

Visual Studioで作ったDjangoアプリをAzure App Serviceで動かす

Python

Django

Azure

Visual Studio

Visual Studio 2015だとDjangoアプリをAzure App ServiceにPublishするだけで動くんですが、Visual Studio 2017、2019では色々手動で設定する必要があります。超面倒です。忘れないよう記録しておきます。

この情報は2019年1月時点の情報です。今後Azureの仕様が変更される可能性があります。

結論

いろいろ面倒くさいです。今はここまでしてDjangoでAzure App Serviceを使わなくていいかも。

Azure App ServiceにPython 3.6をインストールする

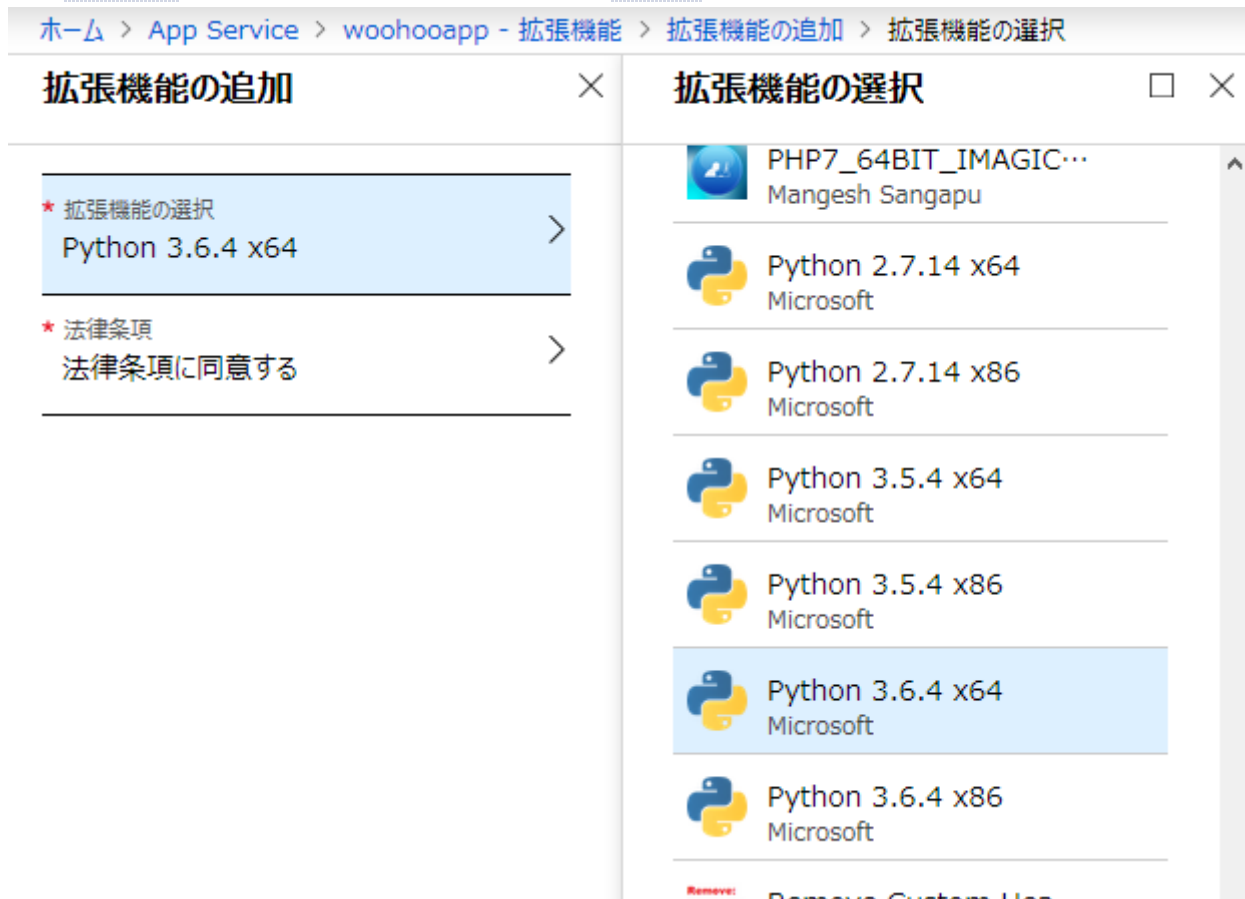
Azure App ServiceにデフォルトでインストールされているPythonのバージョンは3.4です。とりあえずPython 3.6をインストールしましょう。デフォルトのPython 3.4だとインストールフォルダに書き込めないのので以降の作業ができません。あと、新しい機能が動かなかったり、セキュリティ的にも不安です。

2018年の12月ごろにデフォルトPythonが3.6にバージョンアップされるというアナウンスがあったのですが、2019年1月現在まだ未反映の様です。もうちょっと彼らは頑張ってもらいたいです。

「拡張機能」 -> 「追加」を選んで



->「拡張機能の選択」からインストールしたいPythonを選択します。



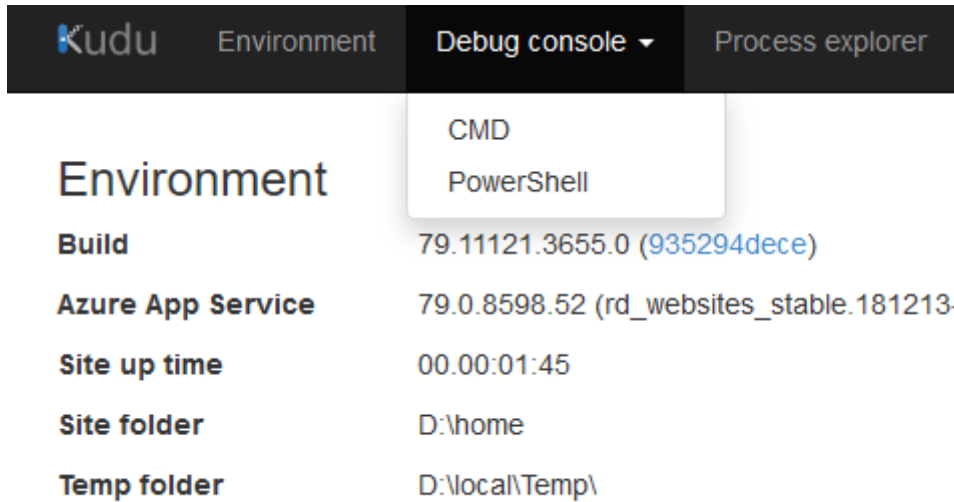
x64でもx86でもどちらでもいいと思います。ここではx64にしました。x86でメモリ不足が発生するプログラムなどは、Azure App Serviceでは到底動かないのでまあどちらでもいいと思います。

python.exeパスの確認

拡張機能でインストールしたPythonのパスが後で必要になるので確認しておきます。「高度なツール」->「移動」の順にクリックします。



「Kudu」が開くので、「Debug console」->「CMD」の順にクリックします。



コマンドプロンプト風の何かが表示されます。

```
Kudu Remote Execution Console
Type 'exit' then hit 'enter' to get a new CMD process.
Type 'cls' to clear the console

Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

D:\home>
```

カレントディレクトリはD:\homeになっていると思います。「ls」コマンドでpython.exeを探しましょう。

```
D:\home>ls
LogFiles
SiteExtensions
python364x64
site

D:\home>
```

今回の場合はpython.exeは「D:\home\python364x64\python.exe」にインストールされていました。このパスを後で使用するので覚えておきましょう。

```
D:\home>ls
LogFiles
SiteExtensions
python364x64
site

D:\home>ls python364x64
DLLs
Lib
pip.bat
python.exe
python3.dll
python36.dll
pythonw.exe
sitecustomize.py
vcruntime140.dll
wfastcgi.py

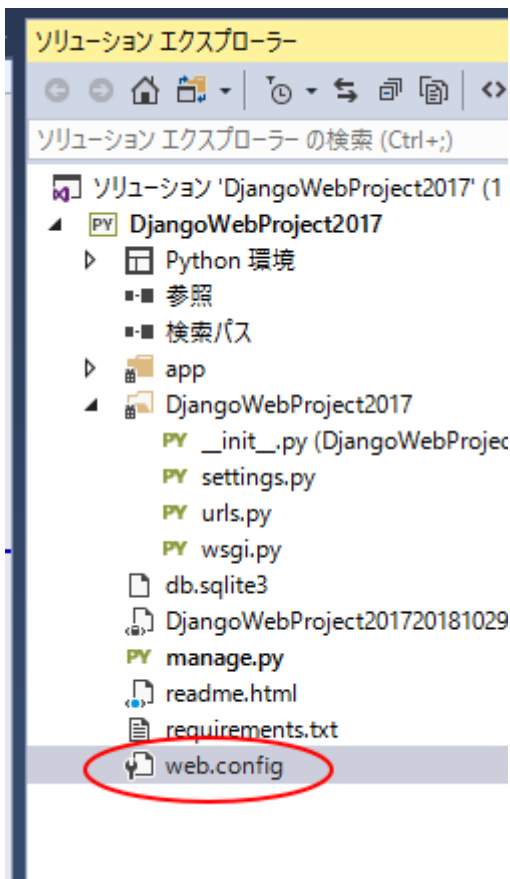
D:\home>
```

web.configの編集

IISからDjangoが呼び出されるよう、Azure側のIISの設定を変更してやる必要があります。普段IISなんて触らないであろうパイソニスタにはこんなのわけわからんと思います。

プロジェクトフォルダ直下にweb.configを作ります。既にある場合は一旦削除して作り直しましょう。

作り方は、Visual Studio のソリューション エクスプローラーで、プロジェクトを右クリックして、[追加] > [新しい項目] を選択します。表示されるダイアログ ボックスで、[Azure web.config (Fast CGI)] テンプレートを選択し、[OK] を選択します。



次の行を

```
<add key="WSGI_HANDLER" value="app.wsgi_app"/>
```

次の行で置き換えます。

```
<add key="WSGI_HANDLER" value="django.core.wsgi.get_wsgi_application()"/>
```

その下に次の行を挿入してDjangoの設定ファイルを指定します。「プロジェクト名」はプロジェクト名で置き換えてください。今回の例ではDjangoWebProject2017ですね。（Visual StudioでDjangoプロジェクトを作成すると、設定ファイル名はプロジェクト名になっているはずです。）

```
<add key="DJANGO_SETTINGS_MODULE" value="プロジェクト名.settings" />
```

以上編集後、次のような記述になります。

```
<appSettings>
  <add key="WSGI_HANDLER" value="django.core.wsgi.get_wsgi_application()"/>
  <add key="DJANGO_SETTINGS_MODULE" value="プロジェクト名.settings" />
  <add key="PYTHONPATH" value="D:\home\site\wwwroot"/>
  <add key="WSGI_LOG" value="D:\home\LogFiles\wfastcgi.log"/>
</appSettings>
```

handlersセクションのPython.exeのパスをさっき確認したパス

「D:\home\python364x64\python.exe|D:\home\python364x64\wfastcgi.py」に置き換えます。

```
<handlers>
  <add name="PythonHandler" path="*" verb="*" modules="FastCgiModule"
```

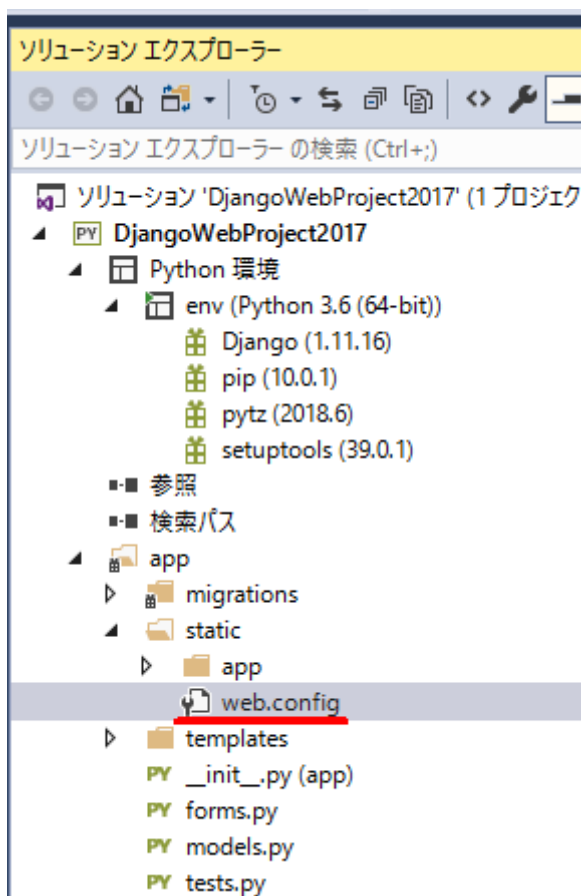
```
scriptProcessor="D:\home\python364x64\python.exe|D:\home\python364x64\wfastcgi.py"
resourceType="Unspecified" requireAccess="Script"/>
</handlers>
```

Djangoプロジェクトの settings.py ファイルで、Azure App Serviceのドメインを ALLOWED_HOSTS に忘れずに追加します。まあこれはAzureじゃなくてDjango特有の設定ですね。

```
ALLOWED_HOSTS = ['woohooapp.azurewebsites.net', 'localhost']
```

staticファイル転送用の設定

さらに面倒なのですが、staticファイルをIISに転送させるために、staticフォルダ直下にweb.configを作る必要があります。IISはフォルダごとに、設定ファイルが必要なようです。



web.configの内容は以下に置き換えます。

```
<configuration>
  <system.webServer>
    <handlers>
      <clear />
      <add
        name="StaticFile"
        path="*" verb="*"
        modules="StaticFileModule,DefaultDocumentModule,DirectoryListingModule"
        resourceType="Either"
        requireAccess="Read" />
    </handlers>
    <staticContent>
      <mimeTypeMap fileExtension="*" mimeType="application/octet-stream" />
    </staticContent>
```

```
</system.webServer>
</configuration>
```

ここで一旦サーバーに発行します。デプロイスロットを使用している場合は、発行先を間違えないように気を付けてください。別スロットに発行するとファイルが見つからなくて悩みます。

発行が完了すると、次のエラーメッセージが表示されますがそれでOKです。

```
The page cannot be displayed because an internal server error has occurred.
```

Pythonパッケージをインストールする

さて、先ほどのエラーはDjango等のパッケージがAzure側にインストールされていないため表示されています。これから手動で必要なPythonパッケージをインストールします。超面倒くさいですね。

[マイクロソフトのご案内](#)によると、手動でインストールしろとのことなので、とりあえずその通りやってみます。

ちなみにVisual Studio 2015の場合は、ローカルのvirtualenvのフォルダをそのままAzure側にアップロードして実行していたようなのですが、手動インストールに変更されたようです。他の環境からvirtualenvのフォルダをコピーしても動作する保証なんて全くありませんので、さすがに彼らもまずいと気付いたのでしょう。

パッケージをインストールするには、先ほど使用した「Kudu」の、「Debug console」->「CMD」を使います。

python.exeのインストールフォルダ(今回の場合はD:\home\python364x64)に移動して次のコマンドを実行します。

```
python -m pip install --upgrade -r /home/site/wwwroot/requirements.txt
```

requirements.txtには必要なpythonパッケージが列挙されています。requirements.txtの作成方法がわからなければググってください。

コマンドの実行画面は次のようになります。

```
D:\home>cd python364x64
```

```
D:\home\python364x64>python -m pip install --upgrade -r /home/site/wwwroot/requirements.txt
```

これで動くはず

さっきエラーメッセージが表示されたAzure App ServiceのURLをリロードするとDjangoのWebサイトが表示されるはずです。staticファイルがダウンロードできるかも確認しましょう。表示されない場合は一度Azure App Serviceを再起動してみると表示されるかもしれません。

ちなみに、アプリケーション設定画面の「プラットフォーム」は32bitにしても64bitにしても関係なく64bitのpythonが動きます。不思議です。「プラットフォーム」ってなんなんでしょうか？

概要

アクティビティ ログ

アクセス制御 (IAM)

タグ

問題の診断と解決

デプロイメント

クイック スタート

デプロイ スロット (プレビュー)

デプロイ スロット

デプロイ センター

設定

アプリケーション設定

認証/承認

Application Insights

ID

バックアップ

カスタム ドメイン

SSL 設定

ネットワーク

スケール アップ (App Service...)

スケール アウト (App Service...)

Web ジョブ

プッシュ

MySQL In App

全般設定

.NET Framework のバージョン v4.7

PHP のバージョン 5.6

App Service は、新しいバージョンの Python のインストールをサポートしています。詳しくは、ここをクリックしてください。

Python バージョン オフ

Java バージョン オフ

Java マイナー バージョン

Java Web コンテナ

プラットフォーム **アップグレードして有効にする**

Web ソケット オフ オン

常時接続 アップグレードして有効にする

マネージド パイプライン バージョン

統合 クラシック

HTTP バージョン

1.1 2.0

アフィニティ Cookie を無効にすると、ステートレス アプリケーションのパフォーマンスを向上させることができます。ステートフル アプリケーションの場合、互換性向上のためアフィニティ Cookie を有効にしたままにする必要があります。詳細については、ここをクリックしてください。

ARR アフィニティ

オフ オン

自動スワップ

アップグレードして有効にする

プラットフォームに関わらず 64bitのpythonが動くようです

所感その2

以上、いかがでしたか？超面倒くさくないですか？

IISを覚えろってことなのかも知れませんが、Nginxの方が私的には機能が豊富で便利なので移行したくないですね。しかも今後Pythonパッケージを追加するたびに、インストールコマンドを手打ちですからね。

私はAzure App Serviceでホスティングするのはやめて、普通にバーチャルマシン+Dockerを使うことにしました。今まであえてVisual StudioでPython開発に挑戦してたんですが、やっぱり普通にPyCharmがいいと思います。

さらに設定は続く

今回、拡張機能でインストールしたPythonのインスタンス数が16とかなり多めに設定されているという罠があって、メモリ消費多めのWebアプリかつAzure App Serviceの安いプランだと同時アクセスされた時にメモリがスワップして激遅になります。

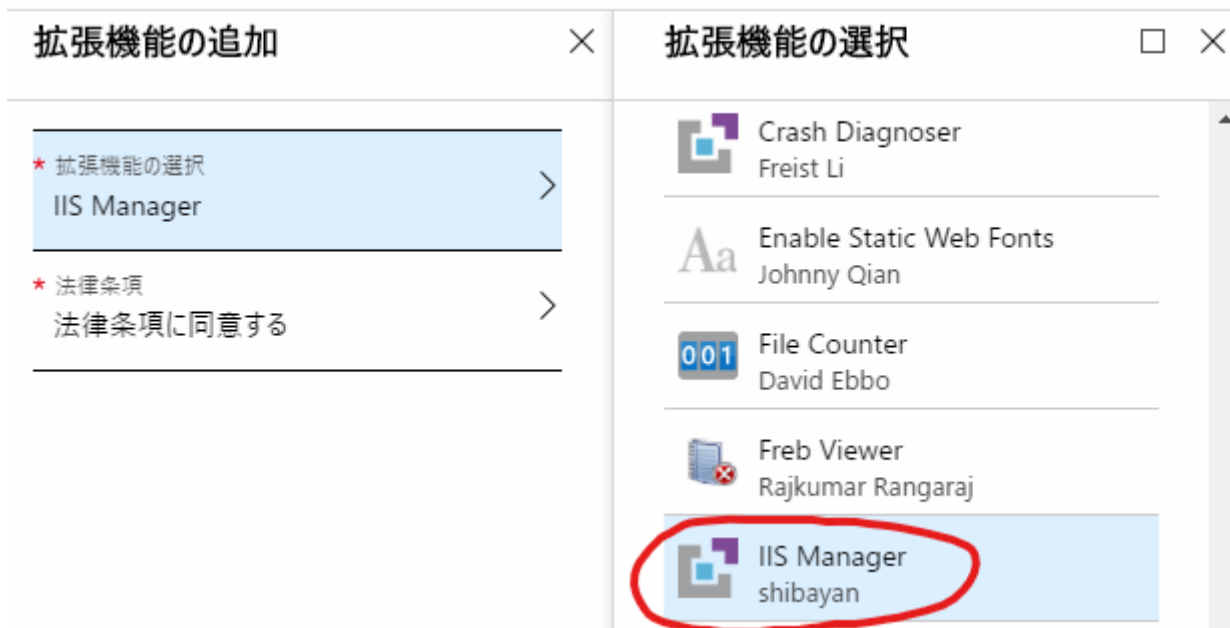
Azure App Serviceのメトリックだとスワップしているのかしてないのか、よくわからないので原因を突き止めるのに時間がかかりました。そういう点でもシステムの情報を把握しやすいDockerの方が生のAzure App Serviceより良いのではと思いました。

で、このインスタンス数を変更するのも面倒です。

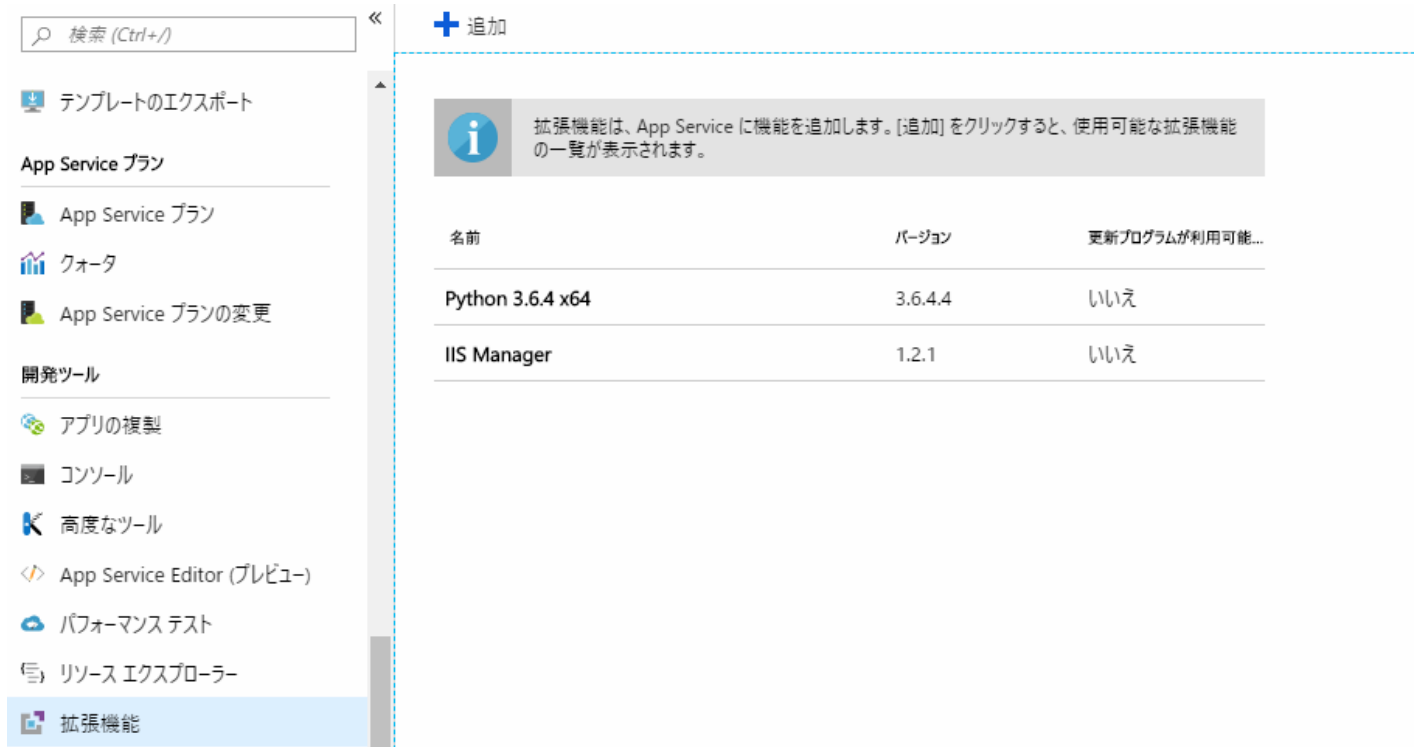
インスタンス数の設定

AzureのIISを設定するには、Windowsにインストールされている（or する）IISマネージャーを使ってAzureに接続することで以前はできてたらしいのですが、今は接続できません。正直全く意味がわかりませんが、AzureではIISを使うなというMicrosoftからのメッセージなののでしょうか。

現在、AzureのIISを設定するには、前述の「拡張機能」から「IIS Manager」を追加するのが手軽です。



IIS Managerをクリックして、





参照をクリックすると、

このような画面でIISの設定を直接変更できます。とりあえず作者のshibayanさんに感謝です。

IIS Manager
Search

Configuration Editor

applicationHost.config applicationHost.xdt

Choose Template...

Load

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <configuration>
3   <configSections>
4     <sectionGroup name="system.applicationHost">
5       <section name="applicationPools" allowDefinition="AppHostOnly" overrideModeDefault="Deny" />
6       <section name="configHistory" allowDefinition="AppHostOnly" overrideModeDefault="Deny" />
7       <section name="customMetadata" allowDefinition="AppHostOnly" overrideModeDefault="Deny" />
8       <section name="listenerAdapters" allowDefinition="AppHostOnly" overrideModeDefault="Deny" />
9       <section name="log" allowDefinition="AppHostOnly" overrideModeDefault="Deny" />
10      <section name="serviceAutoStartProviders" allowDefinition="AppHostOnly" overrideModeDefault="Deny" />
11      <section name="sites" allowDefinition="AppHostOnly" overrideModeDefault="Deny" />
12      <section name="webLimits" allowDefinition="AppHostOnly" overrideModeDefault="Deny" />
13    </sectionGroup>
14    <sectionGroup name="system.webServer">
15      <section name="asp" overrideModeDefault="Allow" />
16      <section name="caching" overrideModeDefault="Allow" />
17      <section name="cgi" overrideModeDefault="Allow" />
18      <section name="defaultDocument" overrideModeDefault="Allow" />
19      <section name="directoryBrowse" overrideModeDefault="Allow" />
20      <section name="fastCgi" allowDefinition="AppHostOnly" overrideModeDefault="Deny" />
21      <section name="globalModules" allowDefinition="AppHostOnly" overrideModeDefault="Deny" />
22      <section name="handlers" overrideModeDefault="Deny" />
23      <section name="httpCompression" overrideModeDefault="Allow" />
24      <section name="httpErrors" overrideModeDefault="Allow" />
25      <section name="httpLogging" overrideModeDefault="Deny" />
26      <section name="httpProtocol" overrideModeDefault="Allow" />
27      <section name="httpRedirect" overrideModeDefault="Allow" />
28      <section name="httpTracing" overrideModeDefault="Allow" />
29      <section name="isapiFilters" allowDefinition="MachineToApplication" overrideModeDefault="Deny" />
30      <section name="modules" allowDefinition="MachineToApplication" overrideModeDefault="Deny" />
31      <section name="applicationInitialization" allowDefinition="MachineToApplication" overrideModeDefault="Allow" />
32      <section name="odbcLogging" overrideModeDefault="Deny" />
33    </sectionGroup>
34    <sectionGroup name="security">
35      <section name="access" overrideModeDefault="Allow" />
36      <section name="applicationDependencies" overrideModeDefault="Deny" />
37    </sectionGroup>
38    <sectionGroup name="authentication">
39      <section name="anonymousAuthentication" overrideModeDefault="Allow" />
40      <section name="basicAuthentication" overrideModeDefault="Allow" />
41      <section name="clientCertificateMappingAuthentication" overrideModeDefault="Allow" />
42      <section name="digestAuthentication" overrideModeDefault="Allow" />
43      <section name="iisClientCertificateMappingAuthentication" overrideModeDefault="Deny" />
44      <section name="windowsAuthentication" overrideModeDefault="Allow" />
45    </sectionGroup>
46  </configuration>

```

© 2019 - IIS Manager

で、この設定の中に、今回「拡張機能」で追加したpython3.6.4のエントリがあるので、maxInstancesの値をお好みの値に変えてやればOKです。

```

<application fullPath="D:\home\python364x64\python.exe"
arguments="D:\home\python364x64\wfastcgi.py" maxInstances="16" idleTimeout="21600"
instanceMaxRequests="10000000" signalBeforeTerminateSeconds="60">

```

お好みの値をどうやって探ればいいのかはわかりません。誰か教えてください。前述のとおりAzure App Serviceのメトリックだとメモリ消費量とかが分かりにくいです。

参考にしたshibayanさんのページです。

blog.shibayan.jp

geroforce 2年前

関連記事

2020-11-28

Visual Studio CodeでCode Spell Checkerが有効にならないときの解決...

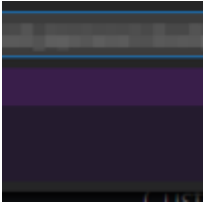
Code Spell CheckerというExtensionをインストールしたんですが...

2020-11-22

Visual Studio CodeでRemote SSHしようとしたら意味不明なエラーが出てつまず...

<https://geroforce.hatenablog.com/entry/2019/01/13/033719>

11/12



Windowsの「Visual Studio Code」に「Remote - SSH」というExte...

2019-10-20

Unity ML-AgentsをAMD製GPU + ROCm + TensorFlowで使う

Unity ML-AgentsはTensorFlowで実装されているんですが、Radeon...

コメントを書く