# 使用 DocumentDB 和 HDInsight 运行 Hadoop 作业

展示如何在带有 DocumentDB 的 Hadoop 连接器的 Azure HDInsight 上运行 [Apache Hive](http://hive.apache.org/)、[Apache Pig](http://pig.apache.org/) 和 [Apache Hadoop](http://hadoop.apache.org/) MapReduce 作业的教程。DocumentDB 的 Hadoop 连接器使 DocumentDB 可以充当 Hive、Pig 以及 MapReduce 作业的源和接收器。本教程将 DocumentDB 用作 Hadoop 作业的数据源和目的地。

完成本教程后，你将能够回答以下问题：

* 如何使用 Hive、Pig 或 MapReduce 作业从 DocumentDB 加载数据？
* 如何使用 Hive、Pig 或 MapReduce 作业从 DocumentDB 存储数据？

我们建议通过观看以下视频来入门，在视频中，我们通过使用 DocumentDB 和 HDInsight 来运行 Hive 作业。

[AZURE.VIDEO use-azure-documentdb-hadoop-connector-with-azure-hdinsight]

然后，返回到本文，在这里你将获得有关如何对 DocumentDB 数据运行分析作业的完整详细信息。

[AZURE.TIP] 本教程假定你之前有使用 Apache Hadoop、Hive 和/或 Pig的经验。如果你不熟悉 Apache Hadoop、Hive 和 Pig，建议访问 [Apache Hadoop documentation（Apache Hadoop 文档）](http://hadoop.apache.org/docs/current/)。本教程还假定你具有使用 DocumentDB 的经验，并且拥有一个 DocumentDB 帐户。如果你不熟悉 DocumentDB 或你没有 DocumentDB 帐户，请查阅[Getting Started（入门）](/documentation/articles/documentdb-get-started)页。

没有时间完成教程，只想获得有关 Hive、Pig 和 MapReduce 的完整示例 PowerShell 脚本？ 没问题，可在[此处](http://portalcontent.blob.core.windows.net/samples/documentdb-hdinsight-samples.zip)获得。此下载还包含对于这些示例的 hql、pig 及 java 文件。

## 最新版本

<tr><th>Hadoop 连接器版本</th>  
 <td>1.2.0</td></tr>  
<tr><th>脚本 URI</th>  
 <td>https://portalcontent.blob.core.windows.net/scriptaction/documentdb-hadoop-installer-v04.ps1</td></tr>  
<tr><th>修改日期</th>  
 <td>2016/04/26</td></tr>  
<tr><th>支持的 HDInsight 版本</th>  
 <td>3.1, 3.2</td></tr>  
<tr><th>更改日志</th>  
 <td>DocumentDB Java SDK 已更新到 1.6.0</br>  
 针对同时作为来源和接收器的已分区集合添加了支持</br>  
 </td></tr>

## 先决条件

在按照本教程中的说明操作之前，请确保已有下列各项：

* DocumentDB 帐户、数据库，以及内部已有文档的集合。有关详细信息，请参阅 [Getting Started with DocumentDB（DocumentDB 入门）](/documentation/articles/documentdb-get-started)。使用 [DocumentDB 导入工具](/documentation/articles/documentdb-import-data)将示例数据导入到你的 DocumentDB 帐户。
* 吞吐量。从 HDInsight 进行的读取和写入操作将计入你为集合分配的请求单位。有关详细信息，请参阅 [Provisioned throughput, request units, and database operations（设置的吞吐量、请求单位和数据库操作）](/documentation/articles/documentdb-manage#ProvThroughput)。
* 在每个输出集合中用于其他存储的步骤的容量。存储过程用于传输生成的文档。有关详细信息，请参阅 [Collections and provisioned throughput（集合和设置的吞吐量）](/documentation/articles/documentdb-manage#IndexOverhead)。
* 从 Hive、Pig 或 MapReduce 作业生成的文档的容量。有关详细信息，请参阅 [Manage DocumentDB capacity and performance（管理 DocumentDB 容量和性能）](/documentation/articles/documentdb-manage#Collections)。
* [可选]其他集合的容量。有关详细信息，请参阅 [Provisioned document storage and index overhead（设置的文档存储和索引开销）](/documentation/articles/documentdb-manage#IndexOverhead)。

[AZURE.WARNING] 为了避免在任何作业期间创建一个新集合，你可以将结果打印到 stdout，将输出保存到你的 WASB 容器，或指定一个现有集合。指定现有集合时，将在集合内创建新文档，如果 *ID* 中有冲突，只会影响现有文档。**连接器将自动覆盖出现 ID 冲突的现有文档**。通过将 upsert 选项设置为 false 可以关闭此功能。如果 upsert 为 false，并且发生冲突，则 Hadoop 作业将失败；并报告 ID 冲突错误。

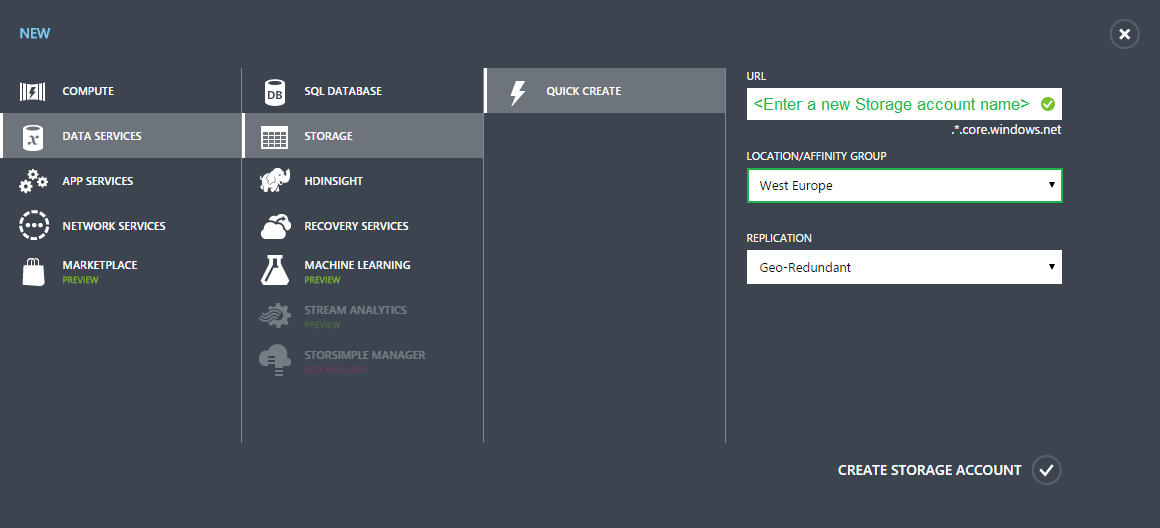
## 步骤 1：创建 Azure 存储帐户

[AZURE.IMPORTANT] 如果你**已**拥有 Azure 存储帐户并且你想要在此帐户内创建的新 blob 容器，则可以跳到[步骤 2：创建自定义的 HDInsight 群集](#ProvisionHDInsight)。

Azure HDInsight 使用 Azure Blob 存储来存储数据。我们称之为 WASB 或 Azure 存储空间 - Blob。WASB 是 Microsoft 在 Azure Blob 存储上的 HDFS 实现。有关详细信息，请参阅[将 Azure Blob 存储与 HDInsight 配合使用](/documentation/articles/hdinsight-hadoop-use-blob-storage)。

你设置 HDInsight 群集时，指定 Azure 存储帐户。将该帐户的一个特定 Blob 存储容器指定为默认文件系统，如同在 HDFS 中一样。默认情况下，HDInsight 群集与你指定的存储帐户设置在同一数据中心内。

**创建 Azure 存储帐户**

1. 登录[Azure 经典门户](https://manage.windowsazure.cn/)。
2. 单击左下角的“+ 新建”，指向“数据服务”，指向“存储”，然后单击“快速创建”。
3. 输入 **URL**、选择“位置”和“复制”的值，然后单击“创建存储帐户”。不支持地缘组。

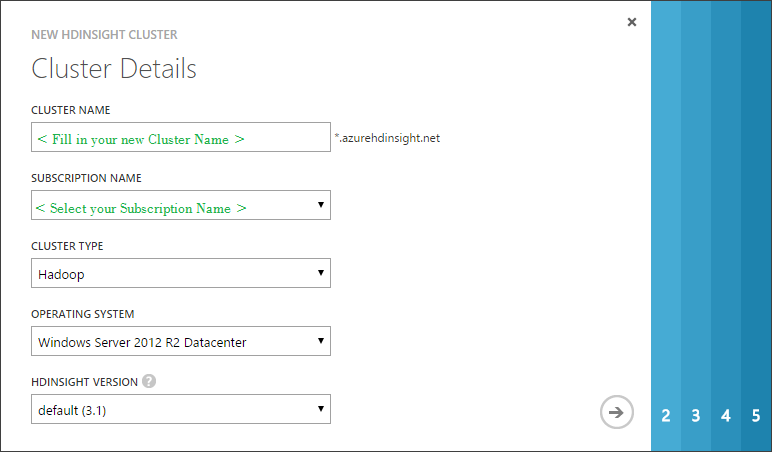
* 你将在存储列表中看到新的存储帐户。
* [AZURE.IMPORTANT] 为获得最佳性能，请确保你的存储帐户、HDInsight 群集和 DocumentDB 帐户位于同一 Azure 区域。

1. 等待直到新存储帐户的**状态**更改为**联机**。

## 步骤 2：创建自定义的 HDInsight 群集

本教程使用 Azure 经典门户中的脚本操作自定义 HDInsight 群集。在本教程中，我们将使用 Azure 经典门户来创建自定义群集。有关如何使用 PowerShell cmdlet 或 HDInsight .NET SDK 的说明，请参阅 [Customize HDInsight clusters using Script Action（使用脚本操作自定义 HDInsight 群集）](/documentation/articles/hdinsight-provision-clusters#powershell)文章。

1. 登录到 [Azure 经典门户](https://manage.windowsazure.cn/)。你可能已在前一步骤中登录。
2. 单击页面底部的“+ 新建’，然后依次单击“数据服务”、“HDINSIGHT”和“自定义创建”。
3. 在“群集详细信息”页上，键入或选择以下值：

* 
* 提供 Hadoop HDInsight 初始群集详细信息
* <tr><th>属性</th><th>值</th></tr>  
  <tr><td>群集名称</td><td>命名群集。<br/>  
   DNS 名称必须以字母数字字符开头和结尾，并且可以包含短划线。<br/>  
   字段必须是介于 3 到 63 个字符之间的字符串。</td></tr>  
  <tr><td>订阅名称</td>  
   <td>如果你有多个 Azure 订阅，选择对应于<strong>步骤 1</strong> 的存储帐户的订阅。</td></tr>  
  <tr><td>群集类型</td>  
   <td>对于群集类型，请选择“Hadoop”。<strong></strong></td></tr>  
  <tr><td>操作系统</td>  
   <td>对于操作系统，请选择“Windows Server 2012 R2 Datacenter”<strong></strong>。</td></tr>  
  <tr><td>HDInsight 版本</td>  
   <td>选择版本。</br>选择“HDInsight 版本 3.1”<Strong></Strong>。</td></tr>  
  </table>
* 输入或选择表中所示的值，然后单击右箭头。

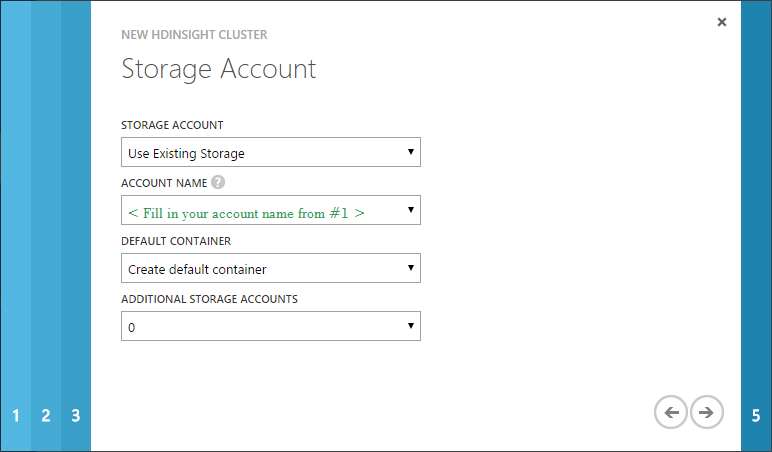
1. 在“配置群集”页上，输入或选择以下值：

* 名称
* 值
* 数据节点
* 要部署的数据节点的数目。记住 HDInsight 的数据节点与性能和定价相关联。
* 区域/虚拟网络
* 选择与你新创建的存储帐户和 DocumentDB 帐户相同的区域。 HDInsight 要求存储帐户位于同一区域中。稍后，在配置中，你只能选择你在此处指定的区域中的存储帐户。
* 单击右箭头。

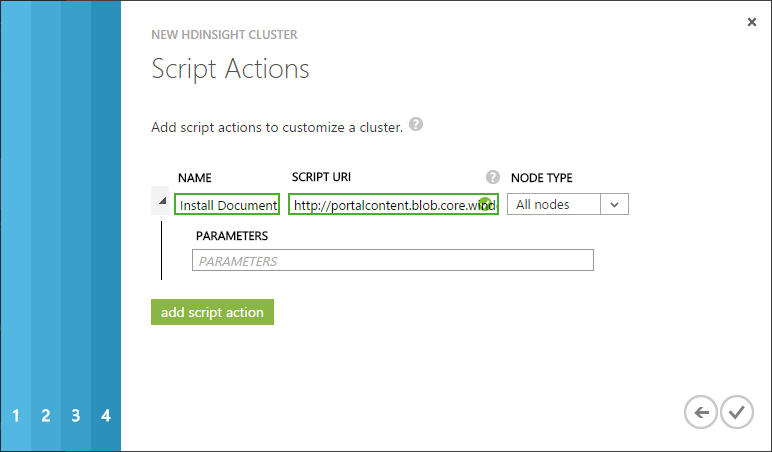
1. 在“配置群集用户”页上提供以下值：

* <tr><th>属性</th><th>值</th></tr>  
  <tr><td>用户名</td>  
   <td>指定 HDInsight 群集用户名。</td></tr>  
  <tr><td>密码/确认密码</td>  
   <td>指定 HDInsight 群集用户密码。</td></tr>
* 单击右箭头。

1. 在“存储帐户”页上提供以下值：

* 
* 提供 Hadoop HDInsight 群集的存储帐户
* <tr><th>属性</th><th>值</th></tr>  
  <tr><td>存储帐户</td>  
   <td>为 HDInsight 群集指定将用作默认文件系统的 Azure 存储帐户。你可以选择三个选项之一：使用现有存储、创建新存储或使用其他订阅中的存储</br></br>  
   选择“使用现有存储”<strong></strong>。  
   </td>  
   </td></tr>  
  <tr><td>帐户名</td>  
   <td>  
   对于“存储帐户名称”<strong></strong>，选择<strong>步骤 1</strong>中创建的帐户。下拉列表中只列出了同一 Azure 订阅下你选择在其中设置群集的同一数据中心内的存储帐户。  
   </td></tr>  
  <tr><td>默认容器</td>  
   <td>指定存储帐户上用作 HDInsight 群集默认文件系统的默认容器。如果你为“存储帐户”<strong></strong>字段选择了“使用现有存储”<strong></strong>，而此帐户中没有现有容器，将按默认值创建容器，其名称与群集名称相同。如果已存在与群集同名的容器，则将在容器名称后追加一个序列号。  
  </td></tr>  
  <tr><td>其他存储帐户</td>  
   <td>HDInsight 支持多个存储帐户。对于可由群集使用的其他存储帐户则没有限制。但是，如果你使用 Azure 经典门户创建群集，由于 UI 限制，上限则为 7。指定的每个其他存储帐户将在向导中添加一个额外的“存储帐户”页，以便你在此指定帐户信息。</td></tr>
* 单击右箭头。

1. 在“脚本操作”页上，单击“添加脚本操作”以提供有关你将要运行以在创建群集时自定义群集的 PowerShell 脚本。此 PowerShell 脚本将在创建群集时将 DocumentDB Hadoop 连接器安装到你的 HDInsight 群集。

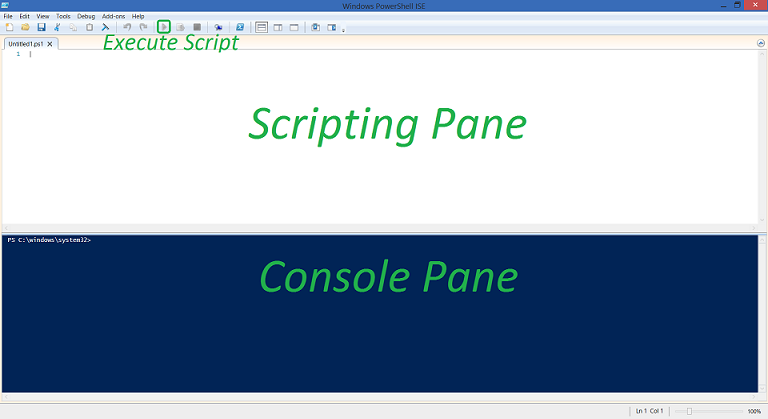
* 
* 配置脚本操作以自定义 HDInsight 群集
* <tr><th>属性</th><th>值</th></tr>  
  <tr><td>Name</td>  
   <td>指定脚本操作的名称。</td></tr>  
  <tr><td>脚本 URI</td>  
   <td>指定要调用来自定义群集的脚本的 URI。</br></br>  
   请输入： </br> <strong>https://portalcontent.blob.core.windows.net/scriptaction/documentdb-hadoop-installer-v03.ps1</strong>。</td></tr>  
  <tr><td>节点类型</td>  
   <td>指定在其上运行自定义脚本的节点。你可以选择“所有节点”、“仅限头节点”或“仅限从节点”<b></b><b></b><b></b>。</br></br>  
   请选择“所有节点”<strong></strong>。</td></tr>  
  <tr><td>Parameters</td>  
   <td>根据脚本的需要，指定参数。</br></br>  
   <strong>无需参数</strong>。</td></tr>
* 单击复选标记以完成群集创建。

## 步骤 3：安装和配置 Azure PowerShell。

1. 安装 Azure PowerShell 中的说明进行操作。可在[此处](/documentation/articles/powershell-install-configure)找到说明。

* [AZURE.NOTE] 或者，只需了解 Hive 查询，可以使用 HDInsight 的联机 Hive 编辑器。若要这样做，请登录到 [Azure 经典门户](https://manage.windowsazure.cn/)，单击左侧窗格中的“HDInsight”以查看 HDInsight 群集的列表。单击要对其运行 Hive 查询的群集，然后单击“查询控制台”。

1. 打开 Azure PowerShell 集成脚本环境：
   * 在运行 Windows 8 或 Windows Server 2012 或更高版本的计算机上，可以使用内置搜索。从“开始”屏幕上，键入 **powershell ise** 并单击“Enter”。
   * 在运行早于 Windows 8 或 Windows Server 2012 的版本的计算机上，请使用“开始”菜单。从“开始”菜单上，在搜索框中键入**命令提示符**，然后在结果列表中，单击“命令提示符”。在命令提示符中，键入 **powershell\_ise** ，然后单击“Enter”。
2. 添加你的 Azure 帐户。
   1. 在控制台窗格中，键入 **Add-AzureAccount** 并单击“Enter”。
   2. 键入与你的 Azure 订阅相关联的电子邮件地址并单击“继续”。
   3. 键入你的 Azure 订阅的密码。
   4. 单击“登录”。
3. 以下关系图标识你的 Azure PowerShell 脚本环境的重要部分。

* 
* Azure PowerShell 的关系图

## 步骤 4：使用 DocumentDB 和 HDInsight 运行 Hive 作业

[AZURE.IMPORTANT] 必须使用你的配置设置填写所有由 < > 表示的变量。

1. 在你的 PowerShell 脚本窗格中设置以下变量。

* # Provide Azure subscription name, the Azure Storage account and container that is used for the default HDInsight file system.  
  $subscriptionName = "<SubscriptionName>"  
  $storageAccountName = "<AzureStorageAccountName>"  
  $containerName = "<AzureStorageContainerName>"  
    
  # Provide the HDInsight cluster name where you want to run the Hive job.  
  $clusterName = "<HDInsightClusterName>"
* 让我们开始构造查询字符串。我们将编写 Hive 查询，该查询采用来自 DocumentDB 集合的所有文档的系统生成的时间戳 (\_ts) 和唯一 ID (\_rid)，按分钟计算所有文档，然后将结果存储回新 DocumentDB 集合。
* 首先，让我们从 DocumentDB 集合创建 Hive 表。将以下代码段添加到 PowerShell 脚本窗格中从 #1 开始的代码段之后。请确保包括可选的 DocumentDB.query 参数，以便将我们的文档调整为 just\_ts 和 \_rid。
* [AZURE.NOTE] **命名 DocumentDB.inputCollections 不是一个错误。** 是，我们允许添加多个集合作为输入： “DocumentDB.inputCollections”=“<DocumentDB Input Collection Name 1>,<DocumentDB Input Collection Name 2>”不使用空格分隔集合名称，仅使用单个逗号。
* # Create a Hive table using data from DocumentDB. Pass DocumentDB the query to filter transferred data to \_rid and \_ts.  
  $queryStringPart1 = "drop table DocumentDB\_timestamps; " +   
   "create external table DocumentDB\_timestamps(id string, ts BIGINT) " +  
   "stored by 'com.microsoft.azure.documentdb.hive.DocumentDBStorageHandler' " +  
   "tblproperties ( " +   
   "'DocumentDB.endpoint' = '<DocumentDB Endpoint>', " +  
   "'DocumentDB.key' = '<DocumentDB Primary Key>', " +  
   "'DocumentDB.db' = '<DocumentDB Database Name>', " +  
   "'DocumentDB.inputCollections' = '<DocumentDB Input Collection Name>', " +  
   "'DocumentDB.query' = 'SELECT r.\_rid AS id, r.\_ts AS ts FROM root r' ); "

1. 接下来，让我们为输出集合创建 Hive 表。输出文档属性将为月、日、小时、分钟和发生次数总数。

* [AZURE.NOTE] **再次，命名 DocumentDB.outputCollections 不是一个错误。** 是，我们允许添加多个集合作为输出： “DocumentDB.outputCollections”=“<DocumentDB Output Collection Name 1>,<DocumentDB Output Collection Name 2>”不使用空格分隔集合名称，仅使用单个逗号。 文档将为跨多个集合的分布式轮循机制。一批文档将存储在一个集合中，第二批文档则存储在下一个集合中，如此类推。
* # Create a Hive table for the output data to DocumentDB.  
  $queryStringPart2 = "drop table DocumentDB\_analytics; " +  
   "create external table DocumentDB\_analytics(Month INT, Day INT, Hour INT, Minute INT, Total INT) " +  
   "stored by 'com.microsoft.azure.documentdb.hive.DocumentDBStorageHandler' " +   
   "tblproperties ( " +   
   "'DocumentDB.endpoint' = '<DocumentDB Endpoint>', " +  
   "'DocumentDB.key' = '<DocumentDB Primary Key>', " +   
   "'DocumentDB.db' = '<DocumentDB Database Name>', " +  
   "'DocumentDB.outputCollections' = '<DocumentDB Output Collection Name>' ); "

1. 最后，让我们按月、日、小时和分钟计算文档，并将结果插入回输出 Hive 表。

* # GROUP BY minute, COUNT entries for each, INSERT INTO output Hive table.  
  $queryStringPart3 = "INSERT INTO table DocumentDB\_analytics " +  
   "SELECT month(from\_unixtime(ts)) as Month, day(from\_unixtime(ts)) as Day, " +  
   "hour(from\_unixtime(ts)) as Hour, minute(from\_unixtime(ts)) as Minute, " +  
   "COUNT(\*) AS Total " +  
   "FROM DocumentDB\_timestamps " +  
   "GROUP BY month(from\_unixtime(ts)), day(from\_unixtime(ts)), " +  
   "hour(from\_unixtime(ts)) , minute(from\_unixtime(ts)); "

1. 添加以下脚本代码段以从之前的查询创建 Hive 作业定义。

* # Create a Hive job definition.  
  $queryString = $queryStringPart1 + $queryStringPart2 + $queryStringPart3  
  $hiveJobDefinition = New-AzureHDInsightHiveJobDefinition -Query $queryString
* 还可以使用 -File 开关来指定 HDFS 上的 HiveQL 脚本文件。

1. 添加以下代码段以保存开始时间并提交 Hive 作业。

* # Save the start time and submit the job to the cluster.  
  $startTime = Get-Date  
  Select-AzureSubscription $subscriptionName  
  $hiveJob = Start-AzureHDInsightJob -Cluster $clusterName -JobDefinition $hiveJobDefinition

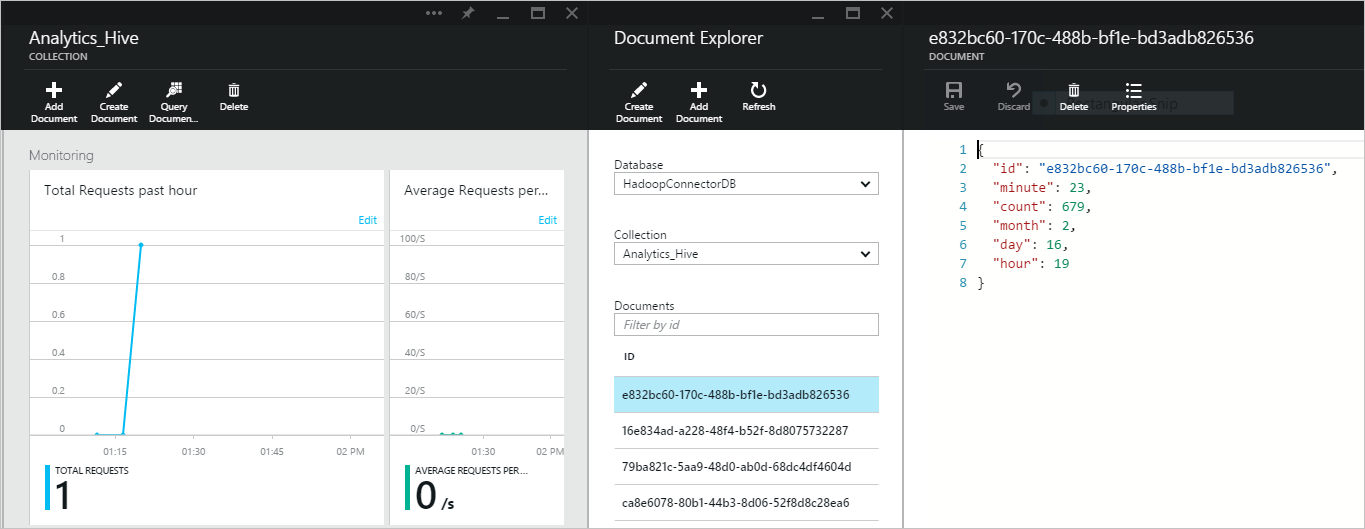
1. 添加以下内容来等待 Hive 作业完成。

* # Wait for the Hive job to complete.  
  Wait-AzureHDInsightJob -Job $hiveJob -WaitTimeoutInSeconds 3600

1. 添加以下内容以打印标准输出以及开始和结束时间。

* # Print the standard error, the standard output of the Hive job, and the start and end time.  
  $endTime = Get-Date  
  Get-AzureHDInsightJobOutput -Cluster $clusterName -JobId $hiveJob.JobId -StandardOutput  
  Write-Host "Start: " $startTime ", End: " $endTime -ForegroundColor Green

1. **运行**新的脚本！ **单击**绿色执行按钮。
2. 检查结果。登录到 [Azure 门户](https://portal.azure.cn/)。
   1. 单击左侧面板上的“浏览”。
   2. 单击浏览面板右上角的“全部”。
   3. 找到并单击“DocumentDB 帐户”。
   4. 接下来，找到你的 DocumentDB 帐户、DocumentDB 数据库和与 Hive 查询中指定的输出集合相关联的 DocumentDB 集合。
   5. 最后，单击“开发人员工具”下方的“文档资源管理器”。

* 你将看到 Hive 查询的结果。
* 
* Hive 查询结果

## 步骤 5：使用 DocumentDB 和 HDInsight 运行 Pig 作业

[AZURE.IMPORTANT] 必须使用你的配置设置填写所有由 < > 表示的变量。

1. 在你的 PowerShell 脚本窗格中设置以下变量。

* # Provide Azure subscription name.  
  $subscriptionName = "Azure Subscription Name"  
    
  # Provide HDInsight cluster name where you want to run the Pig job.  
  $clusterName = "Azure HDInsight Cluster Name"
* 让我们开始构造查询字符串。我们将编写 Pig 查询，该查询采用来自 DocumentDB 集合的所有文档的系统生成的时间戳 (\_ts) 和唯一 ids (\_rid)，按分钟计算所有文档，然后将结果存储回新 DocumentDB 集合。
* 首先，从 DocumentDB 将文档加载到 HDInsight 中。将以下代码段添加到 PowerShell 脚本窗格中从 #1 开始的代码段之后。请确保添加了 DocumentDB.query 到可选的 DocumentDB 查询参数，以便将我们的文档调整到 just\_ts 和 \_rid。
* [AZURE.NOTE] 是，我们允许添加多个集合作为输入： “<DocumentDB Input Collection Name 1>,<DocumentDB Input Collection Name 2>”不使用空格分隔集合名称，仅使用单个逗号。
* 文档将为跨多个集合的分布式轮循机制。一批文档将存储在一个集合中，第二批文档则存储在下一个集合中，如此类推。
* # Load data from DocumentDB. Pass DocumentDB query to filter transferred data to \_rid and \_ts.  
  $queryStringPart1 = "DocumentDB\_timestamps = LOAD '<DocumentDB Endpoint>' USING com.microsoft.azure.documentdb.pig.DocumentDBLoader( " +  
   "'<DocumentDB Primary Key>', " +  
   "'<DocumentDB Database Name>', " +  
   "'<DocumentDB Input Collection Name>', " +  
   "'SELECT r.\_rid AS id, r.\_ts AS ts FROM root r' ); "

1. 接下来，让我们按月、日、小时、分钟和发生次数总数计算文档。

* # GROUP BY minute and COUNT entries for each.  
  $queryStringPart2 = "timestamp\_record = FOREACH DocumentDB\_timestamps GENERATE `$0#'id' as id:int, ToDate((long)(`$0#'ts') \* 1000) as timestamp:datetime; " +  
   "by\_minute = GROUP timestamp\_record BY (GetYear(timestamp), GetMonth(timestamp), GetDay(timestamp), GetHour(timestamp), GetMinute(timestamp)); " +  
   "by\_minute\_count = FOREACH by\_minute GENERATE FLATTEN(group) as (Year:int, Month:int, Day:int, Hour:int, Minute:int), COUNT(timestamp\_record) as Total:int; "

1. 最后，让我们将结果存储到我们新的输出集合。

* [AZURE.NOTE] 是，我们允许添加多个集合作为输出： “<DocumentDB Output Collection Name 1>,<DocumentDB Output Collection Name 2>”不使用空格分隔集合名称，仅使用单个逗号。 文档将是跨多个集合的分布式轮循机制。一批文档将存储在一个集合中，第二批文档则存储在下一个集合中，如此类推。
* # Store output data to DocumentDB.  
  $queryStringPart3 = "STORE by\_minute\_count INTO '<DocumentDB Endpoint>' " +  
   "USING com.microsoft.azure.documentdb.pig.DocumentDBStorage( " +  
   "'<DocumentDB Primary Key>', " +  
   "'<DocumentDB Database Name>', " +  
   "'<DocumentDB Output Collection Name>'); "

1. 添加以下脚本代码段以从之前的查询创建 Pig 作业定义。

* # Create a Pig job definition.  
  $queryString = $queryStringPart1 + $queryStringPart2 + $queryStringPart3  
  $pigJobDefinition = New-AzureHDInsightPigJobDefinition -Query $queryString -StatusFolder $statusFolder
* 你也可以使用 -File 开关来指定 HDFS 上的 Pig 脚本文件。

1. 添加以下代码段以保存开始时间并提交 Pig 作业。

* # Save the start time and submit the job to the cluster.  
  $startTime = Get-Date  
  Select-AzureSubscription $subscriptionName  
  $pigJob = Start-AzureHDInsightJob -Cluster $clusterName -JobDefinition $pigJobDefinition

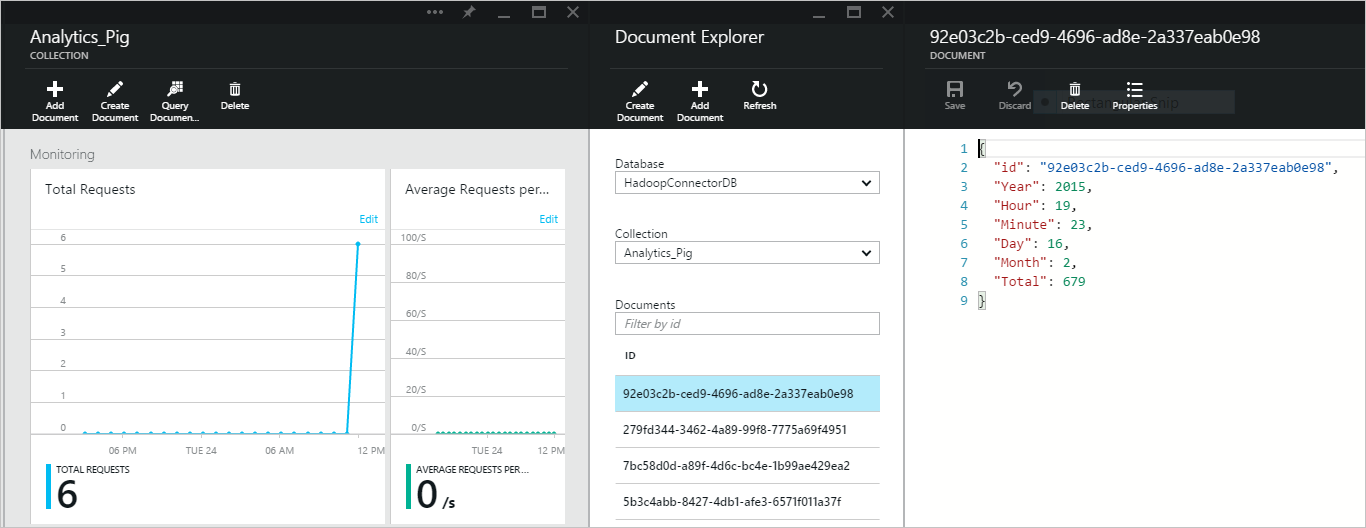
1. 添加以下内容来等待 Pig 作业完成。

* # Wait for the Pig job to complete.  
  Wait-AzureHDInsightJob -Job $pigJob -WaitTimeoutInSeconds 3600

1. 添加以下内容以打印标准输出以及开始和结束时间。

* # Print the standard error, the standard output of the Hive job, and the start and end time.  
  $endTime = Get-Date  
  Get-AzureHDInsightJobOutput -Cluster $clusterName -JobId $pigJob.JobId -StandardOutput  
  Write-Host "Start: " $startTime ", End: " $endTime -ForegroundColor Green

1. **运行**新的脚本！ **单击**绿色执行按钮。
2. 检查结果。登录到 [Azure 门户](https://portal.azure.cn/)。
   1. 单击左侧面板上的“浏览”。
   2. 单击浏览面板右上角的“全部”。
   3. 找到并单击“DocumentDB 帐户”。
   4. 接下来，找到你的 DocumentDB 帐户、DocumentDB 数据库和与 Pig 查询中指定的输出集合相关联的 DocumentDB 集合。
   5. 最后，单击“开发人员工具”下方的“文档资源管理器”。

* 你将看到 Pig 查询的结果。
* 
* Pig 查询结果

## 步骤 6：使用 DocumentDB 和 HDInsight 运行 MapReduce 作业

1. 在你的 PowerShell 脚本窗格中设置以下变量。

* $subscriptionName = "<SubscriptionName>" # Azure subscription name  
  $clusterName = "<ClusterName>" # HDInsight cluster name

1. 我们将执行 MapReduce 作业，该作业从你的 DocumentDB 集合计算每个文档属性的发生次数。在以上代码段**之后**添加此代码段。

* # Define the MapReduce job.  
  $TallyPropertiesJobDefinition = New-AzureHDInsightMapReduceJobDefinition -JarFile "wasb:///example/jars/TallyProperties-v01.jar" -ClassName "TallyProperties" -Arguments "<DocumentDB Endpoint>","<DocumentDB Primary Key>", "<DocumentDB Database Name>","<DocumentDB Input Collection Name>","<DocumentDB Output Collection Name>","<[Optional] DocumentDB Query>"
* [AZURE.NOTE] DocumentDB Hadoop 连接器自定义安装中附带了 TallyProperties-v01.jar。

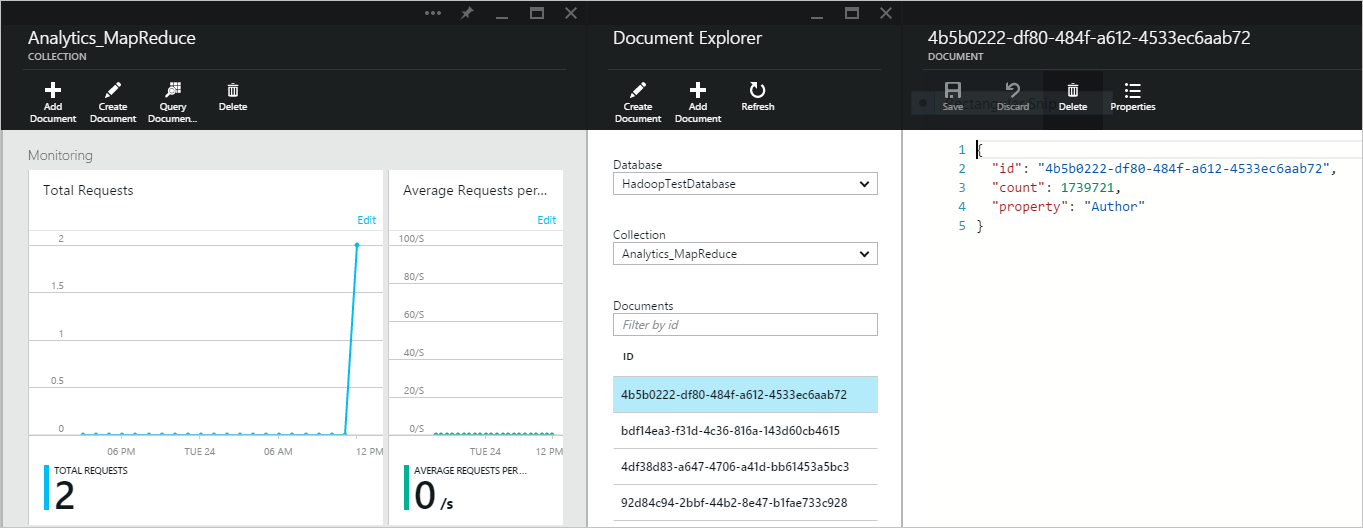
1. 添加以下命令来提交 MapReduce 作业。

* # Save the start time and submit the job.  
  $startTime = Get-Date  
  Select-AzureSubscription $subscriptionName  
  $TallyPropertiesJob = Start-AzureHDInsightJob -Cluster $clusterName -JobDefinition $TallyPropertiesJobDefinition | Wait-AzureHDInsightJob -WaitTimeoutInSeconds 3600
* 除了 MapReduce 作业定义外，你还要提供需运行 MapReduce 作业的 HDInsight 群集名称，以及凭据。Start-AzureHDInsightJob 是异步调用。要检查作业是否完成，请使用 Wait-AzureHDInsightJob cmdlet。

1. 添加以下命令来检查运行 MapReduce 作业时的错误。

* # Get the job output and print the start and end time.  
  $endTime = Get-Date  
  Get-AzureHDInsightJobOutput -Cluster $clusterName -JobId $TallyPropertiesJob.JobId -StandardError  
  Write-Host "Start: " $startTime ", End: " $endTime -ForegroundColor Green

1. **运行**新的脚本！ **单击**绿色执行按钮。
2. 检查结果。登录到 [Azure 门户](https://portal.azure.cn/)。
   1. 单击左侧面板上的“浏览”。
   2. 单击浏览面板右上角的“全部”。
   3. 找到并单击“DocumentDB 帐户”。
   4. 接下来，找到你的 DocumentDB 帐户、DocumentDB 数据库和与 MapReduce 作业中指定的输出集合相关联的 DocumentDB 集合。
   5. 最后，单击“开发人员工具”下方的“文档资源管理器”。

* 你将看到 MapReduce 作业的结果。
* 
* MapReduce 查询结果

## 后续步骤

祝贺你！ 你只需使用 Azure DocumentDB 和 HDInsight 运行你的第一个 Hive、Pig 和 MapReduce 作业。

我们的 Hadoop Connector 是开源的。如果你感兴趣，可以在 [GitHub](https://github.com/Azure/azure-documentdb-hadoop) 上参与。

若要了解更多信息，请参阅下列文章：

* [使用 Documentdb 开发 Java 应用程序](/documentation/articles/documentdb-java-application)
* [为 HDInsight 中的 Hadoop 开发 Java MapReduce 程序](/documentation/articles/hdinsight-develop-deploy-java-mapreduce)
* [将 Hadoop 与 HDInsight 中的 Hive 配合使用以分析手机使用情况](/documentation/articles/hdinsight-hadoop-tutorial-get-started-windows)
* [将 MapReduce 与 HDInsight 配合使用](/documentation/articles/hdinsight-use-mapreduce)
* [将 Hive 与 HDInsight 配合使用](/documentation/articles/hdinsight-use-hive)
* [将 Pig 与 HDInsight 配合使用](/documentation/articles/hdinsight-use-pig)
* [使用脚本操作自定义 HDInsight 群集](/documentation/articles/hdinsight-hadoop-customize-cluster)