Sources: https://docs.python.org/2/howto/webservers.html

https://aws.amazon.com/articles/getting-started-with-aws-and-python/?tag=articles%23keywords%23python

Interfacing with the server

- Common Gateway Interface or “CGI” is the oldest interface that allows python code to interact with the server.
- The server needs to start an interface every time it wants to interact with the python code
- This means that for every request requires a new python instance to be started, thus it is only viable for low levels of use

Mod_python

- Embeds python interpreter into an apache process
- This speeds up response time for each request
- Provides access to apache internals
- Uses caching when executing files so changes require the web server to be restarted
- Every child process of apache ahs its own python interpreter, and apache starts child processes to handle requests

WSGI

- Web server gateway interface
- Regarded as the best way to do python web programming by several sources
- Use a wsgi wrapper, like mod_wsgi
- Awsgi https://pypi.org/project/aws-wsgi/ allows you to use aws api with wsgi wrappers

BOTO

- According to amazon’s tutorials boto is the best way by far to integrate python into aws
- Is actively maintained and has been around for years
- Link at top of sources walks through installation

Django

- A special server type in aws
- A high level python web framework
- Tutorial for creating a django app and hosting it on apache https://aws.amazon.com/getting-started/hands-on/deploy-python-application/
- Django functionality with a rds https://www.youtube.com/watch?v=3HPq12w-dww

Conclusion: I believe if we are to use python for our application creating a django instance and hosting it on an apache server would be the best solution. The drawback would be that we would need two servers instead of just one for an ec2 html javascript service, and that the initial setup process is more complicated than the ec2.