The following tutorial is for installing a Python app on windows in production using WSGI on Windows 10 64 bit.

Replace c:\my\_app with the path to the app

Flask production service setup: After we have a working flask-demo service we will install the flask service using either options:

1. On iis with wfastcgi
2. On Apache2 with mod\_wsgi

How to open command prompt as administrator:

start - cmd - right click on the Command Prompt and choose “Run as Administrator”

**Install Python 3.9 and my\_app and test it**

1. Run install.bat
2. Run the flask-demo service and test it
   1. cd /d c:\my\_app
   2. python app.py
   3. you should get the following message:

\* Serving Flask app "app" (lazy loading)

\* Environment: production

WARNING: This is a development server. Do not use it in a production deployment.

Use a production WSGI server instead.

\* Debug mode: off

\* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

* 1. Run the sanity tests below and then close the demo-service with CTRL+C

**Run sanity tests:**

1. Open <http://127.0.0.1:5000/> with your browser of choice
   1. You should get the PyWPS-4 example service page
2. Run the following commands in a command prompt and verify there are no errors:
   1. set service=<http://127.0.0.1:5000/>
   2. curl “%service%wps?service=WPS&request=execute&version=1.0.0&Identifier=say\_hello&storeExecuteResponse=true&DataInputs=name=World”
   3. curl -H "Content-type: xml" -X POST -d@c:\my\_app\static\requests\buffer.xml %service%wps

**Service installation on IIS using wfastcgi (Option A):**

Following this tutorial:

<https://stackoverflow.com/questions/58261464/how-to-host-python-3-7-flask-application-on-windows-server>

1. Run the following commands in a command prompt as administrator:
   1. python -m pip install wfastcgi
   2. wfastcgi-enable
2. Install CGI for IIS:
   1. Windows 10:
      1. Start -> Turn Windows features on or off
      2. Check CGI: Internet information service - World Wide Web Services - Application Development Features - CGI
   2. Windows Server:
      1. Open: Add roles and features Wizard
      2. From the left pane choose: Web Server Role (IIS)
      3. Check CGI on middle pane: Application development - CGI
3. Set permissions: You need to assign “iusr” and “iis\_iusrs” users full permissions for to the flask site folder and python folder:
   1. Open file explorer, right click the dir c:\my\_app (your site folder)
      1. Properties - Security - Click Edit - Click Add
         1. iusr;Iis\_iusrs
         2. OK
      2. Click on each of these users from the list and assign “Full Control”
      3. OK
      4. OK
   2. Do the same for C:\Python39 (your python folder)
4. Open IIS (Start - IIS)
5. right-click on the server name on the left pane and select add site.
   1. Site name:
      1. my\_app
   2. Physical path:
      1. c:\my\_app
   3. Port:
      1. 5000 (We’ll stick with the default flask port but you may change it)
   4. Host name:
      1. I left it blank but you may enter your host name
   5. OK
6. After adding site select the site name and open the handler mapping feature from the middle pane.
7. Click on “Add Module Mapping” from the right pane
   1. Request path:
      1. \*
   2. Module:
      1. FastCgiModule
   3. Executable - Insert the following line, modify it if your Python is on a different path:
      1. C:\Python39\python.exe|C:\Python39\Lib\site-packages\wfastcgi.py
   4. Name:
      1. my\_app
   5. Click “Request Restrictions”
      1. Make sure “Invoke handler only if request is mapped to:” checkbox is unchecked
      2. OK
   6. OK
   7. Do you want to create FastCGI…
      1. Yes
8. now go back and again select the server name and select “FastCGI Settings” from the middle pane.
   1. Double click the line we added with the python path
   2. Click the “…” for the Environment Variables (Collection) to launch the EnvironmentVariables Collection Editor
      1. Press Add button
         1. Name:
            1. PYTHONPATH
         2. Value:
            1. c:\my\_app
      2. Press Add button
         1. Name:
            1. WSGI\_HANDLER
         2. Value:
            1. app.app
            2. Remark: my Flask app module is named “app.py” and the application name is “app” so the value is app.app — if your module is named site.py and the application name is “application” it would be site.application or similar.
9. Browse to localhost:5000 (or to whichever host:port you selected)
   1. If you get an error message try to run “iisreset” from an administrator command line (sometimes you need to run it iisreset twice if you get permissions error)
10. Run the sanity tests above
11. Done!

**Service installation on Apache2 using mod\_wsgi (Option B):**

1. Following the guides in the following links:
   1. <https://flask.palletsprojects.com/en/1.1.x/deploying/mod_wsgi/>
   2. <https://github.com/GrahamDumpleton/mod_wsgi/blob/develop/win32/README.rst>
2. Download the latest Apache 2.X Win64, I used Apache 2.4.46 Win64
   1. <https://www.apachelounge.com/download/VC15/>
3. Extract Apache24 into C:\Apache24
4. Edit c:\Apache24\conf\httpd.conf:
   1. Change “Define SRVROOT” (unless you installed to the default C:\Apache24)
      1. Define SRVROOT "c:/Apache24"
   2. Replace “Listen 80” line with the following (you may replace 5000 with your port of choice):
      1. Define SRVPORT 5000
      2. Listen ${SRVPORT}
   3. Run the following command from a command line and Copy the output and paste it after the last LoadModule line
      1. mod\_wsgi-express module-config
      2. For my setup with the official Python 3.9 the output was:

LoadFile "C:/Python39/Python39.dll"

LoadModule wsgi\_module "C:/Python39/lib/site-packages/mod\_wsgi/server/mod\_wsgi.cp38-win\_amd64.pyd"

WSGIPythonHome "C:/Python39"

* 1. Find the line # Virtual hosts, and add below it:
     1. Include conf/extra/httpd-vhosts-pywps.conf

1. Create the file c:\Apache24\conf\extra\httpd-vhosts-pywps.conf with the contents:

WSGIApplicationGroup %{GLOBAL}

Define APPROOT "c:\my\_app"

<VirtualHost localhost:${SRVPORT}>

LogLevel info

ErrorLog "logs/my\_app\_error.log"

ServerName localhost

ServerAdmin localhost

WSGIScriptAlias / ${APPROOT}\app.py

<Directory ${APPROOT}>

#AllowOverride None

Require all granted

</Directory>

</VirtualHost>

1. Run the following command:
   1. c:\Apache24\bin\httpd.exe
2. Browse to localhost:5000 (or to whichever host:port you selected)
3. Installing Apache as a Service (instead of running c:\Apache24\bin\httpd.exe)
   1. Run the following commands in a command prompt as administrator:
   2. httpd.exe -k install
   3. Advise <http://httpd.apache.org/docs/2.4/platform/windows.html> for further details
4. Run the sanity tests above
5. Done!