



Deploying Apps and Services using ArcGIS API for Python

Akhil Negi, Atma Mani, Divyansh Jha

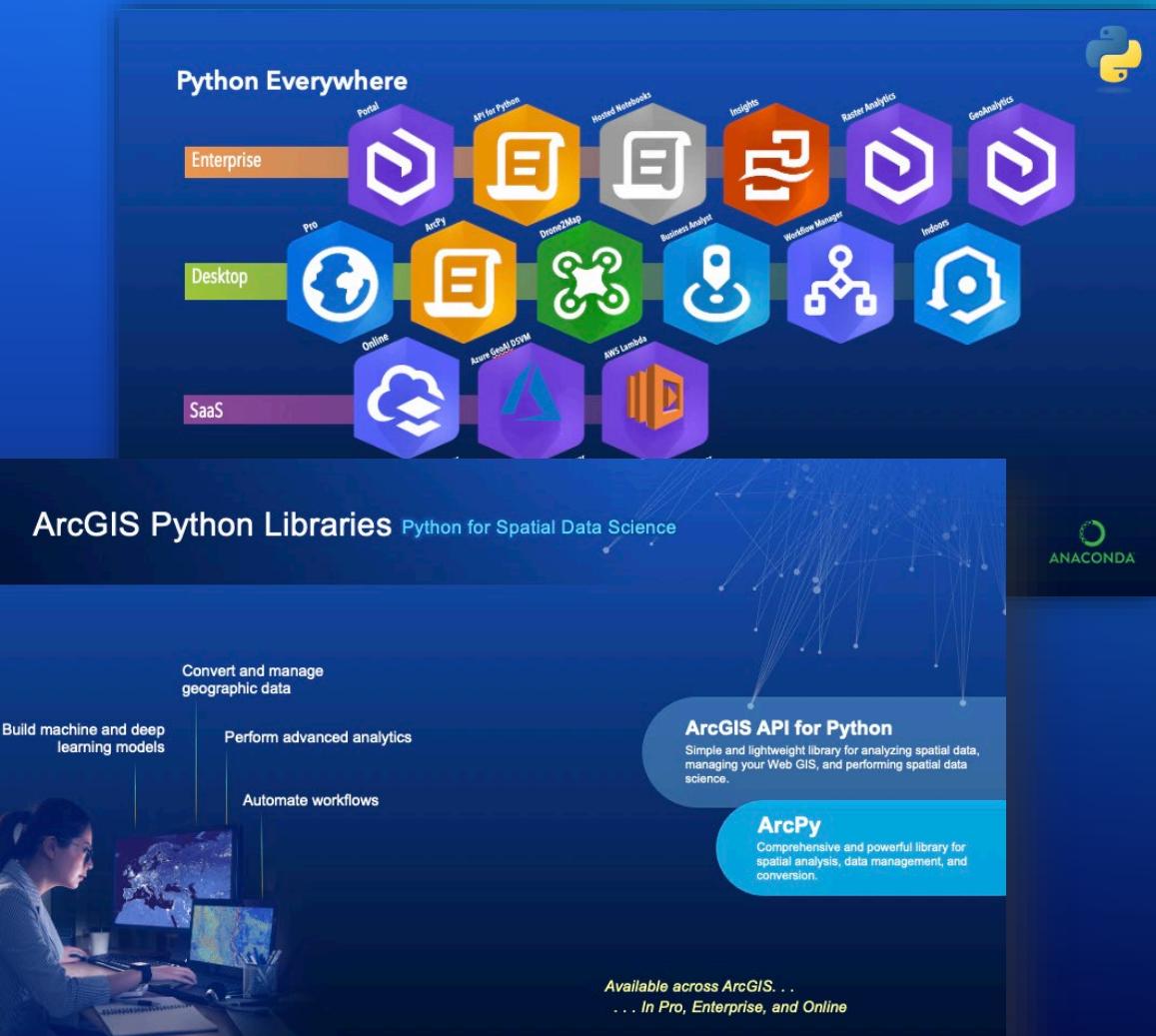
*2021 ESRI
DEVELOPER SUMMIT*

Outline of this talk

- **Introduction**
- **Deploying apps**
 - The `arcgis.apps` module
 - Deploying ArcGIS Survey123 and ArcGIS Dashboard apps
- **Deploying services**
 - Services within the ArcGIS Ecosystem (GP service)
 - Services in your own infrastructure
 - Services in cloud infrastructure
- **Deploying scalable services on FAAS infrastructure**
 - Introduction – understanding deployment packages & API gateways
 - Deploying deep learning model inferencing tools
- Conclusion

Python in ArcGIS

- Python is the preferred scripting language for ArcGIS
- ArcGIS API for Python & ArcPy – 2 powerful Python libraries
- ArcGIS API for Python – unique features
 - pure Python library
 - OS agnostic
 - light weight & modular
 - free of cost



Deploying apps using ArcGIS API for Python

Divyansh Jha

ArcGIS API for Python

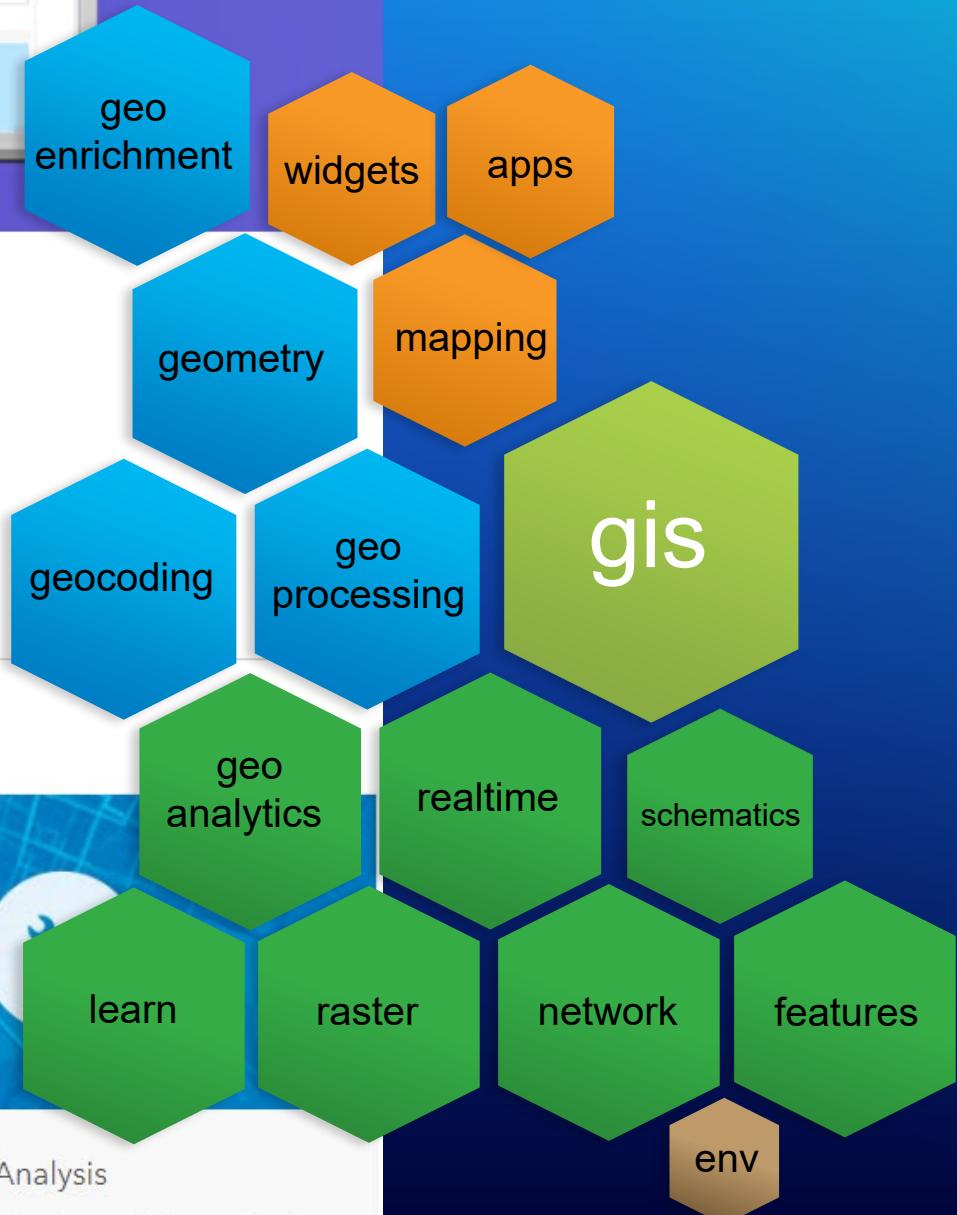
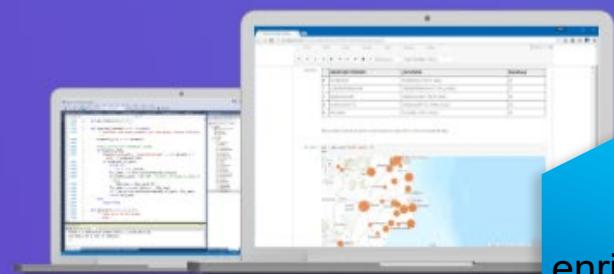
[Install the API](#)

Version 1.6.1 · May 16, 2019

[Home](#) [Guide](#) [Sample Notebooks](#) [API Reference](#) [Community](#)

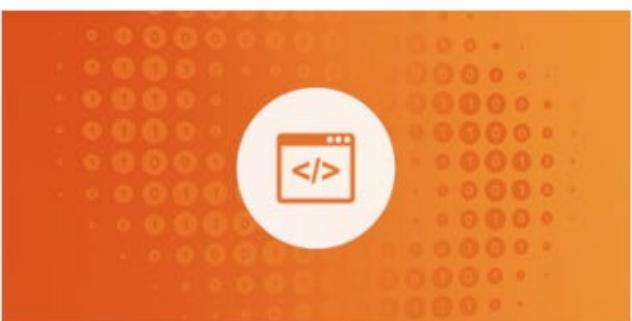
A powerful Python library for spatial analysis, mapping and GIS

ArcGIS API for Python is a Python library for working with maps and geospatial data, powered by web GIS. It provides simple and efficient tools for sophisticated vector and raster analysis, geocoding, map making, routing and directions, as well as for organizing and managing a GIS with users, groups and information items. In addition to working with your own data, the library enables access to ready to use maps and curated geographic data from Esri and other authoritative sources. It also integrates well with the scientific Python ecosystem and includes rich support for Pandas and Jupyter notebook.

[Install the API](#) | [Get started](#) | [View samples](#)

Understand your GIS

This "hello world" style notebook shows how to get started with the GIS and visualize its contents.



Manage your GIS

The ArcGIS API for Python provides APIs and samples for ArcGIS Online administrators to manage their online



Perform Spatial Analysis

Cell sophisticated spatial analysis tools that work with online content, using a few lines of code.



1.8.4

Search docs

ARCGIS

- arcgis.gis module
- arcgis.env module
- arcgis.features module
- arcgis.raster module
- arcgis.network module
- arcgis.geoanalytics module
- arcgis.geocoding module
- arcgis.geoenrichment module
- arcgis.geometry module
- arcgis.geoprocessing module
- arcgis.mapping module
- arcgis.realtime module
- arcgis.schematics module
- arcgis.widgets module
- arcgis.apps module
- arcgis.learn module

API Reference for the ArcGIS API for Python

Contents:

arcgis

- [arcgis.gis module](#)
 - [GIS](#)
 - [Item](#)
 - [User](#)
 - [Group](#)
 - [Datastore](#)
 - [Role](#)
 - [Layer](#)
 - [GroupApplication](#)
 - [CategorySchemaManager](#)
 - [ContentManager](#)
 - [UserManager](#)
 - [GroupManager](#)
 - [GroupMigrationManager](#)
 - [DatastoreManager](#)
 - [RoleManager](#)
 - [ResourceManager](#)
 - [APIKeyManager](#)
 - [APIKey](#)
 - [ProfileManager](#)
 - [InvitationManager](#)
 - [CertificateManager](#)
 - [PortalDataStore](#)
 - [Submodules](#)
- [arcgis.gis.admin module](#)

ArcGIS API for Python

- > Power users / Developers
- > Org Administrators
- > GIS analysts and data scientists
- > Content publishers

Samples

Samples presented here demonstrate various features of the ArcGIS API for Python. The samples are categorized by the user profile they are most relevant to. Most samples are in the form of a [Jupyter Notebook](#), that can be viewed online, or downloaded and run interactively. A few samples are provided as stand-alone Python scripts in the accompanying GitHub SDK repository.

Download and run the sample notebooks

[Download as an archive](#)[Clone the GitHub repository](#)

To run the sample notebooks locally, you need the ArcGIS API for Python installed. See the [Install the ArcGIS API for Python](#) section in the Guide to learn how to download and install the API.

Once the API is installed, you can download the sample notebooks from this page.

Next, extract the archive if you downloaded as an archive or clone the GitHub repository. Then start Jupyter notebook application. Finally, open one of the sample notebooks.

ArcGIS API for Python

- > Get started
- > API Overview
- > The GIS
- > Working with geometries
- > Feature data and analysis
- > Introduction to Data Engineering in Python
- > Working with the Spatially Enabled DataFrame
- > Enriching GIS data with thematic information
- > Imagery and raster analysis
- > Working with big data
- > Using geoprocessing tools
- > Finding places with geocoding
- > Performing network analyses
- > Mapping and visualization

Guide

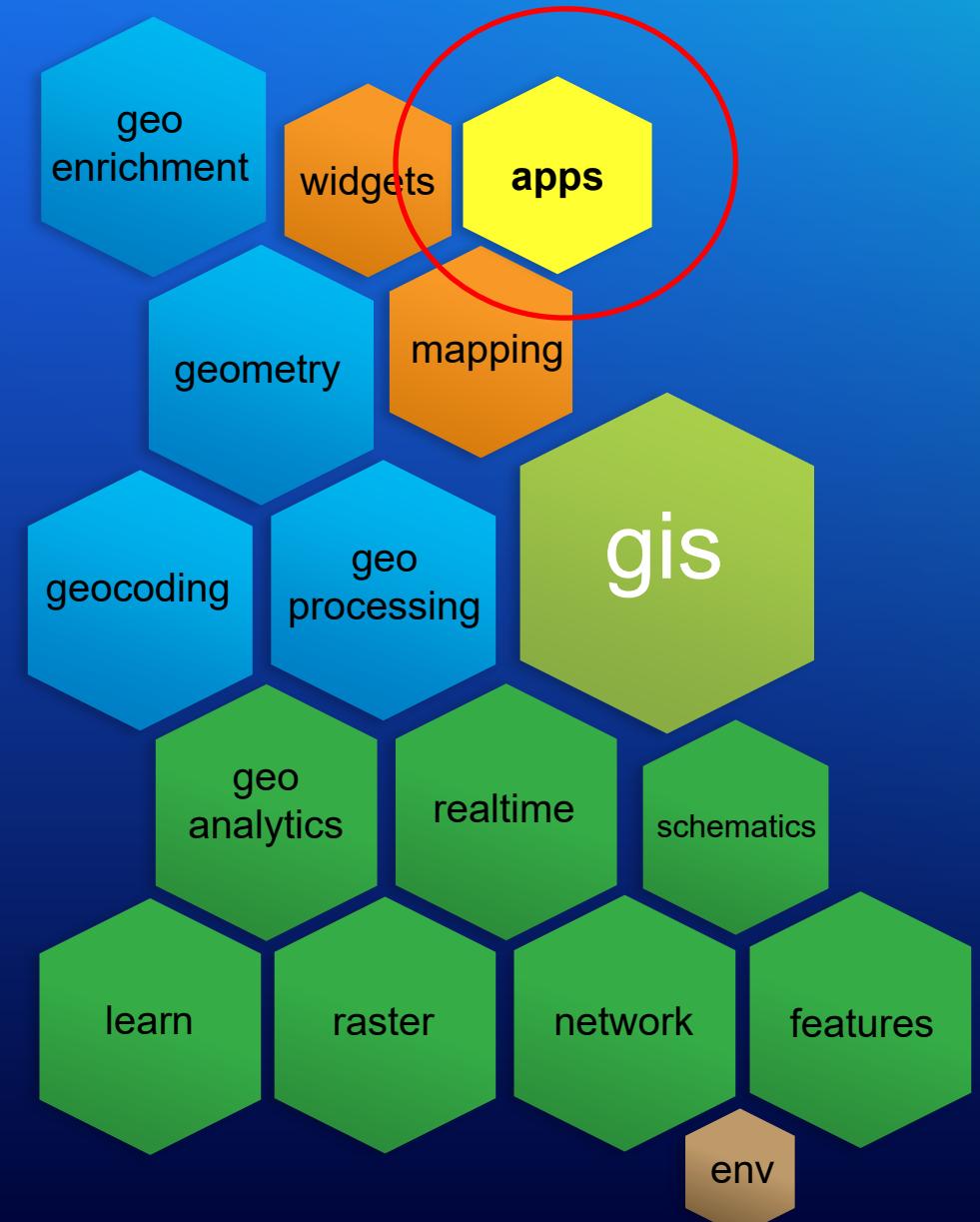
The ArcGIS API for Python is a powerful, modern and easy to use Pythonic library to perform GIS visualization and analysis, spatial data management and GIS system administration tasks that can run both in an interactive fashion, as well as using scripts.

It enables power users, system administrators and developers to leverage the rich SciPy ecosystem for automating their workflows and performing repetitive tasks using scripts. It integrates well with the Jupyter Notebook and enables academics, data scientists, GIS analysts and visualization enthusiasts to share geo-enriched literate programs and reproducible research with others.

This guide describes how to use the ArcGIS API for Python to write Python scripts, incorporating capabilities such as mapping, query, analysis, geocoding, routing, portal administration, and more. A great place to start developing once you've [installed the API](#) is to browse the [sample notebooks](#).

The `arcgis.apps` module

arcgis.apps module
build_collector_url
build_explorer_url
build_field_maps_url
build_navigator_url
build_survey123_url
build_tracker_url
build_workforce_url
Submodules
arcgis.apps.hub module
arcgis.apps.dashboard module
arcgis.apps.workforce module
arcgis.apps.storymap module
arcgis.apps.survey123 module
arcgis.apps.tracker module

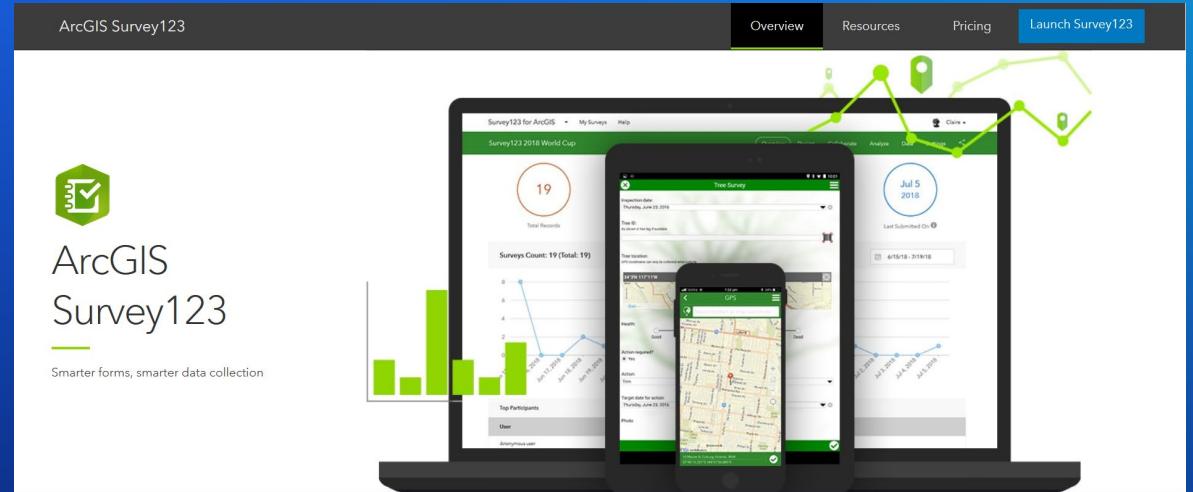


Survey123 for ArcGIS

Create surveys for field data collection

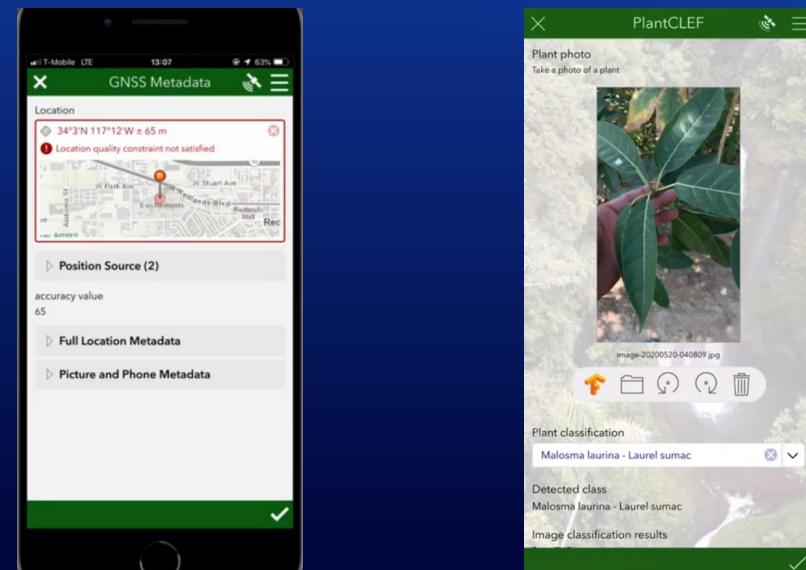
Features:

- Form-centric data collection solution.
- Smarter forms – Makes collections process easier.
- Supported on both web (Online) and mobile devices (Offline).
- Integrated with ArcGIS Online and Enterprise



Applications:

- Disaster response e.g. Building damage assessment
- Inspection of utilities e.g. pipes, pumps
- Environmental activities e.g. Survey of water cleaning





Demo: Tree Inventory

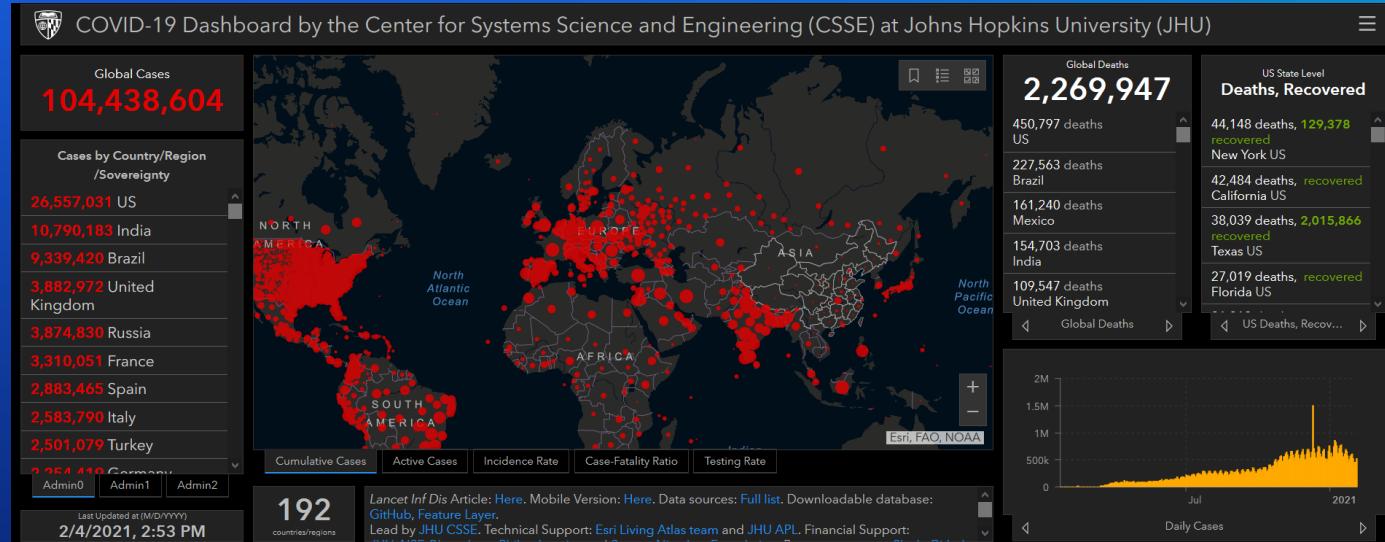
Field data collection with Survey123

ArcGIS Dashboards

Summarize your data using dashboards.

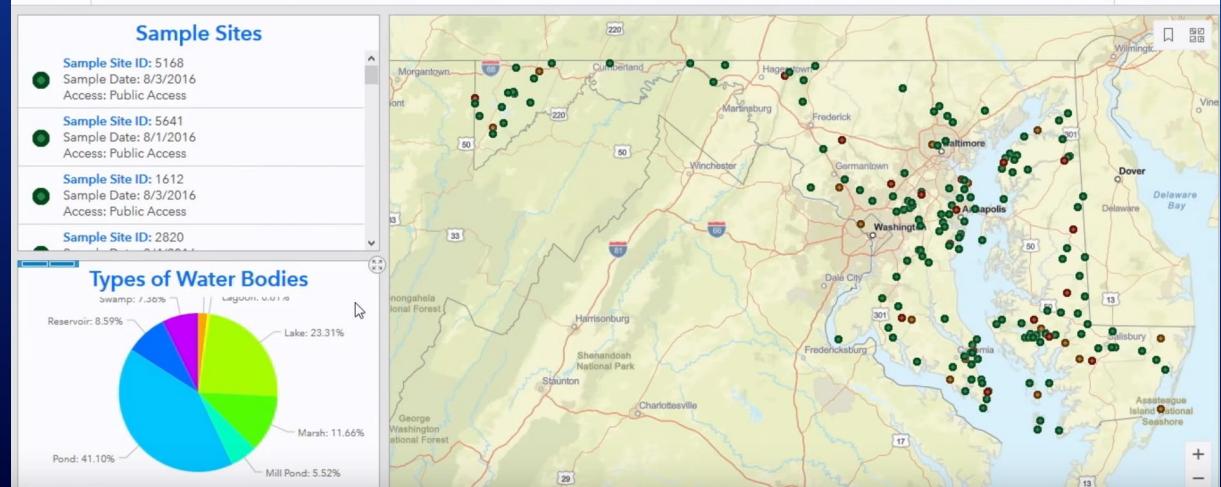
Features:

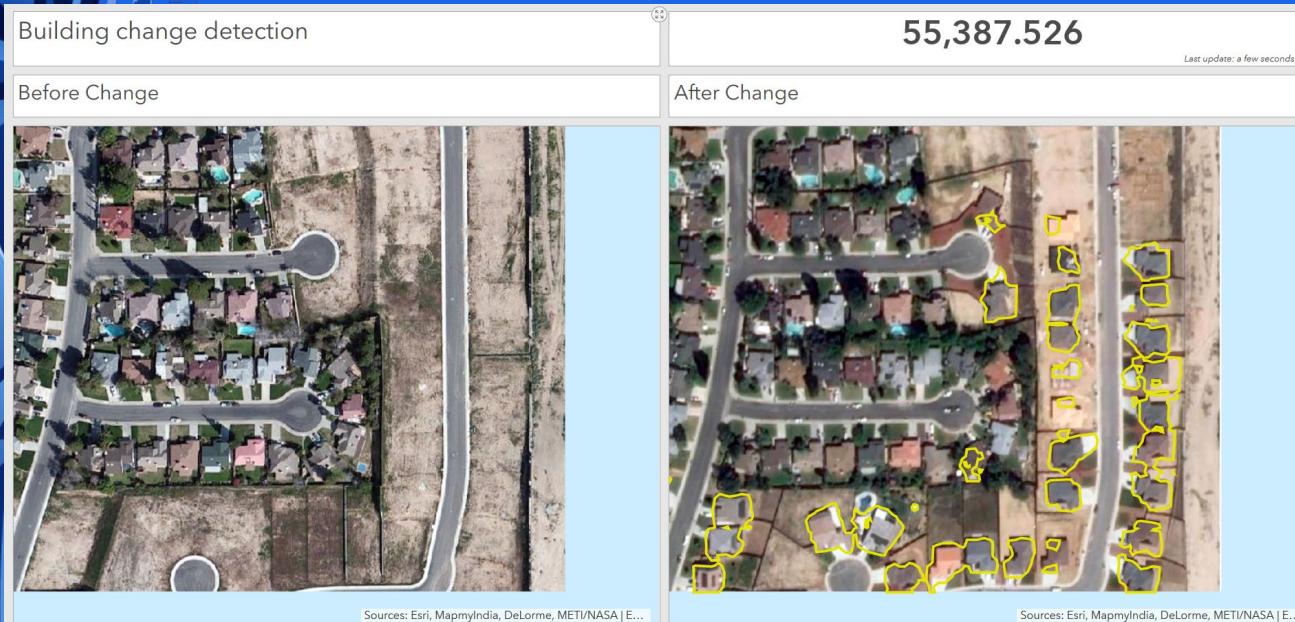
- Provides keys insights from your data on a single screen
- Provides ready to use data visualization using serial charts, pie charts, maps, embedded content etc.



Applications:

- Monitor and manage operations
- Visualize summaries/performance
- Emergency management





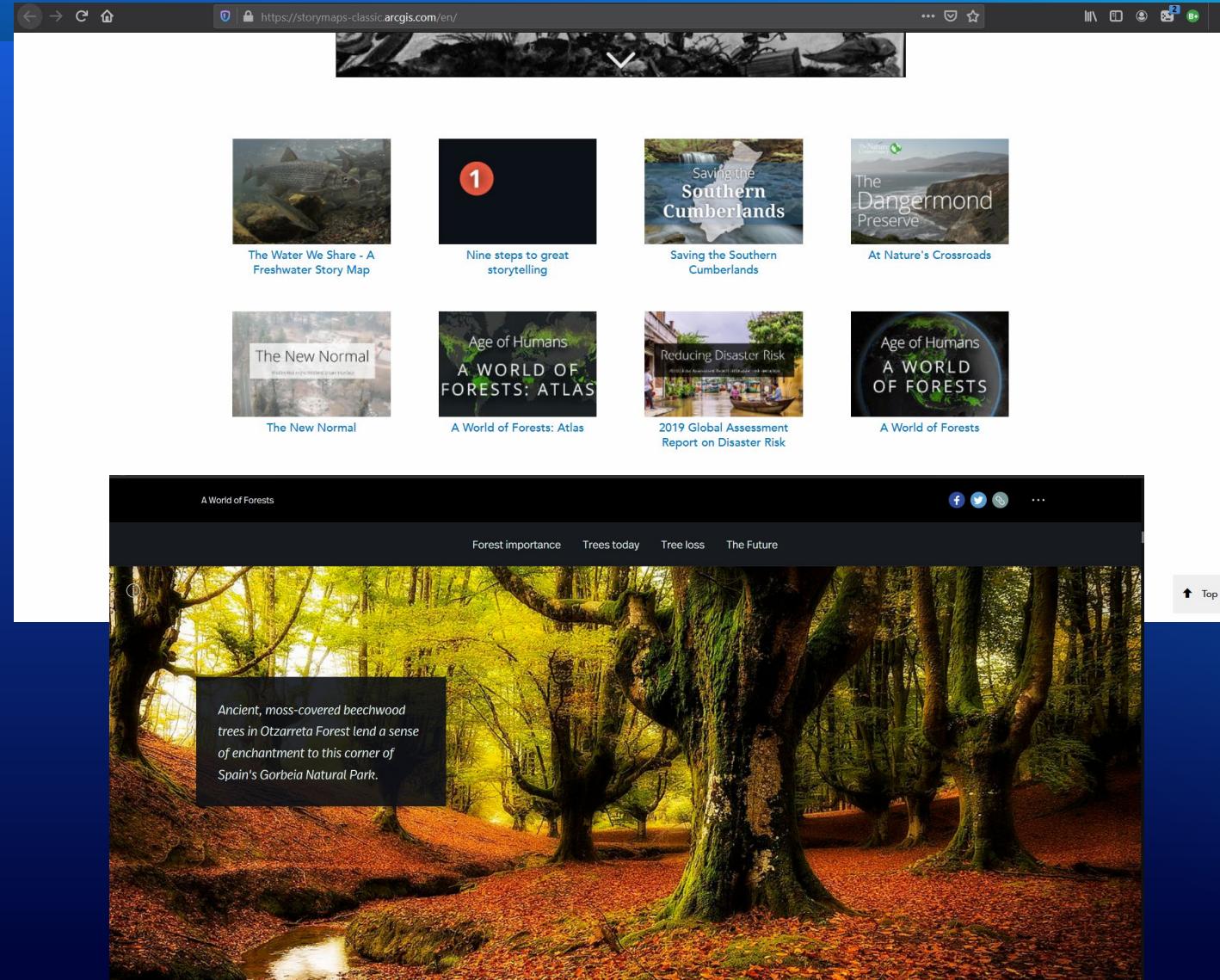
Demo: Dashboard for Change Detection

ArcGIS StoryMaps

Tell a story using ArcGIS Storymaps

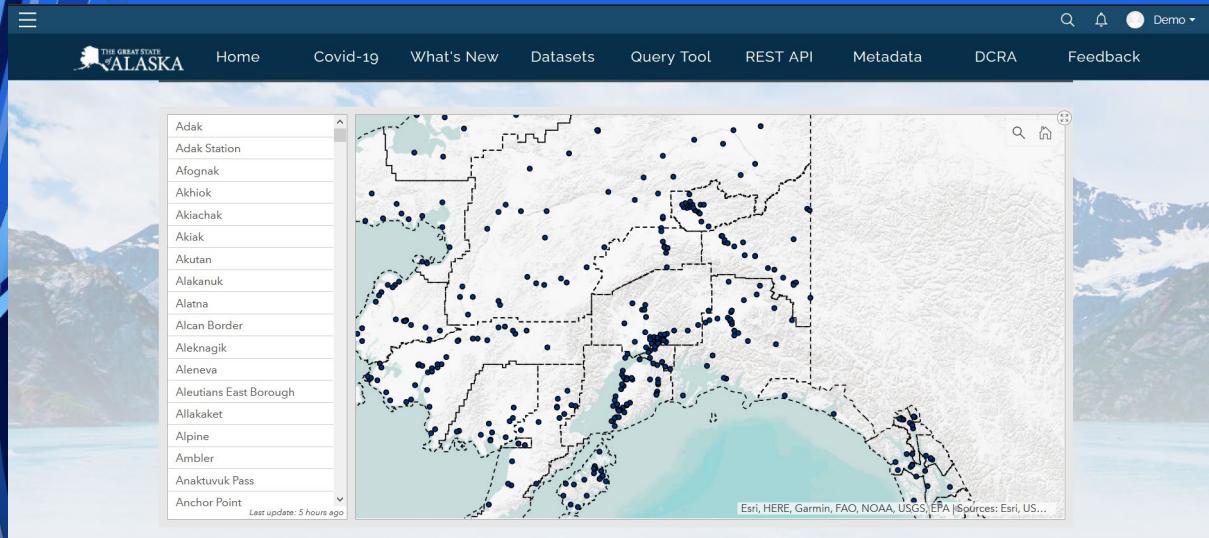
Features:

- Share your analysis in the form of a story.
- Embed dashboards, maps, pictures within the story.



Applications:

- Create convincing stories.
- Bring in multiple dashboards and maps in one place.



Demo:
Iteratively create and
update StoryMaps

Streamlit

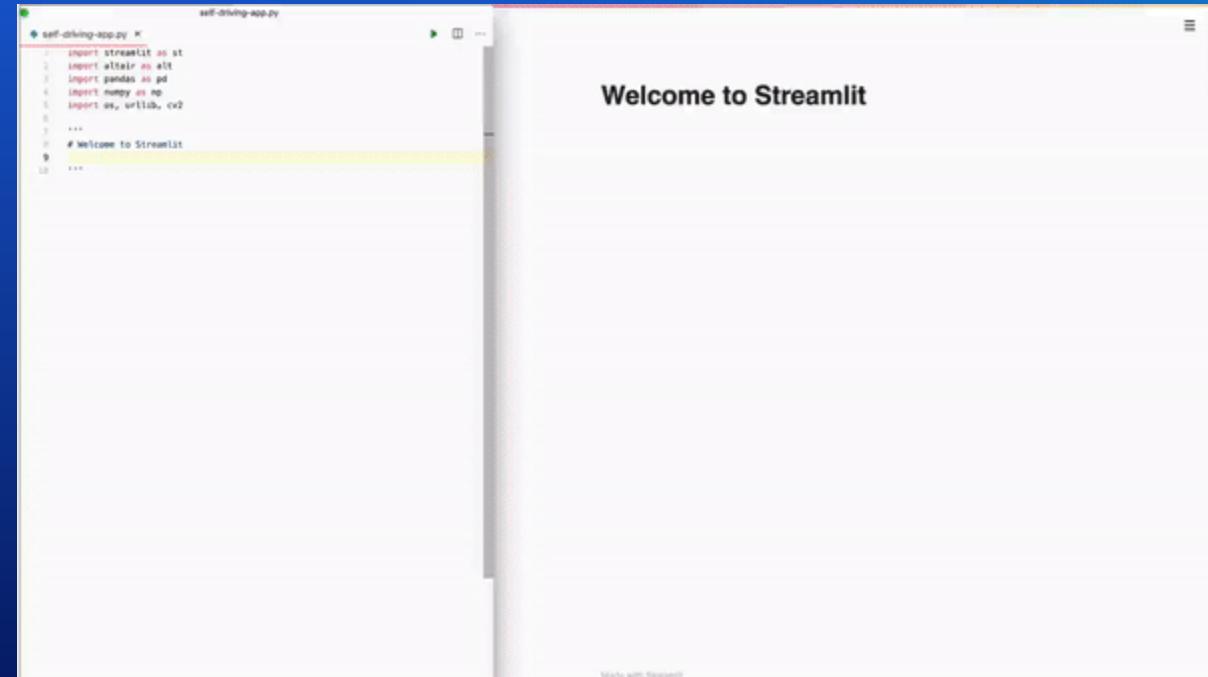
Create data driven webapps in Python. (<https://www.streamlit.io/>)

Features:

- Fast way to create webapps in few lines of Python code
- Easy to use API
- Can be deployed instantly

Applications:

- Data annotation
- Deep learning model deployment



The screenshot shows a Streamlit application interface titled "Geo-Referencing and Digitization of Scanned Maps". On the left, a sidebar titled "Select API" has a dropdown menu set to "Create Mask". Below it, "Adjust API Parameters" includes a "Color List" with sliders for R (100), G (168), and B (8). Other parameters include "Color delta" (0 to 100), "Closing Kernel" (1 to 10.0), "Opening Kernel" (1 to 10.0), and "Kernel type". The main area has two sections: "Initialization" (with a file upload section) and "Create Mask" (with a description and two images: a green landmass on a map and a white silhouette on a black background).

Demo: Integrate Streamlit with ArcGIS API for Python

Deploying Services

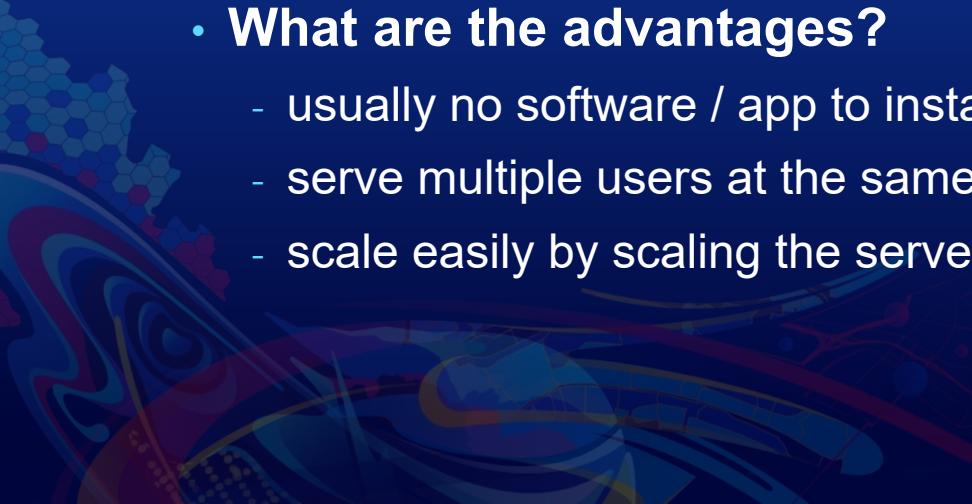
with the ArcGIS API for Python

Atma Mani

Deploying services

QUESTION

- **What is a web service?**
 - a pattern of deploying application using a client-server architecture
 - communication over the web using HTTP using RESTful APIs
- **How does a web service work?**
 - user interacts with the client (typically through a web app)
 - client passes web requests to the server
 - business logic and computation happen on the server
 - user receives the output via the client
- **What are the advantages?**
 - usually no software / app to install
 - serve multiple users at the same time
 - scale easily by scaling the server side



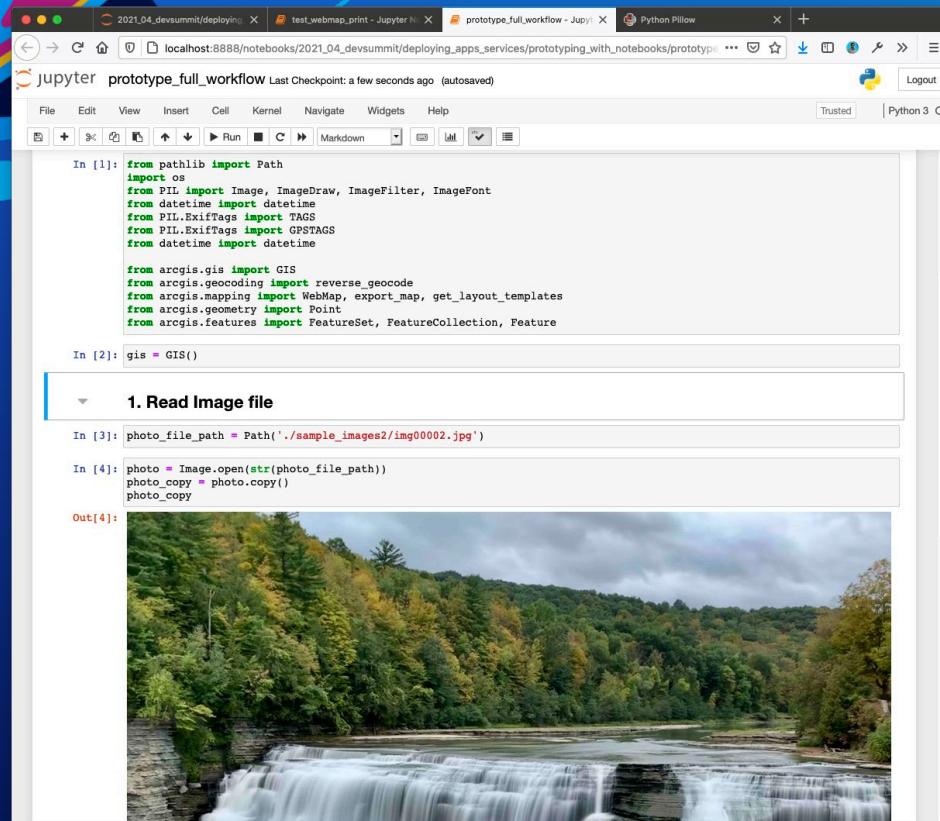
Where was this photo taken? – Web Service to Embed maps within photos

- Example problem: Have you ever wondered where a photo was taken?
- Solution:
 - Read EXIF tags (metadata) from JPG photos
 - Extract location coordinates
 - Make a map of the location
 - Reverse geocode to get the address / POI labels
 - Embed map and text back into the photo
- Approaches
 - Geoprocessing tool within ArcGIS Pro
 - GP service on ArcGIS Enterprise
 - Flask based service on your own infrastructure
 - Flask based service on cloud infrastructure



Deploying a GP Service

Atma Mani



```
In [1]: from pathlib import Path
        ...
        from PIL import Image, ImageDraw, ImageFilter, ImageFont
        from datetime import datetime
        from PIL.ExifTags import TAGS
        from PIL.ExifTags import GPSTAGS
        from datetime import datetime

        from arcgis.gis import GIS
        from arcgis.geocoding import reverse_geocode
        from arcgis.mapping import WebMap, export_map, get_layout_templates
        from arcgis.geometry import Point
        from arcgis.features import FeatureSet, FeatureCollection, Feature

In [2]: gis = GIS()

1. Read Image file

In [3]: photo_file_path = Path('./sample_images2/img00002.jpg')

In [4]: photo = Image.open(str(photo_file_path))
photo_copy = photo.copy()
photo_copy

Out[4]:
```

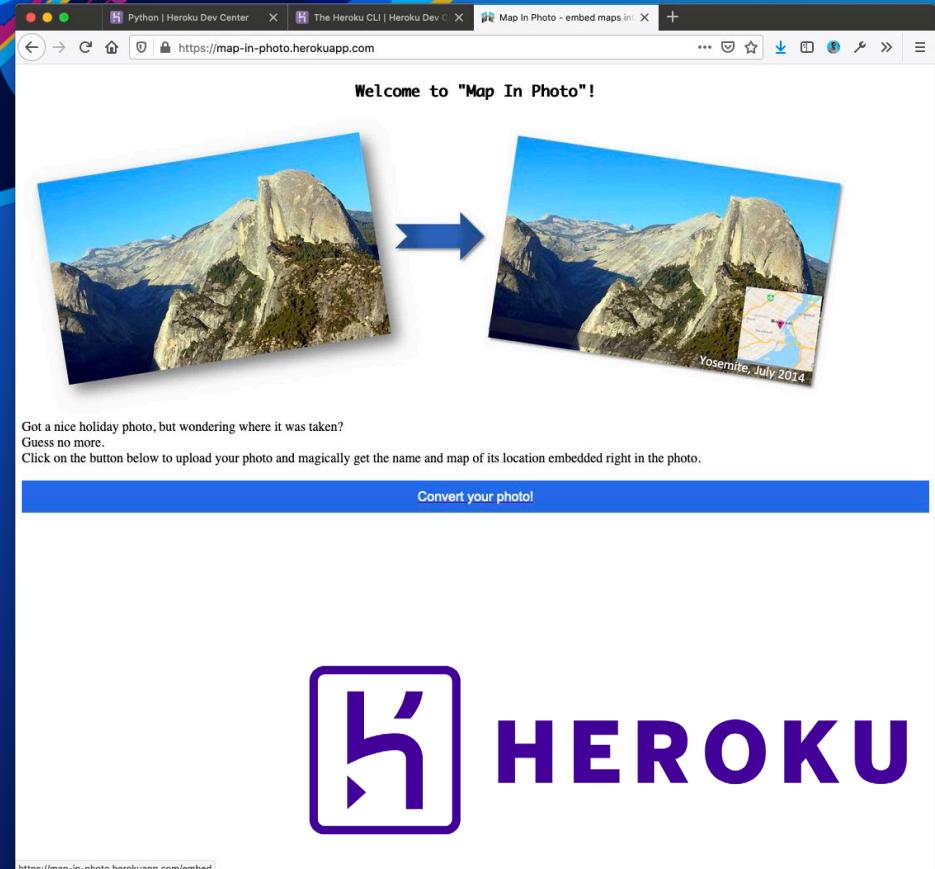


Deploying stand-alone services on your own infrastructure

Atma Mani

The screenshot shows the PyCharm IDE interface. The top navigation bar has tabs for 'File', 'Edit', 'Run', 'View', 'Code', 'Tools', 'Help', and 'File Structure'. The main window displays a Python file named 'app.py' with code for a Flask application. The code includes imports for Flask, os, and various utility functions like 'is_allowed_file', 'get_timestamp', 'index_page', 'embed_orchestrator', and 'output_file'. It defines routes for the root ('/'), embedding ('/embed'), and file download ('/outputs/<filename>'). The bottom part of the interface shows a terminal window with the output of running the Flask app. The terminal shows environment variables set (FLASK_APP=app.py, FLASK_ENV=development, FLASK_DEBUG=0), the app being served at port 5000, and several log entries indicating successful HTTP requests for static files like 'Landing2.jpg' and 'favicon.ico'. The status bar at the bottom right indicates the current file is 'map_in_photo - app.py'.

```
FLASK_APP=app.py
FLASK_ENV=development
FLASK_DEBUG=0
In folder: /Users/AtmaMani/Downloads/Documents/code/pychakras/map_in_photo
/usr/local/Cellar/python@3.8/3.8.5_1/Frameworks/Python.framework/Versions/3.8/libexec/bin/python3.8 -m flask run
  Serving Flask app "app.py"
    + Environment: development
    + Debug mode: off
[2021-02-07 23:21:39,889] WARNING in app: Connected to S3
+ Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [07/Feb/2021 23:21:34] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [07/Feb/2021 23:21:34] "GET /static/Landing2.jpg HTTP/1.1" 200 -
127.0.0.1 - - [07/Feb/2021 23:21:34] "GET /static/favicon.ico HTTP/1.1" 200 -
127.0.0.1 - - [07/Feb/2021 23:21:50] "GET /embed HTTP/1.1" 200 -
127.0.0.1 - - [07/Feb/2021 23:21:50] "GET /favicon.ico HTTP/1.1" 404 -
```



Deploying stand-alone services on cloud infrastructure

Atma Mani

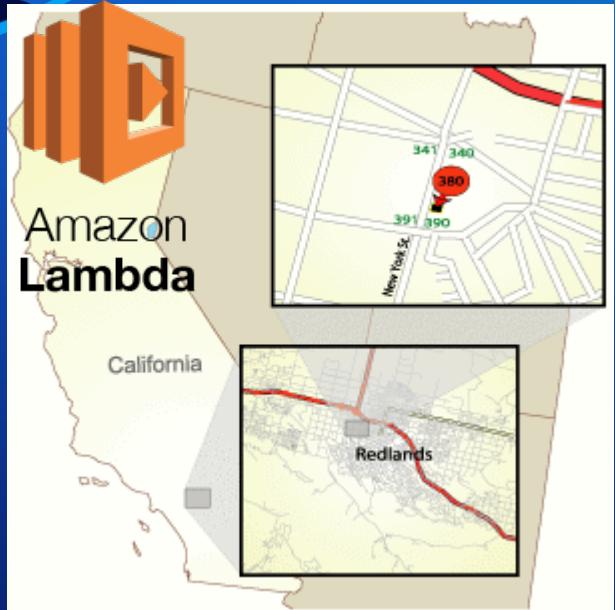
Deploying scalable services on FAAS infrastructure

with the ArcGIS API for Python

Akhil Negi

Deploying geocoding function as a services on cloud infrastructure.

Akhil Negi



2120, POCURU, ESTACIONAMIENTO 309, PROVIDENCIA, METROPOLITANA DE SANTIAGO	CL	1.000000
2681, MOSCONI GENERAL AV.	AR	1.000000
237, 백제고분로19길	KR	1.000000
2461, BONNY DR, COCOA, BREVARD, FL, 32926	US	1.000000
37, Schiefer, 8350	AT	0.997607
29, LATHAM STREET, YARRABILBA, QLD, 4207	AU	1.000000
118a, Hauptstraße, 8740	AT	1.000000
13A, BO LLANO, 39409	ES	1.000000
131, Поддубного, БОСТАНДЫКСКИЙ РАЙОН, АЛМАТЫ	KZ	1.000000
SN, CALLE 37, CIUDAD DEL CARMEN, Carmen, Campeche	MX	1.000000
12, Kościelna, Golubie, kartuski, pomorskie, 83-316	PL	1.000000
12, Werfelstraße, 1170	AT	1.000000
7, SCOTT STREET, MOUNT HOPE, NSW, 2877	AU	1.000000
5422, GROUSSAC, PAUL	AR	0.999999
17, Morelvej, 8920	DK	1.000000
212號, 三興路東勢段, 平鎮市, 東勢里	TW	1.000000
48, Częstochowska, Czarny Las, częstochowski, śląskie, 42-233	PL	1.000000
92H, Moste, Moste, Komenda, Osrednjeslovenska, 1218	SI	1.000000
5551, ECHEANDIA	AR	0.999999
16, Rue de la Republique, Amilly, 28300	FR	1.000000
6號, 東森路55巷, 楊梅市, 永寧里	TW	1.000000
4A, Serupvej, 8600	DK	1.000000
62, 18 DE JULIO (VILLA GUADALUPE), 15562	UY	1.000000
13, HEPNER PLACE, NORTH GEELONG, VIC, 3215	AU	1.000000

Deploying Address classification service on AWS Lambda

Akhil Negi

Recap

- We saw various ways to deploy Python scripts as services:
 - Services within ArcGIS Ecosystem – GP Service
 - Services on your own infrastructure – Flask based service
 - Services on cloud infrastructure – Heroku and AWS Lambda as an example.
- Saw ways to integrate various ArcGIS apps with ArcGIS API for python.
- Went through an example of integrating 3rd party application(Streamlit) with ArcGIS API for Python.

Learning resources

- Doc website: <https://developers.arcgis.com/python>
- SDK Repo: <https://github.com/esri/arcgis-python-api>
- Flask help: <https://flask.palletsprojects.com/>
- Python imaging library: <https://pillow.readthedocs.io/>
- Python API playlist: https://www.youtube.com/playlist?list=PLGZUzt4E4O2JaOMx_XZc85VdMlrqLGaVf
- Change detection sample notebook: <https://developers.arcgis.com/python/sample-notebooks/change-detection-of-buildings-from-satellite-imagery/>
- Address classification sample notebook: <https://developers.arcgis.com/python/sample-notebooks/identifying-country-names-from-incomplete-house-addresses/>
- Container image support for lambda:
<https://docs.aws.amazon.com/lambda/latest/dg/images-create.html>



Copyright © 2021 Esri. All rights reserved.