

## Boto3 Week 4 Exercises

### A. EC2 exercise

Using your *create\_instance.py* program from the Week 3 exercise as a starting point, copy your program to a new name (**week4.py**) and do the following :

- Once you have created the instance, use a boto3 *waiter* to wait until the instance is running – see “Tip 1” below on Waiters.
- When the instance is running write out a message to say “Instance Running.”
- Use the *webbrowser* module to launch a web browser to display the Apache home page served from your instance – see “Tip 2” below on launching a browser from Python. Note: you may need to implement a sleep timer (or a loop that checks every few seconds) to wait for the UserData script to install and start the web server.

### B. S3 exercise

- Next your program should create an S3 bucket (dynamically generating a unique bucket name).
- Enable static website hosting on this bucket - see “Tip 3” below on static website configuration
- Print out the bucket name and a message to say “Upload an index.html file to test it works!”
- Manually upload an index.html file to this bucket and visit the bucket endpoint to test you have successfully enabled website hosting.

### Student Submission

Upload a **ZIP** file to Moodle containing the following two files:

1. Your Python program file – called week4.py
2. A one-page PDF containing the URL of your S3 bucket website endpoint and also a screenshot showing your program running in a terminal.

#### Tip 1: Waiters

After creating an instance programmatically, you will usually need to wait until the instance reaches the “running” state before doing subsequent tasks. There are a few ways to do this. For example, you could write a loop to keep checking the instance state every few seconds, or (better) you could use a *waiter* method. For example if you have an EC2 instance object called *instance*, you could do this with:

```
instance.wait_until_running()
```

In conjunction with this, always call the *reload()* method on an instance before trying to access its properties – just a simple *instance.reload()* (where the object is named *instance*). This will ensure the object’s properties are refreshed.

#### Tip 2: Launching a browser from Python

You can easily open a browser tab using the *webbrowser* module. For example, the following code will open up the google webpage using the default OS browser.

```
import webbrowser
webbrowser.open_new_tab('https://www.google.com')
```

### Tip 3: Static website configuration

You can set up static website hosting on an S3 bucket by creating a BucketWebsite resource. See this [example](#):

```
website_configuration = {
    'ErrorDocument': {'Key': 'error.html'},
    'IndexDocument': {'Suffix': 'index.html'},
}

bucket_website = s3.BucketWebsite('my-bucket-name')    # replace with your
bucket_name or a string variable

response = bucket_website.put(WebsiteConfiguration=website_configuration)
```