This is not the code write up but the logic used to create the workflow.

**Python:**

Python was used as it’s the basic scripting language for ArcGIS. This gives us the ability to leverage sample scripts to access the GIS commands. Python also has the ability to work with internet REST services and process the data with external libraries. The scripts are basically text files allowing them to be easily modified.

**Starting with the site locations:**

To start I was given a point shapefile of the testing sites. It had the locations Lat, Lng, site name, and misc. notes. The coordinates were in Lat. Lng decimal degrees, WGS-84 projection. I added a field “link” and calculated its value from the FID field, knowing the primary key was unique. I used this field in all the steps to allow the tables to be joined together later. There was some thought to using the site name but contents of this field raised issues with commas and periods in the values. The next step was to export the table and bring it into excel. Here I was able to create a simple CSV file I could have python step through.

For the latest run my CSV file was just ID(Link), Site (Name), Lat, Long, without a header line, basically just a list of sites with the fields separated by commas, each location was a separate line in the file.

Example:

287,42SRBCWQ\_WQX-PIPE003.0-4078,40.09952469,-78.33578175

331,USGS-01554260,40.81359497,-76.40461902

353,42SRBCWQ\_WQX-1966,41.99167,-76.58917

371,42SRBCWQ\_WQX-SUSQ077.1-4076,40.3396703,-76.91575096

373,42SRBCWQ\_WQX-SUSQ077.2-4076,40.3365498,-76.91570843

98,21PA\_WQX-WQN0739,39.85305525,-79.92700302

**Watershed Delineation:**

Based on the project area hitting several states, it was not feasible to gather together the larger number of elevation files (DEMs) and prepare them. The amount of storage needed and the processing power and time needed was not practical with the number of locations to process. A quick Google search showed the USGS StreamStats site (<https://www.usgs.gov/mission-areas/water-resources/science/streamstats?qt-science_center_objects=0#qt-science_center_objects>) had an on-line interface that was based on a web service, If You encode a url request it will return a JSON file that would have the watershed delineation coordinates. With further processing I could create a GeoJSON file. For a few locations we had to use the web interface to place the pour point, this manual process created a JSON file that also needed to be parsed to create a GeoJSON file.

To do this I used WatershedDelination.py as my python script. My script loads the extra libraries it needs. Reads a line of the CSV file, breaks it up into 4 variables, for id, site, lat, lng. The it uses then forms the url string it asses to the site. Then it takes the response JSON and forms a GeoJSON file. Named gj + ID +.geojson.

Only issue I ran into was the script would fail every once in a while. I believe it was my local internet connection dropping but it could be something from the USGS site. Another thought was it was a memory issue as it hit a larger watershed and had a large data response. Needless it was easier to restart the script than to troubleshoot it. I first removed all the sites the processed successfully from the CSV before I restarted the script. I didn’t want two files for the same site. My script told you the last file it processed successfully. What was odd is sometimes it failed on the first point, and just rerunning the python script was all you needed to do. That is why I’m leaning towards my internet connection as being faulty.

The next step was to convert all the GeoJSON files to ShapeFile format. For this I used the GDAL library.

**GDAL/OGR:**

The Geospatial Data Abstraction Library is a computer software library for reading and writing raster and vector geospatial data formats Having previously installed QGIS, I had access to the OGR tools. This allowed me using the windows command interface convert 2175 GeoJSON files in a folder to ShapeFiles with one line.

*for /R %f in (\*.geojson) do ogr2ogr -f "ESRI Shapefile" "%~dpnf.shp" "%f"*

This process only took a couple minutes. Now we had a folder of the watersheds as separate features.