Azure Batch Service

Azure Batch Service Setup

March 2017

Contents

[About this solution 3](#_Toc476487727)

[Goals 3](#_Toc476487728)

[Solution 3](#_Toc476487729)

[Solution details 3](#_Toc476487730)

[Batch Service Solution 4](#_Toc476487731)

# About this solution

## Goals

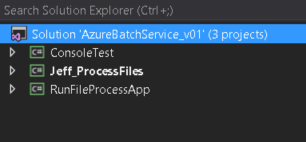
The goal of this document is to building azure batch service to run custom application to process files in parallel to generate output file in very short time.

# Solution

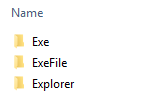
## Solution details

Azure batch service solution has three projects.

1. FileprocessApp project is a utility to run the custom application on the batch service node and copy the output files to azure output directory upon completion.
2. ConsoleTest project is used to generate sample copy of files in azure storage for testing.
3. Jefff\_ProcessFiles project is the main project for batch service configuration and run the service.



Pre-build solution available in package. Which has compiled executables and explorer application to monitor & manage azure batch services.



1. Exe Folder contains precompiled executables which can run directly.
2. It has two executable files and two config files.
   1. Jeff\_ProcessFiles.exe.config file used to configure the batch service & storage service details in xml format.
   2. RunFileProcessApp.exe.config files to configure the storage account to dump the final result files.
3. ExeFile is used to run the test exe file which will run across all batch node to process files in parallel
4. Explorer folder contains the application to monitor & manage the batch service.

## Batch Service Solution

| Steps | Screenshot |
| --- | --- |
| 1. Login to azure portal and go to service select compute and batch service as mentioned in the snapshot. |  |
| 1. You will need an Azure Storage Account to keep files for process and store the result files. 2. Login to your Azure subscription, then navigate to the Storage section. 3. Select Quick Create. |  |
| 1. Go to the newly created storage account and click on Access Keys |  |
| Configuration : Jeff\_ProcessFiles.exe.config  Batch Service configuration section (line no 8-16)  Compute node is denote number of VM  vmSize denote each VM core and RAM configuration  (ref : <https://msdn.microsoft.com/en-us/library/dn168976(v=nav.70).aspx)>  TaskPerNode denote number instances created in each VM.  WaitTime for batch service timeout period to kill the process if any exceptional case.  BatchServiceURl is taken from batch service overview blade in azure portal  Similarly we can fetch batch account name and accountkey can be taken from azure portal.  Storage Details section (line no 18-20)  StorageAccountName and StorageAcountKey for the above configured storage account.  *Note: Preferably batch service and storage account need to be created in same region for better performance.* |  |
|  |
| Configuration : RunFileProcessApp.exe.config  This is utility application which execute the require application and copy the generated output file to azure storage.  StorageAccountName and StorageAcountKey for the above configured storage account.  Containername is the location where the output files are copied by this utility. |  |
| **Source Code**: RunFileProcessApp project program.cs file  Main function is the entry point for the utility application.   1. Loadconfig will read the config value from xml file and update to variables. 2. Followed by setting up the storage account client 3. Line 51, will frame the commandline text which will invoke through command line. 4. It will wait till this process to complete. 5. Finally, it will upload the output files to storage. |  |
| **Source Code**: Jeff\_ProcessFiles project RunJob.cs file  MainAsync is the entry point for this program.   1. Loadconfig will read the config value from xml file and update to variables. 2. Followed by setting up the batch client and storage account client based on the config values 3. Line 121, define the dependent files for copied to all the VM (node) for execution.    1. Create\_test.exe    2. RunFileProcessApp.exe (utility app)    3. RunFileProcessApp.exe.config (xml file) 4. Followed by uploading all these files to azure storage. 5. Line 141- fetch all the input files from input container and load into variable. 6. From line 149 onwards, this will create the pool, job and tasks for execution. 7. Line 164, will wait for the all tasks to complete |  |
| Line 167, Copy all the available files in the azure storage output directory to local folder.  Currently it used temp folder as destination.  Followed by deleting all the files in azure storage.  Then it will ask user to delete the configured batch service pool, job & tasks and do the action based on users input.  *Note: Detailed code available on each function by right click and select GoToDefinition.* |  |