatica binaries and configure the cluster to communicate with the Informatica domain.

Step 2. Update Configuration Files and Connections

In this step, the workflow uses a script that lists configuration files and domain connections and updates them with a variable that represents the IP address of the cluster master node.

The script performs the following actions:

1. Creates an alias for the IP address of the cluster master node, and gets the IP address from the `ClusterCreation.txt` file that you created in the previous script.

For example,

```
ipaddrOfMaster=`cat /tmp/.ClusterCreation.txt | python -c 'import sys, json; print
json.load(sys.stdin) ["Instances"] [0] ["PrivateIpAddress"]`
```

2. Creates Hive, HDFS, Hadoop, and other connections, including authentication values, essential port numbers, and the IP address of the cluster master node.

For example, the following section of the script creates a Hive connection:

```
echo "" >> /tmp/.emr_automation.log
echo "Creating hive connection" >> /tmp/.emr_automation.log

${infa_home}/isp/bin/infacmd.sh createConnection -dn 'domain' -un 'Administrator' -pd
'Administrator' -cn 'Hive conn test automation' -cid 'Hive conn_test_automation' -ct
HIVE -o "connectString=jdbc:hive2://${ipaddrOfMaster}:10000/default enableQuotes=false
metadataConnString=jdbc:hive2://${ipaddrOfMaster}:10000/default
bypassHiveJDBCServer=false pushDownMode=true relationalSourceAndTarget=true
databaseName=default defaultFSURI=hdfs://${ipaddrOfMaster}:8020/
hiveWarehouseDirectoryOnHDFS='/user/hive/warehouse' jobTrackerURI=${ipaddrOfMaster}:8021
metastoreExecutionMode=remote remotemetastoreuri=thrift://${ipaddrOfMaster}:9083
username='hadoop'" >> /tmp/.emr_automation.log
```

Note:

- The script refers to cluster nodes, including the master node, by the IP address only.
 - The script locates each element, such as the default.FS URI and the metastore, as being on the master node.
 - The example contains only mandatory arguments, but you can add optional arguments to give custom attributes to the connection.
3. Edits the `yarn-site.xml` file on the Informatica domain with the IP address of the cluster master node.

For example,

```
echo "" >> /tmp/.emr_automation.log
echo "Updating yarn-site.xml" >> /tmp/.emr_automation.log

cp ${infa_home}/services/shared/hadoop/amazon_emr5.0.0/conf/yarn-site.xml_org $
${infa_home}/services/shared/hadoop/amazon_emr5.0.0/conf/yarn-site.xml
sed -i -- "s/HOSTNAME/${ipaddrOfMaster}/g" ${infa_home}/services/shared/hadoop/
amazon_emr5.0.0/conf/yarn-site.xml
```

4. Returns a message that the step is complete and the workflow can execute the next step.

Step 3. Terminate a Failed Cluster

If the cluster fails to start successfully, this script deletes connections and terminates the cluster.

For example,

```
echo "" >> /tmp/.emr_automation.log
echo "Terminating cluster ..." >> /tmp/.emr_automation.log

clusterId=`cat /tmp/.ClusterCreation.txt | python -c 'import sys, json; print
json.load(sys.stdin) ["ClusterId"]`

aws emr modify-cluster-attributes --cluster-id $clusterId --no-termination-protected
aws emr terminate-clusters --cluster-ids $clusterId
```

```
echo "" >> /tmp/.emr_automation.log
echo "Workflow done" >> /tmp/.emr_automation.log
exit 0
```

Step 4. Run Mappings

A workflow runs a list of mappings and performs other tasks. You design and deploy a workflow from the Developer tool.

This script runs the workflow using the startWorkflow command.

For more information about designing and running workflows to run mappings, see:

- *Informatica 10.1.1 Update 2 Developer Workflow Guide*
- *Informatica 10.1.1 Update 2 Big Data Management User Guide*

Step 5. Delete Connections and Configuration Files

After the mapping run, this script terminates the cluster and deletes connections and domain configuration files.

Domain connections and configuration files to enable the cluster are no longer needed when the cluster is to be terminated. The script for this step deletes the domain connections and configuration files that you updated in step 2.

For example, the following portion removes the Hive connection:

```
${infa_home}/isp/bin/infacmd.sh isp removeConnection -dn domain -un Administrator -pd
Administrator -cn Hive_conn_test_automation >> /tmp/.emr_automation.log

echo "" >> /tmp/.emr_automation.log
echo "Deleting HDFS connection..." >> /tmp/.emr_automation.log
```

The script contains a similar section for each of the connections and files to remove.

Step 6. Terminate the Cluster

After connections and configuration files are deleted, this script terminates the cluster.

An ephemeral cluster is designed to exist only for the time it takes to start and configure the cluster, run mappings and workflows, and perform post-run tasks. When all these tasks are complete, the cluster itself is terminated so that Amazon does not continue to charge you for AWS resources.

Auto-scaling

Big Data Management supports auto-scaling on EMR clusters for Spark mappings. When you enable auto-scaling, the EMR cluster automatically adds or subtracts cluster core and task nodes when Spark mappings activate performance thresholds.

Set up auto-scaling when you provision cluster resources.

When auto-scaling is enabled, the cluster creates new instances with the same hardware configuration as the specified instance type. For example, if you configured m3.xlarge as the core node type, each new core node created during scale-out has the same hardware configuration.

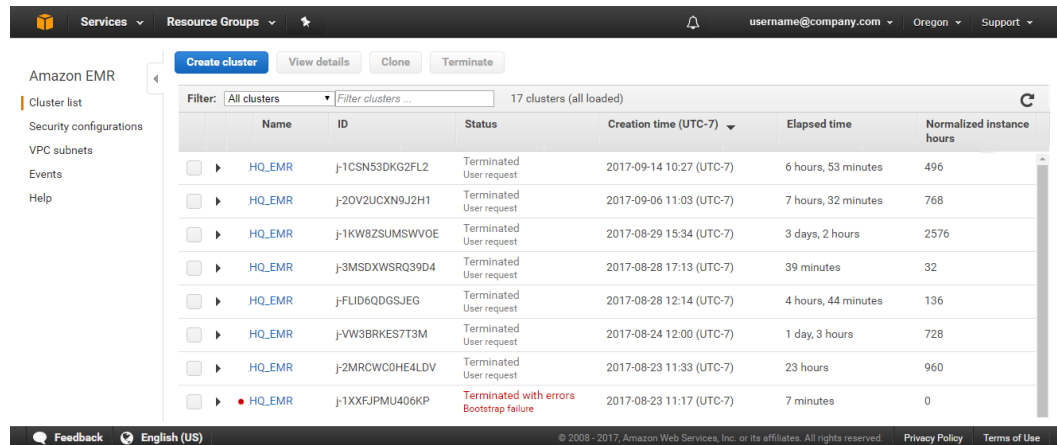
Note: The AWS Marketplace implementation of Big Data Management does not support auto-scaling.

Set Up Auto-scaling

Set up auto-scaling quantities and rules when you want the cluster to automatically provision additional cluster resources to run Spark mappings.



1. Sign in to the AWS Management Console and open the Amazon EMR console at <https://console.aws.amazon.com/elasticmapreduce/>.

The Amazon EMR console opens and displays a list of clusters associated with the account.



2. Select one of the following choices:
 - Create a cluster and configure auto-scaling on it.
 - Edit auto-scaling properties on an existing cluster.
3. To create a cluster and configure auto-scaling on it:
 - a. Select **Create Cluster**.
 - b. Select **Go to Advanced Options**.
 - c. Choose options in the **Step 1: Software and Steps** screen
 - d. Click **Step 2: Hardware Configuration**.
4. To edit auto-scaling options on an existing cluster:
 - a. Click the cluster name.
 - b. Expand the **Hardware** node in the list of configuration options.
5. In the **Auto Scaling** column, click the **Edit** icon to change auto-scaling options for core or task nodes.

The following image shows the locations of the **Edit** icon, which appears as a pencil:

Node type	Instance type	Instance count	Purchasing option	Auto Scaling
Master Master - 1	m3.xlarge 8 vCPU, 15 GiB memory, 80 SSD GB storage EBS Storage: none	1 Instances	<input checked="" type="radio"/> On-demand <input type="radio"/> Spot Maximum bid price: \$	Not available for Master
Core Core - 2	m3.xlarge 8 vCPU, 15 GiB memory, 80 SSD GB storage EBS Storage: none	2 Instances	<input checked="" type="radio"/> On-demand <input type="radio"/> Spot Maximum bid price: \$	Not enabled 
Task Task - 3	m3.xlarge 8 vCPU, 15 GiB memory, 80 SSD GB storage EBS Storage: none	0 Instances	<input checked="" type="radio"/> On-demand <input type="radio"/> Spot Maximum bid price: \$	Not enabled 

The **Auto Scaling** rules window opens.

6. Set the number of maximum and minimum instances.

Maximum instances

The maximum number of cluster nodes, of the type being configured, that all the rules can create.

Minimum instances

The minimum number of cluster nodes of the type being configured.

For example, if you set the number of core Maximum instances to 10 and the Minimum instances to 2, the number of core nodes cannot be greater than 10 or fewer than 2.

7. Verify that **Scale out** is selected, then choose to edit the existing rule or create a new rule.
 - To edit the existing rule, click the **Edit** icon.
 - To add a rule, click **Add rule**.

The following image shows the **Auto Scaling rules** window:

Auto Scaling rules

Maximum instances: ⓘ

Minimum instances: ⓘ

☒ Scale out

Click the Edit icon to edit the rule

Default-scale-out-1: Add instance if YARNMemoryAvailablePercentage is less than for five-minute period with a cooldown of seconds

+ Add rule ← Click Add Rule to add a rule

☒ Scale in

Default-scale-in: Terminate instance if YARNMemoryAvailablePercentage is greater than for five-minute period with a cooldown of seconds

+ Add rule

Done

The **Scale out** section enables you to set rules for increasing the number of cluster nodes.

The following table lists the properties that you can set for each rule:

Property	Description
Rule name	Name of the rule. Type a rule name.
Add	Number of instances to add when the rule conditions are met.
if	Name of the metric to monitor. You can select any AWS CloudWatch metric in the drop-down list.

Property	Description
is	Boolean condition. One of: <ul style="list-style-type: none"> - Greater than or equal to - Greater than - Less than - Less than or equal to
percent	Metric threshold.
for	Evaluation period. Number of consecutive five-minute periods in which to compare the metric to the threshold.
Cooldown period	Number of seconds following activation of a rule during which the rule cannot be re-activated.

For example, the following image shows an example of a configured rule:

☒ **Scale out**

Rule name ✓ ✕

Add Instances ⓘ

if ⓘ

is (percent) ⓘ

for five-minute periods ⓘ

Cooldown period seconds ⓘ

[+ Add rule](#)

These settings translate to the following rule: Add one EC2 instance if the YARNMemoryAvailablePercentage metric is less than 15% for a single five-minute period, and do not reactivate the rule for 300 seconds after it is activated.

- Optionally, click **+ Add rule** to add additional rules.
- Repeat steps 4 and 5 for the **Scale in** section, which sets rules for reducing the number of cluster nodes.
- Click **OK** to complete auto-scaling configuration.
- Click **Next** to continue cluster configuration.

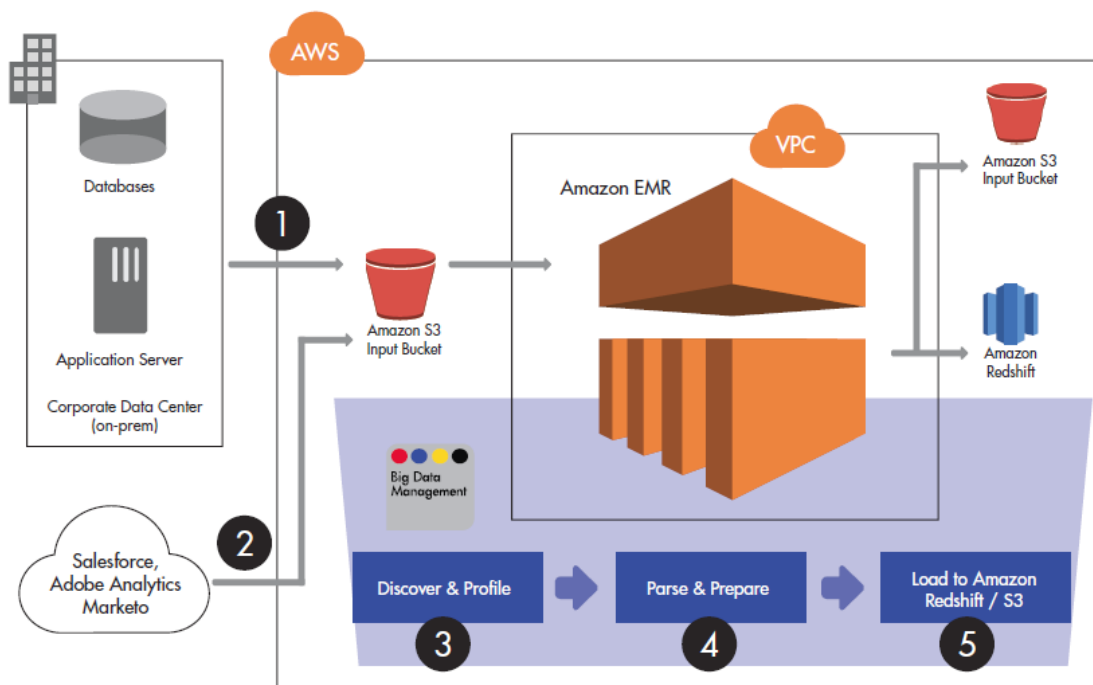
After you set up auto-scaling, you can set up CloudWatch dashboards to display real-time graphs based on the auto-scaling rules that you configured.

Data Lake Use Case

You can leverage the cloud deployment of Big Data Management on Amazon EMR to implement a data lake solution.

When you have data in different repositories owned by a wide spectrum of stakeholders, you can import the data from disparate sources to a data lake to enable access to all the users in your organization.

The following image shows how you can implement a data lake solution using Big Data Management and Amazon EMR, Amazon S3, and Amazon Redshift.



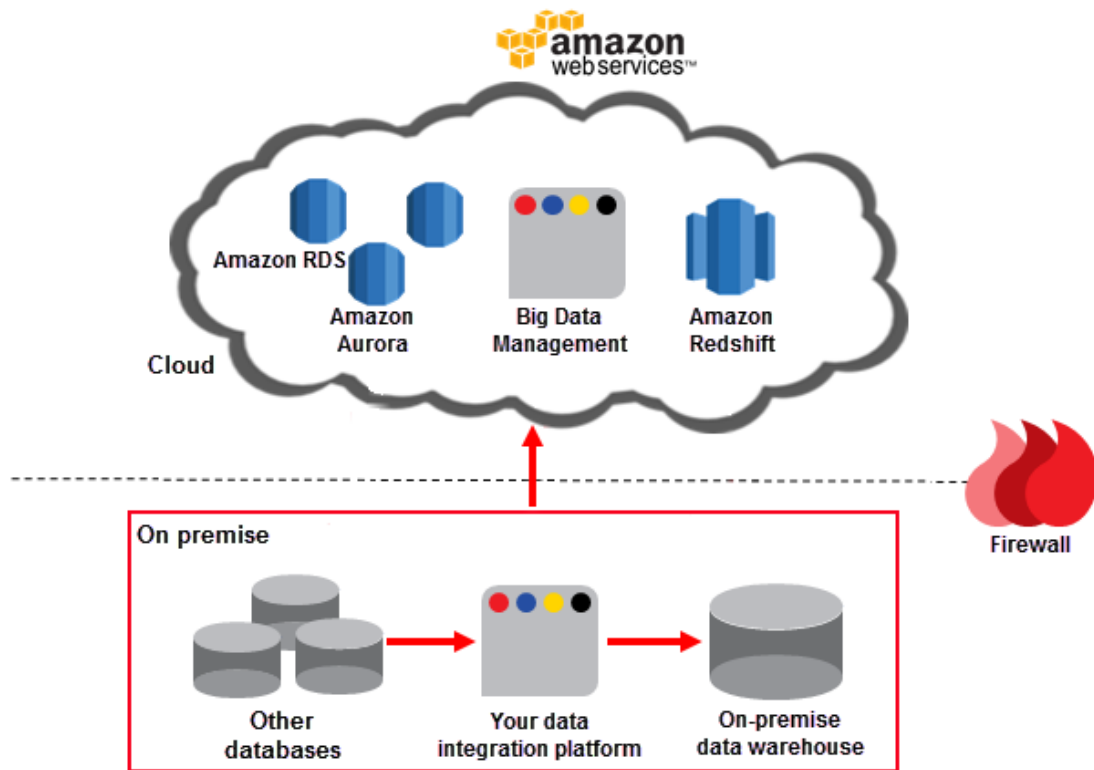
1. Offload infrequently used data and batch load raw data to a defined landing zone in an Amazon S3 bucket. For a data lake, you load raw transactional and multi-structured data directly to Amazon S3, freeing up space in the current Enterprise Data Warehouse..
2. Collect and stream data generated by machines and sensors, including application and weblog files, directly to Amazon S3 instead of staging it in a temporary file system or the data warehouse.
3. Discover and profile data stored on Amazon S3. Profile data to better understand its structure and context. Adding requirements for enterprise accountability, control, and governance for compliance with corporate and governmental regulations and business service level agreements.
4. Parse and prepare data from weblogs, application server logs or sensor data. Typically, these data types are either in multi-structured or unstructured formats which can be parsed to extract features and entities, and data quality techniques can be applied. You can execute pre-built transformations and data quality and matching rules natively in Amazon EMR to prepare data for analysis.
5. After you cleanse and transform data on Amazon EMR, move high-value curated data from EMR to an Amazon S3 output bucket or to Amazon Redshift. From there, users can directly access data with BI reports and applications.

In this process, Amazon EMR does not copy the data to local disk or HDFS. Instead, mappings open multithreaded HTTP connections to Amazon S3, pull data to the Amazon EMR cluster, and process data in streams.

Lift and Shift Use Case

You can move data and processing from an existing on-premise big data solution to the Amazon EMR environment to improve processing speed.

The following image shows this solution:



In this solution, you perform the following tasks:

- Move the contents of on-premise databases to Amazon AWS resources like Amazon Aurora and Amazon RDS.
- Move an on-premise data warehouse to Amazon Redshift.
- Move the on-premise implementation of Informatica to Big Data Management on Amazon EMR.

You can use Informatica connectors for Amazon Redshift and Amazon S3 to move data.

Best Practices

Informatica recommends the following best practices for implementing Big Data Management on the AWS cloud.

Guidelines for Selecting EC2 Instances for the EMR Cluster

When provisioning Big Data Management on the EMR cluster, you choose from available Amazon EC2 instance types for the core and task nodes. These instance types have varying combinations of CPU, memory, storage and networking capacities. Wide selection of Amazon EC2 instance types make choosing the right instance type for the EMR clusters challenging.

Consider the following factors when selecting EC2 instances:

- EC2 instance types
- Workload types and use cases
- Storage types
- Cluster node types

Following a short discussion of these categories of EC2 architecture and mapping types, this section recommends EC2 instance types for each workload type.

EC2 Instance Types

Amazon refers to EC2 instance types by names that represent categories and sizes. For example, the available instances in the "M series" are m1.small, m1.medium, m1.large, and so on. The available instances in the "C series" are c1.medium, c1.xlarge, and so on.

Each type corresponds to an instance configured with a default amount of memory, number of cores, storage capacity and type of storage and other characteristics, along with a price tag representing the cost per hour to use the instance.

Informatica requires at least minimum of 8 CPU VCores and 30 GB memory for the product to function. This minimal configuration is intended for demo scenarios. The m4.2xlarge instance is appropriate.

For production scenarios, the recommended minimum resource requirements is 16 CPU VCores with at least 30 GB memory. The c3.4xlarge instance is appropriate.

For larger workloads, Informatica recommends a minimum of 32 CPU VCores and 60 GB of memory. The c3.8xlarge or m4.10xlarge instances are appropriate. You might also consider instances from the new generation, compute-optimized C4 series.

Mapping Workload Types and Use Cases

Mappings can be categorized using the following workload types:

CPU-bound

A mapping that is limited by the power of the EC2 instance's CPU speed is CPU-bound. For example, a mapping that includes pass-through components, expression evaluations, log parsing, and other use cases is called CPU-bound.

I/O-bound

A mapping that is limited by the network's I/O speed is I/O-bound. For example, a mapping that includes aggregations, joins, sorting and ranking, and other components is called I/O-bound.

Mixed

A mixed-type mapping has a combination of CPU-bound and I/O-bound characteristics. For example, a mapping that has a combination of expression functions and cache-based transformations.

Types of Cluster Nodes

Cluster nodes can be categorized using the following types:

Master node

The master node manages the cluster, distributes tasks to core and task nodes, monitors tasks, and monitors cluster health.

Slave node

Slave nodes can be one of the following types:

- Core nodes. Core nodes host the persistent data on Hadoop Distributed File System (HDFS) and run Hadoop tasks. Core nodes should be reserved for the capacity that is required until your cluster completes.
- Task nodes. Task nodes do not have a persistent data store. They only run Hadoop tasks.

When you create the cluster, AWS chooses one of the EC2 instances as the master node. You can designate the number of core nodes and task nodes as part of the cluster configuration.

Storage Types

Informatica tested the following types of elastic block storage (EBS) volumes:

Throughput-optimized HDD

Throughput-optimized hard disk drives (HDD) are inexpensive magnetic storage, optimized for frequently-accessed, throughput-intensive workloads. Informatica tests showed throughput of up to 160 MB/sec.

This volume type is a good fit for large, sequential workloads such as Amazon EMR, ETL, data warehouses, and log processing. Recommended if minimizing cost is an important concern.

Transformation-specific tests showed that HDD is 8-10% slower than general purpose SSD.

General purpose SSD

General-purpose solid state drives (SSD), also known as GP2 volumes, are intended to balance low cost and performance for a variety of transactional workloads. GP2 volumes showed throughput of up to 200 MB/sec for sequential read and write operations. Recommended for best performance for most batch use cases.

Provisioned IOPS SSD

Provisioned input-output operations-per-second SSD, also known as IO1 drives, are intended for mission-critical applications that require high performance. IO1 volumes support up to 20,000 IOPS and throughput of 330 MB/sec for sequential read and write operations. They are the fastest option and tend to be the most expensive.

Recommendations for Choosing EC2 Instances

The following table contains recommendations for each workload type, supported by Informatica performance testing:

Workload Characteristics	EC2 Instance Type Recommendation
CPU-bound mapping and pass-through mapping	Use C series instances for task nodes. Task nodes are cheaper than core nodes. Note: These task nodes do not have a persistent data store.
I/O-bound mapping	For processing a low volume of data up to 5TB, use core nodes with default storage. For example, the d2.2xlarge instance has default storage of 6x2000 HDD. In general, HDD storage is appropriate for I/O-bound mapping workloads. For a large volume of data (10 TB or higher), add additional EBS HDD volumes to core nodes.
Mixed load	Use core nodes with additional EBS HDD and for computational needs dynamically add task nodes to meet your cluster's varying capacity requirements. Note: General-purpose SSD are faster than HDD but more expensive.

Cluster Sizing Guidelines

Consider three categories of EMR deployments, separated by operational data volume into Sandbox, Small-Medium, and Large.

The following table contains guidelines for sizing various elements of the EMR deployment:

Deployment Element	Sandbox Deployment	Small-Medium Deployment	Large Deployment
Storage Type	HDD optimized	HDD optimized	HDD optimized
Number of EBS Volumes Per Node	2	2-4	6-8
EBS Volume Size for HDFS	100 GB	100-250 GB	250-500 GB

Deployment Element	Sandbox Deployment	Small-Medium Deployment	Large Deployment
Total HDFS Capacity Per Node	200 GB	200-1000 GB	1.5-4.0 TB
Replication Factor	2	2	3
YARN VCores Per Node ¹	14	14-30	36
YARN Memory Per Node ²	28 GB	54 GB	144 GB
Total Operational Data Volume	10 GB	100-500 GB	>1000 GB
Recommended Instance Types	M4.4xlarge C3.4xlarge	C3.8xlarge M4.4xlarge	M4.10xlarge
Recommended Minimum Number of Core Nodes	2	5	7

¹ yarn.nodemanager.resource.cpu-vcores

² yarn.nodemanager.resource.memory-db

Guidelines and Recommendations for Utilizing Clusters and Storage

Amazon EMR has two cluster types: transient and persistent.

Recommendation for Cluster Architecture

Transient or ephemeral clusters load input data, process the data, store the output results into a persistent data store, and then automatically shut down. Persistent clusters continue to run even after data processing is complete.

The following qualities characterize each cluster type:

Transient clusters

Launch transient or ephemeral clusters for data processing only, then transfer mapping results to S3 storage. Use a script to launch transient clusters for mapping runs and terminate them when data transfer is complete.

For more information, see ["Ephemeral Clusters" on page 17](#).

Persistent clusters

Persistent clusters are always available for processing or storing data that requires quick access. Each cluster node is charged by the second, so costs accumulate quickly

Recommendation: Informatica recommends transient cluster architecture, in which you retain data on a store like S3 or Redshift, and use the EMR cluster only for processing.

Guidelines For Using S3 Storage

Consider the following guidelines for utilizing S3 storage:

- Avoid uploading many small files. Instead, upload a smaller number of larger files.
- Reducing the number of files stored on Amazon S3 or on HDFS provides better performance when a mapping processes data on Amazon EMR.
- Use data compression for data sources and targets. Data compression helps reduce S3 storage costs and bandwidth costs.

- Data partitioning helps data optimization, allows the creation of unique buckets of data, and eliminates the need for a data processing job to read the entire data set.
- Use the AWS multipart upload feature (`--multipart-chunk-size-mb`) to upload larger files (>100 MB) to S3. The default chunk size is 15MB, the minimum allowed chunk size is 5MB, and the maximum is 5GB. Using this feature, Informatica testing showed an improvement of 10-12% when uploading a 77GB file to an S3 bucket.

Performance Best Practices

To achieve the best performance for Big Data Management on the AWS cloud, implement the following practices:

- Create the Informatica domain on an EC2 instances in the same region as the EMR cluster.
- Allocate 90% of CPU vCores and memory in `yarn-site.xml` when spawning an EMR cluster.
For example, for an instance type of `m4.4xLarge`, with 32 vCores (90% = 29 vCores) and 64 GB memory (90% = ~58 GB):

```
[{"Classification": "yarn-site", "Properties": {"yarn.nodemanager.resource.cpu-vcores": "32", "yarn.nodemanager.resource.memory-mb": "58000", "yarn.scheduler.maximum-allocation-mb": "16384", "yarn.scheduler.minimum-allocation-mb": "256", "yarn.nodemanager.vmem-check-enabled": "false"}, "Configurations": []}]
```
- Because HDFS data durability is not guaranteed, always use S3 buckets as persistent data storage.
- With data residing in S3 buckets, the EMR cluster can be terminated after the job is completed, providing significant cost savings.
- Locate S3 storage in the same region as that of the EMR cluster. Cross-region access is 1.5x to 4x slower. To write to an S3 bucket located in another region, enable [cross region replication for S3 buckets](#).
- If writing to an S3 bucket is slow, use a data copying utility like `S3DistCp` to move data from HDFS to S3.
- Spark shuffle service is enabled by default if Spark is added as an application during EMR cluster creation.
- To run Spark jobs, enable dynamic allocation parameter in `hadoopEnv.properties` in the following path on the Data Integration Service node in EC2: `$INFA_HOME/services/shared/hadoop/<Hadoop distribution>/infaConf`
- For large volume data processing, set device ring parameters for EMR core nodes to max level. The default setting for Rx is 512 and Tx is 1024.

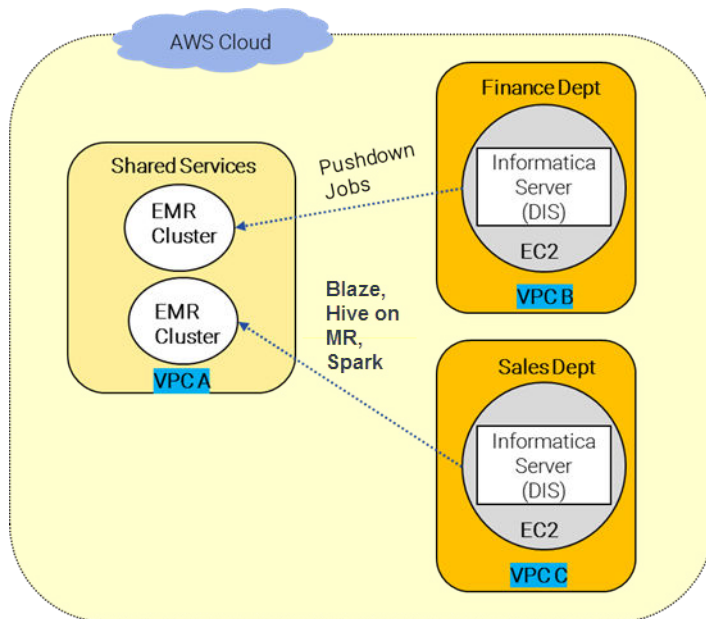
Using VPC to Access Resources on Multiple Clusters

Amazon Virtual Private Cloud (Amazon VPC) enables the provision of a logically isolated section of the Amazon Web Services (AWS) cloud. In this location, you can launch AWS resources in a virtual network that you define. Network communication via private IPs is only facilitated through VPC peering.

VPC peering allows a user of one VPC to access the resources of another VPC. For example, the Informatica domain is installed on VPC-A, while the EMR cluster is on VPC-B. When you enable VPC peering, you enable access to the EMR cluster from the Informatica domain.

Enable VPC peering when the Informatica domain and the Data Integration Service are on not on the same Virtual Private Cloud (VPC).

The following illustration shows Informatica services on different VPCs submitting various kinds of jobs to a shared EMR cluster on VPC-A:



Based on internal tests, VPC peering within the same region adds no performance overhead to mapping runs.

To enable VPC peering, see the Amazon AWS Peering Guide at the following URL:
<http://docs.aws.amazon.com/AmazonVPC/latest/PeeringGuide/Welcome.html>.

Note: Cross-region VPC pairing is not supported.

Case Studies

This section contains the results of performance testing using various cluster deployments, data storage types, and other factors.

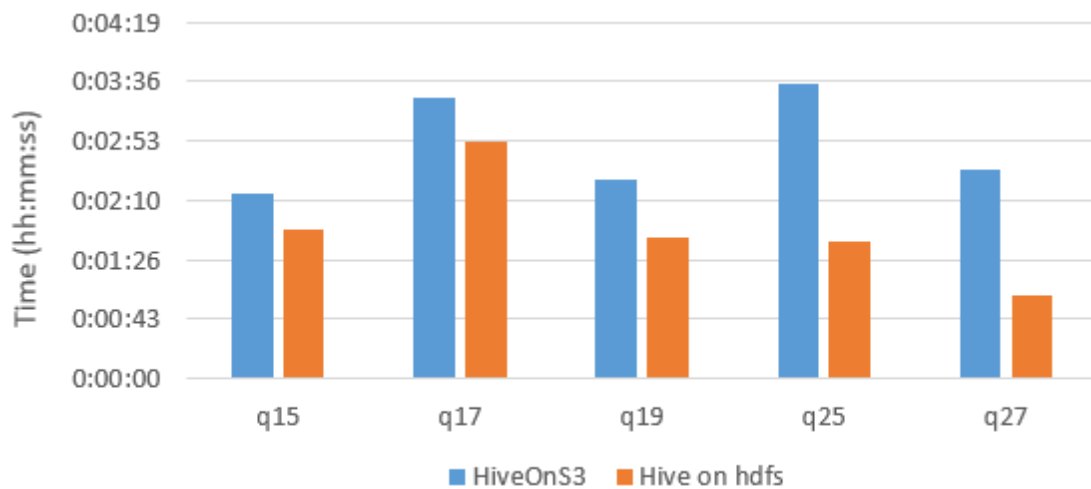
The following table shows the EC2 instance types used in the test environment:

EC2 Instance Type	vCPU	Memory (GB)
m4.4xlarge	16	64
c3.4xlarge	16	30
c3.8xlarge	32	60
m4.10xlarge	40	160

Hive on S3 Versus Hive on HDFS

This test compared access performance for a TPC-DS query against two storage types: Hive on S3, and Hive on HDFS.

The following chart compares access performance:



The chart shows that a TPC-DS query using Hive on S3 as a source or target is 20-150% slower than the same query using Hive on HDFS as a source or target.

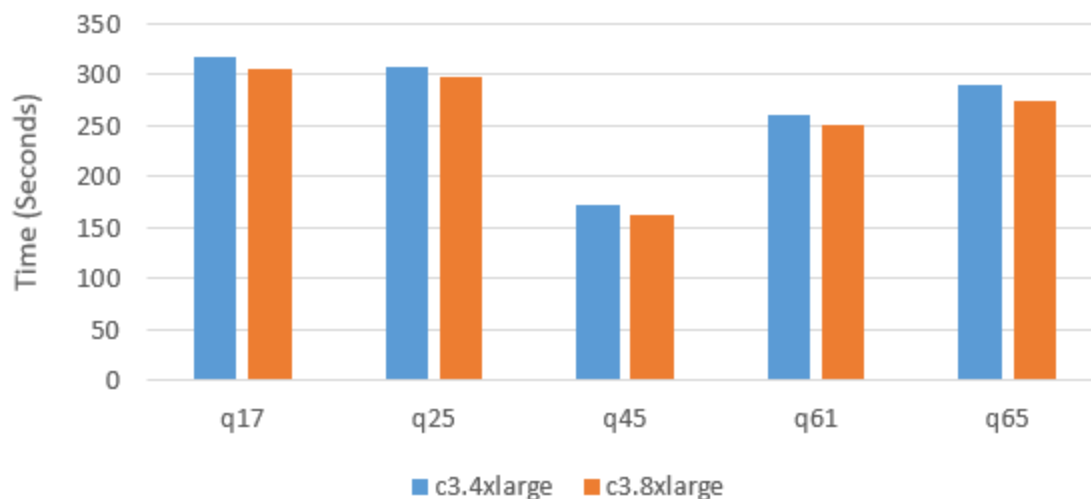
If you use an ephemeral cluster strategy, EMR clusters terminate after jobs complete, and the data cached on the cluster is lost. Data is also lost in case a cluster terminates unexpectedly. For these reasons, Informatica recommends S3 storage for sources and targets to avoid data loss.

Mappings using the Blaze run-time engine ran on a cluster using EC2 instances of type m4.10xlarge with 13-core nodes and a data volume scale factor of 500 GB.

Spark Queries on Different EC2 Instance Sizes

This test compared the performance of various TPC-DS queries on two EC2 instance sizes.

The following chart compares query performance:



The chart shows that a c3.8xlarge EC2 instance, with 60 GB memory, performs better than a c3.4xlarge instance with 30 GB memory.

Mappings using the Spark runtime engine ran on EC2 instances with 13-core nodes and a data volume scale factor of 500 GB.

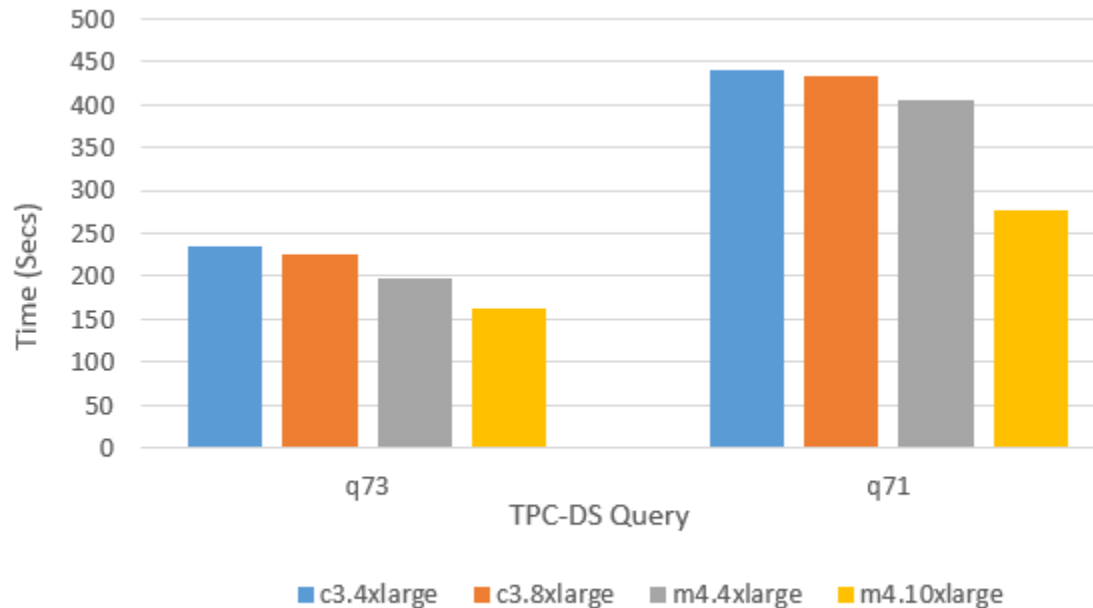
Spark Query Performance on Various EC2 Instance Types

This test compared the performance of TPC-DS queries on four different EC2 instance types.

The test measured the performance of two different queries:

- q71 is a complex mapping containing 8 sources, 5 joiners, an aggregator, and 5 filters.
- q73 is a simpler mapping containing 3 sources, 2 joiners, an aggregator, and 3 filters.

The following chart compares the performance of each query using the Spark run-time engine on four different EC2 instance types:



The chart shows that Query 71 ran up to 60% faster on the m4.10xlarge instance than on the other instances, while the performance of Query 73 was roughly the same across instance types. The results suggest that more complex queries get a greater benefit when running on large EC2 instances.

Mappings using the Spark runtime engine ran on EC2 instances with 13-core nodes and a data volume scale factor of 500 GB.

For More Information

You can access more information about Big Data Management 10.2 on AWS in the following resources:

[One-Click Deployment Process Video](#)

You can watch a video of the one-click deployment process. Go to this link on the Informatica Network to watch the video: [Big Data Management on the Amazon Cloud](#).

[Disaster Recovery](#)

Read an article on the Informatica Network about high availability and disaster recovery for Big Data Management 10.2 on Amazon AWS:

[Implementing a Disaster Recovery Strategy for Informatica Big Data Management 10.2 on Amazon AWS](#).

Appendix: Bootstrap Script Example

The following text is an example of the bootstrap script that you can use as part of the script to create a cluster.

```
#!/bin/bash
echo s3 location of RPM
export S3_LOCATION_RPM=s3://<s3 bucket name>
echo Temp location to extract the RPM export TEMP_DIR=/tmp/<TEMP-DIR-TO-EXTRACT-RPM>
echo Default location to install Informatica RPM
#make sure that INFA_RPM_INSTALL_HOME will have enough space to install the Informatica RPM
export INFA_RPM_INSTALL_HOME=/opt/
echo Extracting the prefix part from the rpm file name
echo The rpm installer name would be InformaticaHadoop-10.1.1.Linux-x64.tar.gz
export INFA_RPM_FILE_PREFIX=InformaticaHadoop-10.1.1.Linux-x64 export
INFA_RPM_FOLDER=InformaticaHadoop-10.1.1-1.231
echo S3_LOCATION_RPM = $S3_LOCATION_RPM
echo TEMP_DIR = $TEMP_DIR
echo INFA_RPM_INSTALL_HOME = $INFA_RPM_INSTALL_HOME echo INFA_RPM_FILE_PREFIX =
$INFA_RPM_FILE_PREFIX
echo Installing the RPM: echo "Creating temporary folder for rpm extraction"
sudo mkdir -p $TEMP_DIR cd $TEMP_DIR/
echo "current directory =" $(pwd)
echo Getting RPM installer
echo Copying the rpm installer $S3_LOCATION_RPM/$INFA_RPM_FILE_PREFIX.tar.gz to $(pwd) sudo aws
s3 cp $S3_LOCATION_RPM/$INFA_RPM_FILE_PREFIX.tar.gz . sudo tar -zxvf
$INFA_RPM_FILE_PREFIX.tar.gz
cd $INFA_RPM_FOLDER
echo Installing RPM to $INFA_RPM_INSTALL_HOME
sudo rpm -ivh --replacefiles --replacepkgs InformaticaHadoop-10.1.1-1.x86_64.rpm --prefix=
$INFA_RPM_INSTALL_HOME # You can insert additional tasks at this point in the script.
echo Contents of $INFA_RPM_INSTALL_HOME echo $(ls $INFA_RPM_INSTALL_HOME)
echo chmod cd $INFA_RPM_INSTALL_HOME sudo mkdir Informatica/blazeLogs sudo chmod 766 -R
Informatica/blazeLogs/ echo removing temporary folder sudo rm -rf $TEMP_DIR/ echo done
```

Authors

Amit Kara

Product Manager

Mark Pritchard

Principal Technical Writer