Implementing Big Data Management 10.1.1 with Ephemeral Clusters in a MS Azure Cloud Environment

Abstract

You can take advantage of cloud computing efficiencies and power by deploying the Informatica Big Data Management solution in the Microsoft Azure 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 Azure HDInsight environment to improve processing speed. This article describes how to implement the one-click Big Data Management deployment in the Azure environment.

Supported Versions

- Informatica Big Data Management 10.1.1 Update 2

Table of Contents

Overview

Customers of Microsoft Azure and Informatica can deploy Big Data Management in the Azure public cloud.

Use the Azure Marketplace to install a new instance of Big Data Management in the same virtual network or vnet where you run an HDInsight cluster.

When you install Big Data Management from the Azure Marketplace, you use your existing HDInsight cluster. You choose the size of the machine for the Informatica domain database.

After you complete installation, use Big Data Management to run mappings and workflows on the HDInsight cluster.

Using Big Data Management on Azure 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 Azure 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 the Azure 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 Azure, 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.

Pre-Implementation Tasks

Before you provision Azure resources and configure Big Data Management in the Microsoft Azure cloud environment, verify the following prerequisites:

Prerequisites

- You have purchased a license for Big Data Management.
  The license file has a name like `BDMLicense.key`. During configuration, you browse to this file on your local system. It is not necessary to upload it to the Azure environment.
- You have an Azure subscription and its account authentication information.
- You have created an HDInsight WASB cluster in the Azure environment.

Note: When you provision Big Data Management, you must use the virtual network or vnet where the current instance of HDInsight is configured.

Gather Cluster Information

Gather the following information from the cluster, and input the information during configuration of the Big Data Management instance:

- Ambari host machine, port number, and user authentication details
- SSH port and user authentication details

Create Staging Directories on the Cluster

In the HDFS storage location, create the following directories on the cluster and set permissions to 777:

- - Blazeworkingdir

  - SPARK_HDFS_STAGING_DIR

  - SPARK_EVENTLOG_DIR

Provision Azure Resources and the Informatica Domain

Use the Azure Marketplace website to provision Azure cluster resources including a Big Data Management deployment.

1. Select Big Data Management for setup.
   a. In the Azure marketplace, click the + button to create a new resource.
   b. Search on "Informatica" to find Informatica offerings in the Azure marketplace.
   c. Select **Big Data Management 10.1.1 U2 BYOL.**

      The Create Big Data Management Enterprise Edition tab opens. It displays all the steps necessary to configure and launch Big Data Management on an HDInsight cluster.
2. Supply information in the **Basics** panel, and then click **OK**.

   Subscription

      Select the Azure subscription account that you want to use for Big Data Management.

Charges for this instance of Big Data Management will go to this subscription.

Resource Group

Select a resource group to contain the BDM implementation.

Usually, you select an existing resource group where you have a running HDInsight cluster.

Location

Location of the resource group.

Accept the location that is already associated with the resource group.

3. Supply information in the **Informatica Domain Settings** panel, and then click **OK**.

   This tab allows you to configure additional details of the Informatica domain. All properties in this tab are mandatory.

   Informatica Domain Name

   Create a name for the Informatica domain.

   Informatica domain administrator name

   Login to use to administer the Informatica domain.

   Password

   Password for the Informatica administrator.

   Keyphrase for encryption key

   Create a keyphrase to create an encryption key.

   Informatica license file

   Click the Folder icon to browse to the location of the Informatica license file on your local system.

   When you select the license file and click OK, Azure uploads the file.

4. Supply information in the **Node Settings** panel, and then click **OK**.

   This tab allows you to configure details of the Informatica domain.

   Select the OS for the VM.

   Select **Red Hat Enterprise Linux 7.3**.

   Number of nodes in the domain.

   Default is 1.

   You can configure up to 29 nodes.

   Machine prefix

   Type an alphanumeric string that will be a prefix on the name of each virtual machine in the Informatica domain.

   For example, if you use the prefix "infa" then Azure will identify virtual machines in the domain with this string at the beginning of the name.

   VM Username

   Username that you use to log in to the virtual machine that hosts the Informatica domain.

   Authentication type

   Authentication protocol you use to communicate with the Informatica domain.

   Default is **Password.**

Password

Password to use to log in to the virtual machine that hosts the Informatica domain.

Machine size

Select from among the available preconfigured VMs. The default is 1x Standard DS11.

5. Supply information in the **Database Settings** panel, and then click **OK**.

   This tab allows you to configure settings for the storage where Informatica metadata will be stored.

   Database type

   Select **SQL Server 2014.**

   Database machine name

   Name for the virtual machine that hosts the domain database.

   Username

   Username for the administrator of the virtual machine host of the database.

   These credentials to log into the virtual machine where the database is hosted.

   Password

   Password for the database machine administrator.

   Database machine size

   Select a size from among the available preconfigured virtual machines. The default is 1x Standard DS3.

   Informatica Domain DB User

   Name of the database user.

   The Informatica domain uses this account to communicate with the domain database.

   Informatica Domain DB Password

   Password for the database user.

6. Supply information in the **Informatica Big Data Management Configuration** panel, and then click OK.

   This tab allows you to configure credentials that allow the Informatica domain to communicate with the HDInsight cluster. Get the information for these settings from HDInsight cluster settings panels and the Ambari cluster management tool.

   HDInsight Cluster Hostname

   Name of the HDInsight cluster where you want to create the Informatica domain.

   HDInsight Cluster Login Username

   User login for the cluster. This is usually the same login you use to log in to the Ambari cluster management tool.

   Password

   Password for the HDInsight cluster user.

   HDInsight Cluster SSH Hostname

   SSH name for the cluster.

   HDInsight Cluster SSH Username

   Account name you use to log in to the cluster head node.

Password

Password to access the cluster SSH host.

Ambari port

Port to access the Ambari cluster management web page. Default is 443.

7. Supply information in the **BDM Services Settings** panel, and then click **OK**.

   Use this tab to set up Informatica services. For more information about Informatica services, see the *Informatica 10.1.x Application Service Guide*.

   Model Repository Service Name (MRS)

   Name of the Model Repository Service.

   The Model Repository Service manages metadata about mappings, workflows and applications.

   Data Integration Service Name (DIS)

   Name of the Data Integration Service.

   The Data Integration Service runs mappings and workflows.

   MRS Database username

   Username for the Model Repository Service database.

   The Model Repository Service database stores Model Repository data.

   MRS Database password

   Password for the Model Repository Service database.

8. Supply information in the **BDM Connection Settings** panel, and then click **OK**.

   Use this tab to configure connections between the Informatica domain and the HDInsight cluster. The Data Integration Service uses the connections to run mappings on the cluster.
   The HDFS User Name property is mandatory. The other properties in this tab are optional.

   For more information about connections, see the Connections appendix of the *Informatica Big Data Management User Guide.*

   HDFS User Name

   User name for the HDFS connection.

   HBase Zookeeper Hosts

   List HBase Zookeeper hosts, separated by the comma (,) character.

   Hive User Name

   User name for the Hive connection.

   Select Hive Execution Mode

   Select the Hive execution mode. Default is Remote.

   Hadoop Impersonation User Name

   User name of the impersonation user.

   Note: The Data Integration Service uses this user name to execute mappings on a secure cluster.

   Hadoop Blaze Working Directory

   Path to the Blaze working directory on the HDInsight cluster.

   For example,

   ```
   /blaze/workdir
   ```

Hadoop Blaze Service User Name

User name for the Blaze engine service on the HDInsight cluster.

Hadoop Spark HDFS Staging Directory

Path to the Spark HDFS staging directory on the HDInsight cluster.

For example,

```
/tmp/sparkdir
```

Hadoop Spark Event Log Directory

Path to the Spark HDFS event log directory on the HDInsight cluster.

Hadoop Spark Execution Parameters

Optional. Run-time parameters that override Spark parameters on the cluster. You can enter a list separated by the characters "&:" (ampersand colon).

For example, when you want to set run-time parameters for the spark.driver.memory, spark.executor.memory, and spark.executor.cores properties, enter a string like:

```
Spark.driver.memory=1G&:spark.executor.memory=4G&:spark.executor.cores=2
```

9. Supply information in the **Infrastructure Settings** panel, and then click **OK**.

   Use this tab to set up cluster resources for the Big Data Management implementation.

   Storage account

   Storage resource that the virtual machines that run the Big Data Management implementation will use for data storage.

   Select from the existing storage accounts.

   Virtual network

   Virtual network for the Big Data Management implementation to belong to. Select the same network as the one that you used to create the HDInsight cluster.

   Subnets

   The subnet that the virtual network contains.

   Choose from among the subnets that are available in the virtual network.

10. Verify the choices in the **Summary** panel, and then click **OK**.
11. Read the terms of use in the **Buy** panel, and then click **Create**.

    When you click **Create**, Azure deploys Big Data Management and creates resources in the environment that you configured.

Logs and Results Files

The installer creates the following logs and results files:

- Informatica installation logs. Location: `/home/<operating system user name>/Informatica/10.1.1`

  - `Informatica_10.1.1_InstallLog.log`

  - `Informatica_10.1.1_Services_<time stamp>.log`

- Big Data Management solution deployment log. Location: `/home/<operating system user name>/Oneclicksoluion_results.log`
- The Big Data Management installer runs from a Debian package. The log for this operation is in the following location on each node: `/home/<operating system user name>/debianresults`

Post-Implementation: Import Files for Data Decryption

If you configured the SSH public key user as a user of the Big Data Management instance, perform the following task when the Big Data Management installer completes.

The node that hosts the Data Integration Service requires the following files from the Hadoop cluster to decrypt data:

- key_decryption_cert.prv
- decrypt.sh

Perform the following steps to import the files:

1. Locate these files on the cluster head node.
2. Copy the files from the Hadoop cluster to the node that runs the Data Integration Service. The directory structure must be the same as that of the Hadoop cluster.
3. Verify that permissions on the directory are 775.

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 terminated.

You can implement an ephemeral cluster strategy in your Azure HDInsight implementation using an Azure automation account and runbooks.

To implement the strategy, you prepare the scripts with details about the cluster and how the Informatica domain can connect to it, then import the scripts and create runbooks within an Azure automation account. After you have prepared scripts and created runbooks, you can execute the ephemeral cluster strategy.

Note: The process in this section assumes you already have an Informatica domain installed in the Azure environment.

The following list of steps shows a typical process for using an ephemeral cluster:

Step 1. Configure and import the Informatica integration script.

This script contains the login and authentication properties and other information to configure integration of the Informatica domain with the cluster.

Step 2. Configure and execute a runbook to create the cluster and execute Informatica integration.

The cluster creation runbook specifies Azure account details and cluster configuration resources, and creates and starts cluster resources, including cluster nodes, storage resources, and credentials. Then the runbook executes the shell script to integrate the Informatica domain.

Step 3. Run mappings from the Developer tool.

A developer runs mappings that access cluster resources for data retrieval, processing and results caching.

Note: You cannot configure a runbook to run mappings through the Developer tool. However, it is possible to integrate cluster creation, mapping execution, and cluster termination tasks in a single workflow. For information about how to create and configure a workflow, see the *Informatica Developer Workflow Guide*.

Step 4. Configure and execute a runbook to terminate the cluster.

The runbook script terminates cluster nodes and unregisters cluster resources.

Prepare and Import Scripts and Create Runbooks

To implement an ephemeral cluster strategy, prepare the following scripts to automate cluster creation, Informatica domain installation and configuration, and cluster termination:

Informatica installation and configuration script

The script contains authentication credentials and cluster host location and authentication credentials. The script installs a Debian package that contains Big Data Management binaries on the cluster, creates staging directories so the cluster can process mappings, creates and configures connections between the domain and the cluster and performs other tasks.

After you populate script property values, import the runbook in the Azure automation account.

Cluster creation runbook

The runbook script authenticates the user, sets options for cluster resources, registers a storage account, creates credentials, and establishes a new cluster resource group consisting of a number of cluster nodes with designated size and type, OS version and type, and other details. It also calls the Informatica installation and configuration script.

After you populate script property values, import the script to a runbook.

Cluster termination runbook

The script terminates cluster nodes and unregisters cluster resources.

After you populate script property values, create the runbook and import the script.

Edit and Upload the hdireconfigure.sh Script

The hdireconfigure.sh script installs and configures Informatica on the HDInsight cluster.

The script contains authentication credentials and domain host location and authentication credentials. The script installs a Debian package that contains Big Data Management binaries on the cluster, creates staging directories so the cluster can process mappings, creates and configures connections between the domain and the cluster and performs other tasks.

Edit hdireconfigure.sh

To edit the script, substitute actual values for the placeholder values. For example, edit the following line:

```
HDIClusterName=${1}
```

to supply the name of the HDInsight cluster:

```
HDIClusterName=${TEST010101cluster01}
```

Follow Azure naming conventions for host names and user names.

Configure the following properties:

Property	Description
HDIClusterName	Name of the HDInsight cluster where the Informatica domain resides.
HDIClusterLoginUsername	User login for the cluster.
HDIClusterLoginPassword	Password for the HDInsight cluster user.
HDIClusterSSHHostname	Host name for the machine that uses SSH to connect to the cluster.

Property	Description
HDIClusterSSHUsername	Account name you use to log in to the SSH host.
HDIClusterSSHPassword	Password to access the cluster SSH host.
blazeworkingdir	Path to an HDFS directory on the cluster to cache intermediate results of a Blaze mapping run. For example, `/blaze/workdir`
SPARK_HDFS_STAGING_DIR	Path to an HDFS directory on the cluster to cache intermediate results of a Spark mapping run. For example, `/tmp/sparkdir`
SPARK_EVENTLOG_DIR	Path to the Spark HDFS event log directory on the HDInsight cluster.
osUserName	User name for the account to create an SSH connection to the domain host. The cluster uses this connection when it runs the configuration script.
osPassword	Password for the OS user.
domainHost	name of the host where the Informatica domain resides.
debianLocation	Location on the domain where the script will place the Debian package of Informatica binaries. For example: `debianlocation=/home/azuretestuser/InformaticaHadoop-10.1.1U2-Deb`

Upload the Script

Upload the edited script to a location on the domain host. Supply the path to this location in the ClusterConfigScriptPath variable when you configure the cluster creation runboook.

hdireconfigure.sh Template

Copy the following script template:

```
HDIClusterName=${1}
HDIClusterLoginUsername=${2}
HDIClusterLoginPassword=${3}
HDIClusterSSHHostname=${4}
HDIClusterSSHUsername=${5}
HDIClusterSSHPassword=${6}
blazeworkingdir=${7}
SPARK_HDFS_STAGING_DIR=${8}
SPARK_EVENTLOG_DIR=${9}
osUserName=${10}
osPwd=${11}
domainHost=${12}

debianlocation=/home/azuretestuser/InformaticaHadoop-10.1.1U2-Deb

getclusterdetails()
{
  echo "Getting list of hosts from ambari"
  hostsJson=$(curl -u $HDIClusterLoginUsername:$HDIClusterLoginPassword -X GET https://
$HDIClusterName.azurehdinsight.net/api/v1/clusters/$HDIClusterName/hosts)

  echo "Parsing list of hosts"
  hosts=$(echo $hostsJson | sed 's/\\\\\//\//g' | sed 's/[{}]//g' | awk -v k="text"
'{n=split($0,a,","); for (i=1; i<=n; i++) print a[i]}' | sed 's/\"\:\"/\|/g' | sed 's/[\,]/ /g'
| sed 's/\"//g' | grep -w 'host_name')
```

```
  echo $hosts

  #additional configurations required
  echo "Extracting headnode0"
  headnode0=$(echo $hosts | grep -Eo '\bhn0-([^[:space:]]*)\b')
  echo $headnode0
  echo "Extracting headnode0 IP addresses"
  headnode0ip=$(dig +short $headnode0)
  echo "headnode0 IP: $headnode0ip"
  resulthost=$(sshpass -p $HDIClusterSSHPassword ssh -o StrictHostKeyChecking=no 
$HDIClusterSSHUsername@$headnode0ip "uname -a | cut -d ' ' -f 2")
  echo "resulthost name is:"$resulthost

  #Add a new line to the end of hosts file
  echo "">>/etc/hosts
  echo "Adding headnode IP addresses"
  echo "$headnode0ip headnodehost $resulthost $headnode0">>/etc/hosts


  echo "Extracting workernode"
  workernodes=$(echo $hosts | grep -Eo '\bwn([^[:space:]]*)\b')
  echo "Extracting workernodes IP addresses"
  echo "workernodes : $workernodes"
  wnArr=$(echo $workernodes | tr "\n" "\n")
}

createstagingdir()
{
  sshpass -p $HDIClusterSSHPassword ssh -o StrictHostKeyChecking=no $HDIClusterSSHUsername@
$headnode0ip "sudo ln -f -s /bin/bash /bin/sh"
  sshpass -p $HDIClusterSSHPassword ssh -o StrictHostKeyChecking=no $HDIClusterSSHUsername@
$headnode0ip "sudo mkdir" $blazeworkingdir
  sshpass -p $HDIClusterSSHPassword ssh -o StrictHostKeyChecking=no $HDIClusterSSHUsername@
$headnode0ip "sudo chmod -R 777" $blazeworkingdir
  sshpass -p $HDIClusterSSHPassword ssh -o StrictHostKeyChecking=no $HDIClusterSSHUsername@
$headnode0ip "sudo mkdir" $SPARK_HDFS_STAGING_DIR
  sshpass -p $HDIClusterSSHPassword ssh -o StrictHostKeyChecking=no $HDIClusterSSHUsername@
$headnode0ip "sudo chmod -R 777" $SPARK_HDFS_STAGING_DIR
  sshpass -p $HDIClusterSSHPassword ssh -o StrictHostKeyChecking=no $HDIClusterSSHUsername@
$headnode0ip "sudo mkdir" $SPARK_EVENTLOG_DIR
  sshpass -p $HDIClusterSSHPassword ssh -o StrictHostKeyChecking=no $HDIClusterSSHUsername@
$headnode0ip "sudo chmod -R 777" $SPARK_EVENTLOG_DIR

}

createshellscript()
{

    shelltowrite="test.sh"

    echo "#!/bin/sh" > $shelltowrite
    echo "workernodeip=\$1">>$shelltowrite
    echo "HDIClusterSSHUsername=\$2">>$shelltowrite
    echo "HDIClusterSSHPassword=\$3">>$shelltowrite
    echo "sshpass -p \$HDIClusterSSHPassword ssh -q -o StrictHostKeyChecking=no \
$HDIClusterSSHUsername@\$workernodeip \"sudo mkdir ~/rpmtemp\"">>$shelltowrite
    echo "sshpass -p \$HDIClusterSSHPassword ssh -q -o StrictHostKeyChecking=no \
$HDIClusterSSHUsername@\$workernodeip \"sudo chmod 777 ~/rpmtemp\"">>$shelltowrite
    echo "echo \"copying Binaries to\" \$workernodeip">>$shelltowrite
    echo "sshpass -p \$HDIClusterSSHPassword scp -q -o StrictHostKeyChecking=no $debianlocation/
informatica_10.1.1U2-1.deb \$HDIClusterSSHUsername@\$workernodeip:\"~/rpmtemp/\"">>$shelltowrite
    echo "echo \"Installing Debian in\" \$workernodeip">>$shelltowrite
    echo "sshpass -p \$HDIClusterSSHPassword ssh -q -o StrictHostKeyChecking=no \
$HDIClusterSSHUsername@\$workernodeip \"sudo chmod -R 777 ~/rpmtemp\"">>$shelltowrite
    echo "sshpass -p \$HDIClusterSSHPassword ssh -q -o StrictHostKeyChecking=no \
$HDIClusterSSHUsername@\$workernodeip \"sudo dpkg --force-all -i ~/rpmtemp/
informatica_10.1.1U2-1.deb\"">>$shelltowrite
    echo "sshpass -p \$HDIClusterSSHPassword ssh -q -o StrictHostKeyChecking=no \
$HDIClusterSSHUsername@\$workernodeip \"sudo rm -rf ~/rpmtemp\"">>$shelltowrite
    echo "sshpass -p \$HDIClusterSSHPassword ssh -q -o StrictHostKeyChecking=no \
$HDIClusterSSHUsername@\$workernodeip \"sudo ln -f -s /bin/bash /bin/sh\"">>$shelltowrite
```

```
    echo "echo \"Debian Installation completed\"">>$shelltowrite
    chmod -R 777 $shelltowrite

}

installdebian()
{
  echo "Installing debian"
  for workernode in $wnArr
  do
    echo "[$workernode]"
    workernodeip=$(dig +short $workernode)
    echo "workernode $workernodeip"
       sudo sh  $shelltowrite $workernodeip $HDIClusterSSHUsername $HDIClusterSSHPassword >
$workernodeip.txt &

  done
  wait
  echo "out of wait"
  echo "Debian installation successful"

}

copyhelperfilesfromcluster()
{

#remove already existing authentication id of vm if any
remote_knownhostsfile="/home/"$HDIClusterSSHUsername"/.ssh/known_hosts"
sshpass -p $HDIClusterSSHPassword ssh -q -o StrictHostKeyChecking=no $HDIClusterSSHUsername@
$headnode0ip ""sudo ssh-keygen -f "$remote_knownhostsfile" -R " $domainHost"


echo "Installing sshpass on cluster"
sshpass -p $HDIClusterSSHPassword ssh -q -o StrictHostKeyChecking=no $HDIClusterSSHUsername@
$headnode0ip "sudo apt install sshpass "
echo "searching for file in remote cluster"
sshpass -p $HDIClusterSSHPassword ssh -q -o StrictHostKeyChecking=no $HDIClusterSSHUsername@
$headnode0ip "sudo find / -name decrypt.sh >oneclicksnap.txt"
sshpass -p $HDIClusterSSHPassword ssh -q -o StrictHostKeyChecking=no $HDIClusterSSHUsername@
$headnode0ip "sudo find / -name key_decryption_cert.prv >>oneclicksnap.txt"
sleep 5
echo "downloading oneclicksnap.txt"
echo sshpass -p $HDIClusterSSHPassword ssh -q -o StrictHostKeyChecking=no
$HDIClusterSSHUsername@$headnode0ip ""sshpass -p" $osPwd "scp -q -o StrictHostKeyChecking=no
oneclicksnap.txt "$osUserName"@"$domainHost":""/home/"$osUserName"

sshpass -p $HDIClusterSSHPassword ssh -q -o StrictHostKeyChecking=no $HDIClusterSSHUsername@
$headnode0ip ""sshpass -p" $osPwd "scp -q -o StrictHostKeyChecking=no oneclicksnap.txt
"$osUserName"@"$domainHost":""/home/"$osUserName"

echo "downloading done"
sleep 20

#code to iterate snap.txt and download the file and copy to it to local directory

counter=0
skipcount=2
filename="/home/"$osUserName"/oneclicksnap.txt"
echo "displaying the content of downloaded file"
cat $filename

echo "parsing and processing the file contents"

IFS=$'\n' read -d '' -r -a totalfiles < "$filename"

for line in "${totalfiles[@]}"
do
  name="$line"
  echo "downloading file:"$name

  sshpass -p $HDIClusterSSHPassword ssh -q -o StrictHostKeyChecking=no $HDIClusterSSHUsername@
```

```
$headnode0ip ""sshpass -p" $osPwd "scp -q -o StrictHostKeyChecking=no "$name
$osUserName"@"$domainHost":""~""

  IFS='/' read -ra NAMES <<< "$name"
  counter=${#NAMES[@]}
  ((chckcounter=$counter - $skipcount))
   #$basechkcounter=$chckcounter

  intermediatestring=""
  while [ $chckcounter -gt 0 ]
  do
    #echo ${NAMES[$chckcounter]}
    intermediatestring=${NAMES[$chckcounter]}/$intermediatestring
    ((chckcounter=$chckcounter - 1))
  done

  intermediatestring=/$intermediatestring
  #echo $intermediatestring
  #echo ${NAMES[(counter-1)]}
  echo "creating directory:"$intermediatestring
  mkdir -p $intermediatestring
  sleep 5
  echo "moving file:"${NAMES[(counter-1)]}
  mv /home/$osUserName/${NAMES[(counter-1)]} $intermediatestring
  chmod -R 777 $intermediatestring
done

  echo "Removing sshpass installation on cluster"
  sshpass -p $HDIClusterSSHPassword ssh -q -o StrictHostKeyChecking=no $HDIClusterSSHUsername@
$headnode0ip "sudo apt-get --purge remove sshpass --assume-yes"

}

fixforBDM7342()
{
   workernodehelperdir="/home/helper"
   for workernode in $wnArr
   do
    #statements
    workernodeip=$(dig +short $workernode)
    echo "creating directory in :"$workernode

    sshpass -p $HDIClusterSSHPassword ssh -q -o StrictHostKeyChecking=no $HDIClusterSSHUsername@
$workernodeip ""sudo mkdir "$workernodehelperdir"

    sshpass -p $HDIClusterSSHPassword ssh -q -o StrictHostKeyChecking=no $HDIClusterSSHUsername@
$workernodeip ""sudo ln -sf /etc/hadoop/conf/hdfs-site.xml "$workernodehelperdir"
    sshpass -p $HDIClusterSSHPassword ssh -q -o StrictHostKeyChecking=no $HDIClusterSSHUsername@
$workernodeip ""sudo ln -sf /etc/hadoop/conf/core-site.xml "$workernodehelperdir"
    sshpass -p $HDIClusterSSHPassword ssh -q -o StrictHostKeyChecking=no $HDIClusterSSHUsername@
$workernodeip ""sudo ln -sf /etc/hadoop/conf/mapred-site.xml "$workernodehelperdir"

   done
}

echo "Inside Main"
getclusterdetails
createstagingdir
createshellscript
installdebian
copyhelperfilesfromcluster
fixforBDM7342
```

Edit the Cluster Creation Script and Create the Runbook

Create the cluster creation runbook using a script to specify Azure account details and cluster resources.

Edit the Cluster Creation Script

The cluster creation script contains properties to create cluster resources. Copy the script template from the end of this topic.

Edit the script to specify values for the following properties.

Note: For more information about these properties, see the "Cluster Types and Configuration" section on the page https://docs.microsoft.com/en-us/azure/hdinsight/hdinsight-hadoop-provision-linux-clusters.

Azure Subscription Details

Identify the Azure subscription to use.

Property	Description
SubscriptionID	A string in `xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx` format that identifies the Azure subscription account to use.
Location	The region where you want to set up the cluster. The following page contains available regions: https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-manager-template-location

Previous HDI Cluster Infrastructure Details

Identify cluster infrastructure details of previously implemented clusters.

Property	Description
ResourceGroupName	Specify a resource group to contain the Big Data Management implementation.
ExistingStorageAccount	Storage account that the virtual machines that run the Big Data Management implementation will use to store data.
ExistingContainerName	Name of the container in the storage account to use.
VNetResourceGroupName	Virtual network group that contains the virtual network for the Big Data Management implementation to belong to. Use an existing VNet.
VNetName	Name of the virtual network where the Informatica domain is running.
SubnetName	One of the subnets that the virtual network contains.
ClusterConfigScriptPath	Location on the domain of the hdireconfigure.sh script.
ClusterName	Name of the cluster that was previously implemented.
SSHUserName	Account name you use to log in to the cluster head node.
SSHPassword	Password to access the cluster SSH host.

Property	Description
HTTPUserName	User name of the HTTP user.
HTTPPassword	Password for the HTTP user.

Cluster Node Details

Identify configuration details for each cluster node.

Property	Description
ClusterSizeInNodes	Number of cluster nodes to create.
ClusterVersion	HDInsight version to use in the cluster.
ClusterType	Type of the cluster. Select **Hadoop** from the drop-down list. Other cluster types are not supported.
ClusterOS	Enter "Linux".

Integration Shell Script Location

Perform the following steps to add the location of hdireconfigure.sh, the domain-cluster integration shell script:

1. At the bottom of the Create Runbook script file, find a line like:

   ```
   $CommandScriptBlock = [scriptblock]::Create("sh <path_to_start_script>.sh")
   ```

2. Insert the path to the hdireconfig.sh script file.

Save and Upload the Script

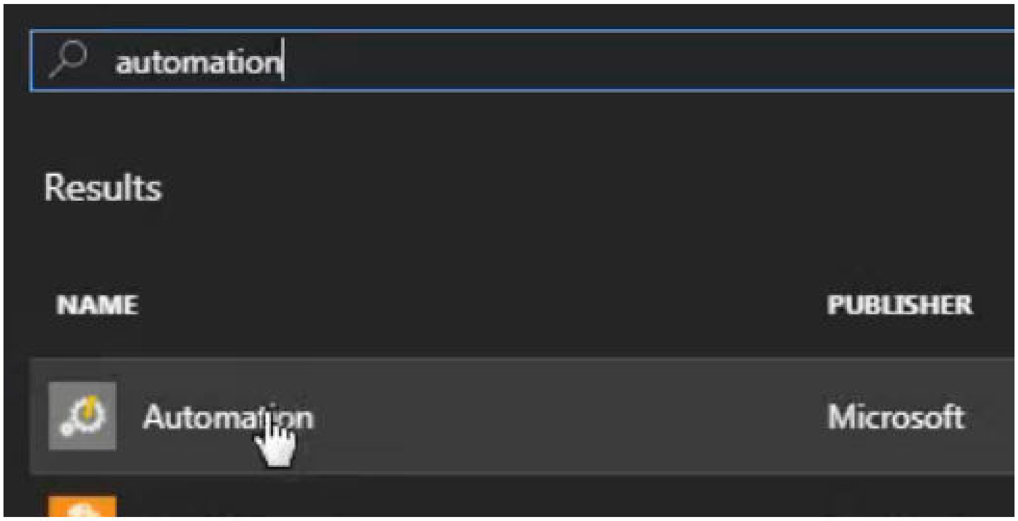
To make the script available for the cluster creation runbook, complete the following steps:

1. Save the Create Cluster script with a file suffix of `ps1`.
2. Upload the script to a location on the domain. Supply this path in step 4.d. below when you configure the cluster creation runbook.

Create the Runbook

To create the cluster creation runbook, log into the Azure portal and complete the following steps:

1. Select an existing automation account from your dashboard, or create an automation account through the following steps:
   a. From the dashboard, click **New**.
   b. Search the Marketplace for "automation."
   c. In the results, click **Automation** to create a new account.
   The image below shows Automation in the results list:



d. Click **Create**.

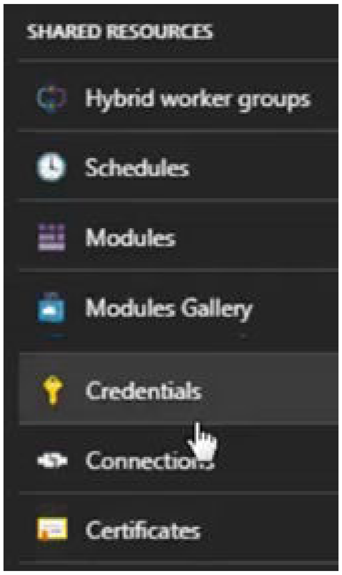
e. Enter values for the following properties:

- Name. Enter a name for the automation account.
- Subscription. Select a subscription to charge for the clusters that the runbook creates.
- Resource group. Select an existing resource group for the cluster to belong to, or create a new one.
- Location. Location where cluster resources will be created. Select from the drop-down list.
- Create Azure Run As account. Select **Yes**.

f. Click **Create**.

The automation account opens.

2. Select credentials for the user who will log in to the Azure portal, create the cluster, and run mappings. To select credentials, perform the following steps:
    a. In the **Shared Resources** area, click **Credentials**.
    The following image shows the Credentials choice:



    b. Select the credential that you use to log in to the Azure portal.
3. In the **Shared Resources** area, click **Modules** and import the following modules:
    - SSH
    - Azure.Storage
    - AzureRM.Profile
    - AzureRM.Storage
    - AzureRM.HDInsight
    - AzureRM.Network

    If any of the modules are already in the automation account, update them to the latest version.
4. Select a connection for the cluster to connect to the domain to run the hdireconfigure.sh script. To select the connection, perform the following steps:
    a. In the **Shared Resources** area, click **Connections**
    b. Select an existing connection, or click **Add a connection** and enter values for the following properties:
        - Name. Name for the connection.
        - Description. Optionally type a description.
        - Type. Select **SSH** from the drop-down list.
        - Computer Name. Network name or IP address of the Data Integration Service machine.
        - Port. Port that the Data Integration Service machine uses for an SSH connection.
        - User Name. User account on the Data Integration machine to use to connect to the cluster and run mappings.

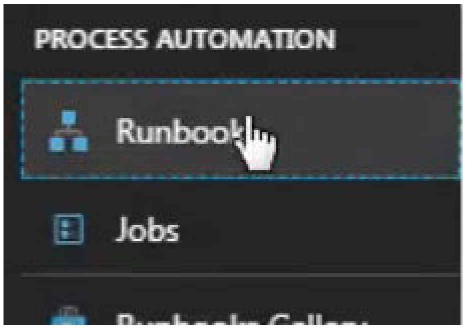
- Password. Password for the Data Integration Service user.

c. Click **Create** to create the connection.

The connection appears in the list of connections.

5. Perform the following steps to choose or create a runbook:

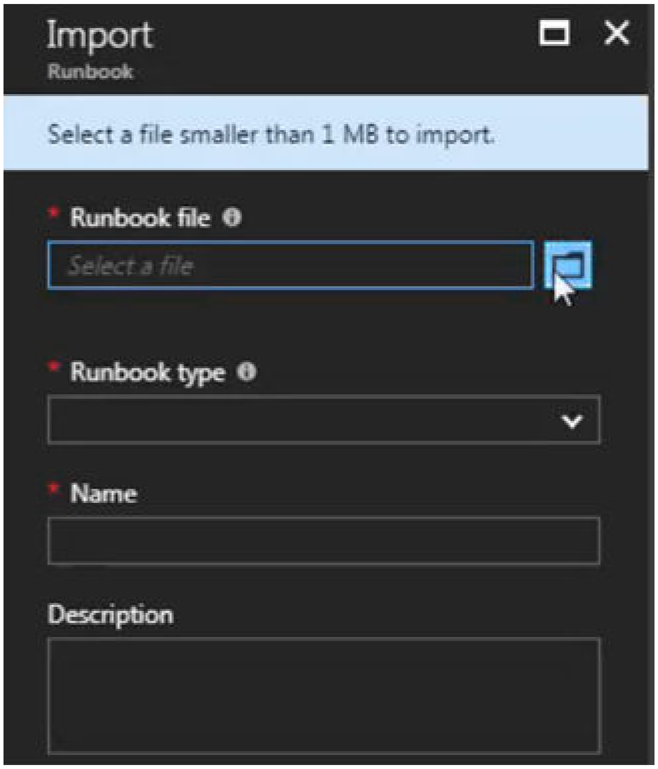
a. In the **Process Automation** area, click **Runbooks**.
The following image shows the Runbooks choice:



b. Select an existing runbook, or click **Add a runbook**.

c. Choose **Import an existing runboook**.
When you choose to import a runbook, you can load the runbook creation script that you prepared.

The Import Runbook dialog box opens:



Select a file smaller than 1 MB to import.

d. Click the **Browse Folder** icon and browse to the Create Cluster script that you uploaded to the Azure environment. Click **OK** to load it in the dialog box.

e. Click the Runbook type drop-down list and select **PowerShell**.

f. Type a name for the runbook.

g. Click **Create**.

The runbook appears on the list of runbooks. To run it, view its contents, or perform other actions, select the runbook in the list.

Create Cluster Runbook Script Template

Copy the following script template:

```
$cred =Get-AutomationPSCredential -Name ''

#azure subscription details
$subscriptionid=""
$location=""

#previous HDI cluster infrastructure details
$resourcegroupname=""
$exisitngstorageaccount=""
$existingcontainername=""

$vnetresourcegroupname=""
$vnetname=""
$subnetname=""


#previous HDI Cluster details
$clusterName=""
$sshusername="hdssh"
$sshpassword=ConvertTo-SecureString "" -AsPlainText -Force
$httpusername="admin"
$httppassword=ConvertTo-SecureString "" -AsPlainText -Force

#cluster node details
$clusterSizeInNodes = "4"
$clusterVersion = "3.5"
$clusterType = "HBase"
$clusterOS = "Linux"

#Login using credentials
Login-AzureRmAccount -Credential $cred

#get storage account details
$defaultStorageAccountKey = (Get-AzureRmStorageAccountKey -ResourceGroupName $resourceGroupName
-Name $exisitngstorageaccount)[0].Value
$defaultStorageContext = New-AzureStorageContext -StorageAccountName $exisitngstorageaccount -
StorageAccountKey $defaultStorageAccountKey

#create credential objects
$sshcred=new-object -typename System.Management.Automation.PSCredential -argumentlist
($sshusername, $sshpassword)
$httpcred=new-object -typename System.Management.Automation.PSCredential -argumentlist
($httpusername, $httppassword)


#Get exsisting vnet resource group details
$vnet = Get-AzureRmVirtualNetwork -Name $vnetName -ResourceGroupName $vnetresourcegroupname
$subnet = $vnet.Subnets | Where-Object {$_.Name -eq $subnetName}

#automation script starts from here

Select-AzureRmSubscription -SubscriptionId $subscriptionid
write-output "triggerring HDI creation"
#New-AzureRmResourceGroup -Name $resourcegroupname -Location $location
New-AzureRmHDInsightCluster -ResourceGroupName $resourceGroupName -ClusterName $clusterName -
```

```
Location $location -ClusterSizeInNodes $clusterSizeInNodes -ClusterType $clusterType -OSType
$clusterOS -Version $clusterVersion -HttpCredential $httpcred -DefaultStorageAccountName
"$exisitngstorageaccount.blob.core.windows.net" -DefaultStorageAccountKey
$defaultStorageAccountKey -DefaultStorageContainer $existingcontainername -SshCredential
$sshcred -VirtualNetworkId $vnet.Id -SubnetName $subnet.Id
write-output "created resource"
```

Edit the Terminate Cluster Script and Create the Runbook

Create the cluster termination runbook using a script to specify Azure account details and cluster resources.

Edit the Cluster Termination Script

The cluster termination script contains properties to identify cluster resources to stop and remove. Copy the script template from the end of this topic.

Edit the script to specify values for the following properties:

$cred

In the following line, insert the name of the cluster credential between the single quotes:

```
$cred =Get-AutomationPSCredential -Name '<credential name>'
```

In the remaining lines of the script, insert values for properties. All of the properties are defined in the topic "Edit the Cluster Creation Script and Create the Runbook."

Save the cluster termination runbook script as a text file.

Create the Runbook

Use the steps in the topic "Edit the Cluster Creation Script and Create the Runbook" as a guide to create a Terminate Cluster runbook.

The runbook appears on the list of runbooks. To run it, view its contents, or perform other actions, select the runbook in the list.

Cluster Termination Runbook Script Template

Copy the following script template:

```
$cred =Get-AutomationPSCredential -Name ''
$subscriptionid=""
$clusterName=""
$storagecontainername=""
$resourcegroupname=""
$exisitngstorageaccount=""

Login-AzureRmAccount -Credential $cred
Select-AzureRmSubscription -SubscriptionId $subscriptionid
write-output "triggering deletion"
Remove-AzureRmHDInsightCluster -ClusterName $clusterName

$defaultStorageAccountKey = (Get-AzureRmStorageAccountKey -ResourceGroupName $resourceGroupName
-Name $exisitngstorageaccount)[0].Value
$defaultStorageContext = New-AzureStorageContext -StorageAccountName $exisitngstorageaccount -
StorageAccountKey $defaultStorageAccountKey

Get-AzureStorageBlob -Context $defaultStorageContext -Container $storagecontainername | Remove-
AzureStorageBlob
#Remove-AzureStorageContainer -Name $storagecontainername  -Context $defaultStorageContext
write-output "deleted the cluster"
```

Execute Runbooks

To execute the ephemeral cluster strategy, perform the following steps:

1. Select the Cluster Creation runboook from the list of runbooks and click **Start**.
   Azure executes the runbook script, including the Domain-Cluster Integration script that it calls.
2. Run mappings on the HDInsight cluster from the Developer tool.
3. Select the Cluster Termination runbook from the list of Runbooks and click **Start**.
   Azure executes the runbook script to terminate the cluster and remove cluster resources.

Next Steps

To see how to use Big Data Management, read Big Data Management documentation.

Each of these guides is available in the Big Data Management documentation set on the Informatica Network at http://network.informatica.com.

Informatica Application Service Guide

Describes the Model Repository Service, Data Integration Service, and other application services that Big Data Management uses.

Big Data Management User Guide

Describes how to use Informatica Developer and Informatica Administrator to manage connections between the Informatica domain and the cluster, and how to create mappings in the Developer tool.

Informatica Developer User Guide

Contains full details about how to use the Developer tool to create and run mappings and workflows.

Informatica Developer Mapping Guide

Contains full details about how to develop mappings in the Developer tool.

Author

Mark Pritchard
Principal Technical Writer