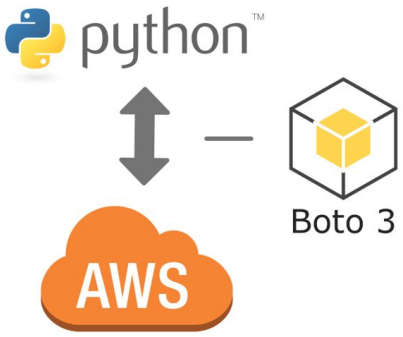
AWS Boto3

CLARUSWAY
WAY TO REINVENT YOURSELF

Boto3

What is Boto3?



Boto3 is the **AWS SDK** (Software Development Kit) for **Python**. It enables you to create, update, and delete AWS resources with your **Python scripts**. It is basically a **Python library**.

CLARUSWAY
WAY TO REINVENT YOURSELF

Boto3

- Boto3 runs on top of botocore, which is foundation for AWS CLI
- A session initiates the connectivity to AWS services. A default session uses the default credential profile(e.g. ~/.aws/credentials, or assume your EC2 using IAM instance profile)

Default session

```
import boto3

# Using the default session
sqs = boto3.client('sqs')
s3 = boto3.resource('s3')
```

Custom session

```
import boto3
import boto3.session

# Create your own session
my_session = boto3.session.Session()
```


Boto3

- Client vs Resource
  - Client:
    - Low-level AWS service access
    - Exposes botocore client to the developer
    - Supports all AWS service operations

Example

```
s3 = boto3.client('s3')
```

Boto3

Example

```
import boto3


    client = boto3.client('s3')
    response = client.list_objects(Bucket='example')
    for content in response['cont']:
        obj_dict = client.get_object(Bucket='example', Key=cont['Key'])
        print(cont['Key'], obj_dict['LastModified'])
```


Boto3

- Resource:
  - Higher-level, object oriented API
  - Exposes sub processes of AWS resources
  - Does not provide 100% coverage of AWS API

Example

```
s3 = boto3.resource('s3')
```

Boto3

Example

```
import boto3


    s3 = boto3.resource('s3')
    bucket = s3.Bucket('example')
    for obj in bucket.objects.all():
        print(obj.key, obj.last_modified)
```


Boto3

Installation and Configuration

Installation

pip install boto3 (pip3 install boto3 for Python3)

Access Configuration

aws configure and supply access keys

Boto3

Using Boto3

```python
import boto3

# Use Amazon S3
s3 = boto3.resource('s3')

# Print out all bucket names
for bucket in s3.buckets.all():
    print(bucket.name)

```


Boto3

Using Boto3

```python
import boto3

# Use Amazon S3
s3 = boto3.resource('s3')

# Upload a new file
data = open('test.jpg', 'rb')
s3.Bucket('my-bucket').put_object(Key='test.jpg', Body=data)

```

Boto3

Boto3 Labs

Lab1:

- Install boto3 on Amazon Linux 2 ec2
- Configure boto3 access
- Test AWS access by listing instances

Lab2:

- Launch ec2 instance
- Resize an instance
- Stop/Start/Terminate an instance

Boto3

Boto3 Labs

Amazon Linux 2 - Install Python and boto3

```
yum install python3
curl -O https://bootstrap.pypa.io/get-pip.py
python3 get-pip.py
pip install boto3
```

CLARUSWAY
WAY TO REINVENT YOURSELF

Boto3

List instances

```
#!/usr/bin/env python

import boto3
ec2 = boto3.resource('ec2')
for instance in ec2.instances.all():
 print (instance.id, instance.state)
```


Boto3

Launch instances

```
#!/usr/bin/env python

import boto3
ec2 = boto3.resource('ec2')

# create a new EC2 instance
instances = ec2.create_instances(
    ImageId='ami-09d95fab7fff3776c',  # <== Update with your  instance ID
    MinCount=1,
    MaxCount=2,
    InstanceType='t2.micro',
    KeyName='key-pair'                # <== Update with your keypair
)
```

Boto3

List instances

```
#!/usr/bin/env python

import boto3
ec2 = boto3.client('ec2')

# choose an EC2 instance with id
instance_id = 'i-03eb9211faac95add'    # <== Update with your  instance ID

# Stop the instance
ec2.stop_instances(InstanceIds=[instance_id])
waiter=ec2.get_waiter('instance_stopped')
waiter.wait(InstanceIds=[instance_id])

# Change the instance type
ec2.modify_instance_attribute(InstanceId=instance_id, Attribute='instanceType', Value='t2.micro')

# Start the instance
ec2.start_instances(InstanceIds=[instance_id])
```


Boto3

Instance codes

0 : pending
16 : running
32 : shutting-down
48 : terminated
64 : stopping
80 : stopped

Boto3

Stop instance

```
#!/usr/bin/env python

import boto3, sys
ec2 = boto3.resource('ec2')

for id in sys.argv[1:]:
    instance =  ec2.Instance(id)
    print(instance.stop())
```


Boto3

Start instance

```
#!/usr/bin/env python

import boto3, sys
ec2 = boto3.resource('ec2')

for id in sys.argv[1:]:
    instance =  ec2.Instance(id)
    print(instance.start())
```

Boto3

Terminate instance

```
#!/usr/bin/env python

import boto3, sys
ec2 = boto3.resource('ec2')

for id in sys.argv[1:]:
    instance =  ec2.Instance(id)
    print(instance.terminate())
```


THANKS!

Any questions?

You can find me at:

- @guile
- guile@clarusway.com