Lab: Getting Started with Boto

Lab. Getting Started with Boto

Boto is "an integrated interface to current and future infrastructural services offered by Amazon Web Services". It enables you to get connected to AWS using Python and send requests to it. To install Boto in windows, first you will need to download the zipped package of Boto at: <u>https://github.com/boto/boto/downloads/</u> and install it on your computer. A detailed documentation for Boto can be found at: <u>http://boto.readthedocs.org/</u>.

1. Working with S3

a) Creating a connection to S3

In order to get connected to S3, you'll need to set your Access Key ID and Secret Access Key, which you can create or delete in the "Security Credentials" page.

* Developers * Community	Support Account	Access Credentials
		There are three types of access credentials used to authenticate your requests to AWS services: (a) access keys, (b) X.509 certificates, and (c) key pairs. Each access credential type is explained below.
	AWS Management Console	Access Keys X.509 Certificates Key Pairs
Personal Information by View and edit personal contact information and set communication preferences for email subscriptions.	Access and manage Amazon Web Services through a point-and-click, web-based user interface. Get more info.	Use access keys to make secure REST or Query protocol requests to any AWS service API. We create one for you when your account is created — see your access key below.
Security Credentials AWS uses two types of access identifiers to authenticate requests to AWS and to identify the sender of a request.	Sign in to the AWS Management Console:	Your Access Keys Created Access Key ID Secret Access Key Status Access Key ID Character Access Key Access Key ID Character Access Key
le Usage Reports Download customizable usage reports for each escular you are subscribed to	Amazon EC2 : Go Save this as your default console	Create a new Access Key
		For your protection, you should never share your secret access keys with anyone. In addition, industry best practice recommends frequent key rotation.
	close ×	C Learn more about Access Keys

if not boto.config.has_section('Credentials'): boto.config.add_section('Credentials')

```
# Set the keys
boto.config.set('Credentials', 'aws_access_key_id', 'your access key id')
boto.config.set('Credentials', 'aws_secret_access_key', 'your secret access key')
```

```
# Connect to S3
s3=boto.connect_s3()
```

b) Creating a bucket

Bucket is a container used to store key/value pairs in S3. You can choose to create one bucket for all of your data or to create separate buckets for different types of data. The name of a bucket needs to be unique. For example, you can name a bucket by including a prefix of your access key:

mybucket=s3.create_bucket('YOUR_AWS_ACCESS_KEY_myBucket')

Once a bucket is created, you can access it or list all the buckets.

mybucket = s3.get_bucket('YOUR_AWS_ACCESS_KEY_myBucket')

c) Storing data in S3

Before you can store your data, a unique key needs to be created first and used to keep track of your data stored in S3.

```
# Create a new key
key = mybucket.new_key('examples/first_file.csv')
```

```
# Set contents for the key
key.set_contents_from_filename('D:\\aws\\first_file.csv')
```

Set accessibility
key.set_acl('public-read')

d) Downloading and copying data

The data stored in S3 can be downloaded or copied from one bucket to another

```
# Download file
```

```
Key=s3.get_bucket(YOUR_AWS_ACCESS_KEY_myBucket').get_key('examples/first_file.csv') key.get_contents_to_filename('/myfile.csv')
```

Copy file

new_key = key.copy('YOUR_AWS_ACCESS_KEY_myBucket2', 'examples2/second_file.csv')

2. Working with EC2

Creating a connection to EC2

Before getting connected to EC2, you need to make sure your credential information is set properly. Please refer to 1.a) for more information.

a) Launching instances

An instance can now be launched as follows. You can either build your own image with ami-image-id or go with a prebuilt one, which will launch an instance in the specified region with the default parameters.

A key pair can then be created if you do not have one.

```
key_pair = ec2.create_key_pair('ec2-sample-key')
key_pair.save('/Users/xxx/.ssh')
```

b) Stopping Instances and Terminating Instances

You can stop or even terminate specified instances if you would need to.

```
ec2.stop_instances (instance_ids=['instance-id-1','instance-id-2', ...])
ec2.terminate_instances (instance_ids=['instance-id-1','instance-id-2', ...])
```

c) Monitoring Instances

It is often needed to know which instances are currently running or not by accessing a specific reservation and its associated instances.

```
reservations = conn.get_all_reservations()
instances = reservations[0].instances
```

To get more detailed information about an instance, you can run the following lines :

inst = instances[0]
inst.instance_type # returns u'c1.xlarge'for example
inst.placement # returns u'us-west-2'for example

References

•

Getting started with AWS and Python

https://aws.amazon.com/articles/3998 .

Getting started with Amazon S3 <u>http://aws.amazon.com/s3/getting-started/</u>.

Getting started with Amazon EC2 <u>http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide</u> /<u>EC2Win_GetStarted.html</u>.

Creator Name	Jie Tian
Content Title	Getting started with Boto
Content Type	Lab Assignment
Part Of	Module 3: High Performance Geospatial Computing
Learning Objectives	
Background Knowledge	
Resources Needed	
Work Mode	
Relation to Project	
Feedback Needed	

Download Word Document Copy of Lab Assignment