

DevOps Kitchen Workshop

AWS Boto3 - Python to Terraform

#week_twenty - Boto3 SDK [🔗](#)

duration: 1 week

ForgTech company wanna test your ability to deliver their requirements utilizing AWS Python SDK libraries, This will help you build a good reputation. The purpose of this task is to build a CLI tool using Python/Boto3 calls to create Terraform code for AWS resources.

The FrogTech Cloud Team requests you to implement local Python scripts with AWS Boto3 SDK that meet the following requirements:

1. Enable the user to specify the region.
2. List the specified region resources (*i.e.* VPC, EC2, enough as a POC)
3. Create a Terraform file that contains the required details to create the fetchd VPCs.
4. Consider docstrings and typehints.
5. By using docstrings and typehints, create a public documentation that explains your tool by MkDocs.
6. Consider using function-based:
 - a. List functions under `utils` directory.

```
1  utils
2  |__ function.py
3  |__ function2.py
4  main.py
```

- b. A function does one thing only.
 - c. List related functions in the same file.
7. Consider using the main file only to call all the functions and handle their dependencies.
 8. Consider the Python PEP8 code standardization (*i.e.* by installing *vscode* extensions like *pylint*, *Black*, or even *ruff*)
 9. Consider using assertions for debugging purposes only.


As well as build a Readme file explaining what your tool is doing and how to use it.

Bonus [🔗](#)

1. Read about [pyproject.toml](https://packaging.python.org/en/latest/tutorials/packaging-projects/)
2. Read about [Linear Regression in Python](https://www.tensorflow.org/tutorials/linear_regression).

3. Read about [How to Publish an Open-Source Python Package to PyPI](#)

References:

- [Boto3 Docs](#)
- [PEP 257 – Docstring Conventions | _peps.python.org](#)
-  [styleguide](#)
- [typehint.](#)
- [MkDocs](#)
- [Assertion](#)
- [PEP 8 – Style Guide for Python Code](#)