ユートリアル:Azure Functions を使用して Batch ジョブをトリガーする

2019/05/30

この記事の内容

このチュートリアルでは、Azure Functions を使用して Batch ジョブをトリガーする方法について説明します。 Azure Storage Blob コンテナーに追加されたドキュメントに Azure Batch を介して光学式文字認識 (OCR) を適用する例を見ていきます。 OCR 処理を効率化するために、BLOB コンテナーにファイルが追加されるたびに Batch OCR ジョブを実行する Azure 関数を構成します。 学習内容は次のとおりです。

Batch Explorer を使用して、プールおよびジョブを作成する
Storage Explorer を使用して、BLOB コンテナーと Shared Access Signature (SAS) を作成する
BLOB によってトリガーされる Azure 関数を作成する
Storage に入力ファイルをアップロードする
タスクの実行を監視する
出力ファイルを取得する

前提条件

- Azure サブスクリプション。 お持ちでない場合は、開始する前に無料アカウントを作成してください。
- Azure Batch アカウントおよびリンクされた Azure ストレージ アカウント。 アカウントを作成およびリンクする方法の詳細については、「Batch アカウントを作成する」を参照してください。
- Batch Explorer
- Azure 記憶域エクスプローラー

Azure へのサインイン

Azure portal にサインインします。

Batch Explorer を使用して Batch プールと Batch ジョブを作成する

このセクションでは、Batch Explorer を使用して、OCR タスクを実行する Batch プールと Batch ジョブを作成します。

プールを作成する

1. ご自分の Azure 資格情報を使用して、Batch Explorer にサインインします。
2. 左側のバーにある **[Pools](プール)** 、検索フォームの上にある **[Add](追加)** ボタンの順に選択して、プールを作成します。
   a. ID と表示名を選択します。 この例では、 `ocr-pool` を使用します。
   b. スケールの種類を **[Fixed size](固定サイズ)** に設定し、専用ノードの数を 3 に設定します。
   c. オペレーティング システムとして **[Ubuntu 18.04-LTS]** を選択します。
   d. 仮想マシンのサイズとして `[Standard_f2s_v2]` を選択します。
   e. 開始タスクを有効にし、コマンド `/bin/bash -c "sudo update-locale LC_ALL=C.UTF-8 LANG=C.UTF-8; sudo apt-get update; sudo apt-get -y install ocrmypdf"` を追加します。 必ずユーザー ID を **[Task default user (Admin)](タスクの既定のユーザー (管理者))** として設定します。これにより、開始タスクに `sudo` を使用したコマンドを含めることができます。
   f. **[OK]** を選択します。

ジョブの作成

1. 左側のバーにある **[Jobs](ジョブ)**、検索フォームの上にある **[Add](追加)** ボタンの順に選択して、プール上にジョブを作成します。
   a. ID と表示名を選択します。 この例では、 `ocr-job` を使用します。
   b. プールを `ocr-pool` またはご自分のプールに対して選択した名前に設定します。
   c. **[OK]** を選択します。

BLOB コンテナーを作成する

ここで、OCR Batch ジョブの実際の入力および出力ファイルを格納する BLOB コンテナーを作成します。

1. ご自分の Azure 資格情報を使用して、Storage Explorer にサインインします。
2. ご自分の Batch アカウントにリンクされているストレージ アカウントを使用し、「BLOB コンテナーを作成する」の手順に従って、2 つの BLOB コンテナー (1 つは入力ファイル用、1 つは出力ファイル用) を作成します。

この例では、入力コンテナーは `input` という名前で、そこには OCR が適用されていないドキュメントがすべて処理のために最初にアップロードされます。 出力コンテナーは `output` という名前で、そこには OCR が適用された処理済みのドキュメントが Batch ジョブによって書き込まれます。
* この例では、入力コンテナーを `input`、出力コンテナーを `output` と呼ぶことにします。
* 入力コンテナーには、OCR が適用されていないドキュメントがすべて最初にアップロードされます。
* 出力コンテナーには、OCR が適用されたドキュメントが Batch ジョブによって書き込まれます。

Storage Explorer 内で、ご自分の出力コンテナーの Shared Access Signature を作成します。 これを行うには、出力コンテナーを右クリックし、 **[Get Shared Access Signature](Shared Access Signature の取得)** を選択します。 **[Permissions](アクセス許可)** の下にある **[Write](書き込み)** チェック ボックスをオンにします。 それ以外のアクセス許可は必要ありません。

Azure Function の作成

このセクションでは、ファイルがご自分の入力コンテナーにアップロードされるたびに OCR Batch ジョブをトリガーする Azure 関数を作成します。

1. 「Azure Blob Storage によってトリガーされる関数の作成」の手順に従って、関数を作成します。
   a. ストレージ アカウントを求められたら、ご自分の Batch アカウントにリンクしたのと同じストレージ アカウントを使用します。
   b. **[ランタイム スタック]** で [.NET] を選択します。 Batch .NET SDK を活用するために、C# で関数を作成します。
2. BLOB によってトリガーされる関数を作成したら、その関数内で GitHub の run.csx および function.proj を使用します。

   - `run.csx` は、ご自分の入力 BLOB コンテナーに新しい BLOB が追加されると実行されます。
   - `function.proj` は、Batch .NET SDK など、ご自分の関数コード内で外部ライブラリを列挙します。

3. `run.csx` ファイルの `Run()` 関数内にある変数のプレースホルダーの値を変更して、ご自分の Batch およびストレージの資格情報を反映させます。 ご自分の Batch およびストレージ アカウントの資格情報は、Azure portal 内にあるご自分の Batch アカウントの **[キー]** セクションで確認できます。

   - Azure portal 内にあるご自分の Batch アカウントの **[キー]** セクションでご自分の Batch およびストレージ アカウントの資格情報を取得します。

関数をトリガーし、結果を取得する

スキャン済みファイルの一部またはすべてを GitHub 上の input_files ディレクトリからご自分の入力コンテナーにアップロードします。 Batch Explorer を監視して、各ファイルのタスクが `ocr-pool` に追加されることを確認します。 数秒後に、OCR が適用されたファイルが出力コンテナーに追加されます。 その後、ファイルは Storage Explorer 上で表示および取得できるようになります。

さらに、Azure Functions Web エディター ウィンドウの下部にあるログ ファイルを確認できます。そこには、ご自分の入力コンテナーにアップロードしたすべてのファイルに関する、このようなメッセージが表示されます。

コピー

```
2019-05-29T19:45:25.846 [Information] Creating job...
2019-05-29T19:45:25.847 [Information] Accessing input container <inputContainer>...
2019-05-29T19:45:25.847 [Information] Adding <fileName> as a resource file...
2019-05-29T19:45:25.848 [Information] Name of output text file: <outputTxtFile>
2019-05-29T19:45:25.848 [Information] Name of output PDF file: <outputPdfFile>
2019-05-29T19:45:26.200 [Information] Adding OCR task <taskID> for <fileName> <size of
fileName>...
```

Storage Explorer からご使用のローカル マシンに出力ファイルをダウンロードするには、まず対象のファイルを選択した後、上部のリボンにある **[Download](ダウンロード)** を選択します。

ヒント

ダウンロードしたファイルは、PDF リーダーで開くと、検索できます。

リソースをクリーンアップする

ジョブがスケジュールされていない場合でも、ノードの実行中はプールに対して料金が発生します。 プールは不要になったら、削除してください。 アカウント ビューで、**[プール]** およびプールの名前を選択します。 次に、 **[削除]** を選択します。 プールを削除すると、ノード上のタスク出力はすべて削除されます。 ただし、出力ファイルはストレージ アカウントに残ります。 Batch アカウントとストレージ アカウントも、不要になったら削除できます。

次のステップ

このチュートリアルでは、次の作業を行う方法を学びました。

Batch Explorer を使用して、プールおよびジョブを作成する
Storage Explorer を使用して、BLOB コンテナーと Shared Access Signature (SAS) を作成する
BLOB によってトリガーされる Azure 関数を作成する
Storage に入力ファイルをアップロードする
タスクの実行を監視する
出力ファイルを取得する

Batch Explorer で利用できるレンダリング アプリケーションを [**ギャラリー**] セクションで探して次に進みます。 アプリケーションごとにいくつかのテンプレートが用意され、今後も拡充されていく予定です。 たとえば Blender については、単一の画像を複数のタイルに分割することで 1 つの画像を構成する各部分を並列にレンダリングできるテンプレートが存在します。

.NET API を使用して Batch ワークロードのスケジュール設定と処理を行う他の例については、GitHub のサンプルを参照してください。

Batch C# のサンプル

このページはお役に立ちましたか?

Yes No