

```

import boto

ec2 = boto.connect_ec2()

instance = ec2.get_all_instances(filters={'paws' : None})[0].instances[0]

# Allocate an Elastic IP Address. This will be associated with your
# account until you explicitly release it.
address = ec2.allocate_address()

# Associate our new Elastic IP Address with our instance.
ec2.associate_address(instance.id, address.public_ip)

# Alternatively, you could do this.
instance.use_ip(address)

# import boto
from boto.s3.connection import Location

def create_bucket(bucket_name, location=Location.DEFAULT):
    """
    bucket_name - The name of the bucket to be created.
    location - The location in which the bucket should be
    created. The location class is a simple
    enum-like static class that has the following attributes:
    # DEFAULT|EU|USWest|APNorthEast|APSouthEast
    s3 = boto.connect_s3()

    # First let's see if we already have a bucket of this name.
    # The lookup method will return a Bucket object if the
    # bucket exists and we have access to it or None.
    bucket = s3.lookup(bucket_name)
    if bucket:
        print 'Bucket (%s) already exists' % bucket_name
    else:
        # Let's try to create the bucket. This will fail if
        # the bucket has already been created by someone else.
        try:
            bucket = s3.create_bucket(bucket_name, location=location)
        except s3.provider.storage_create_error, e:
            print 'Bucket (%s) is owned by another user' % bucket_name
    return bucket

```