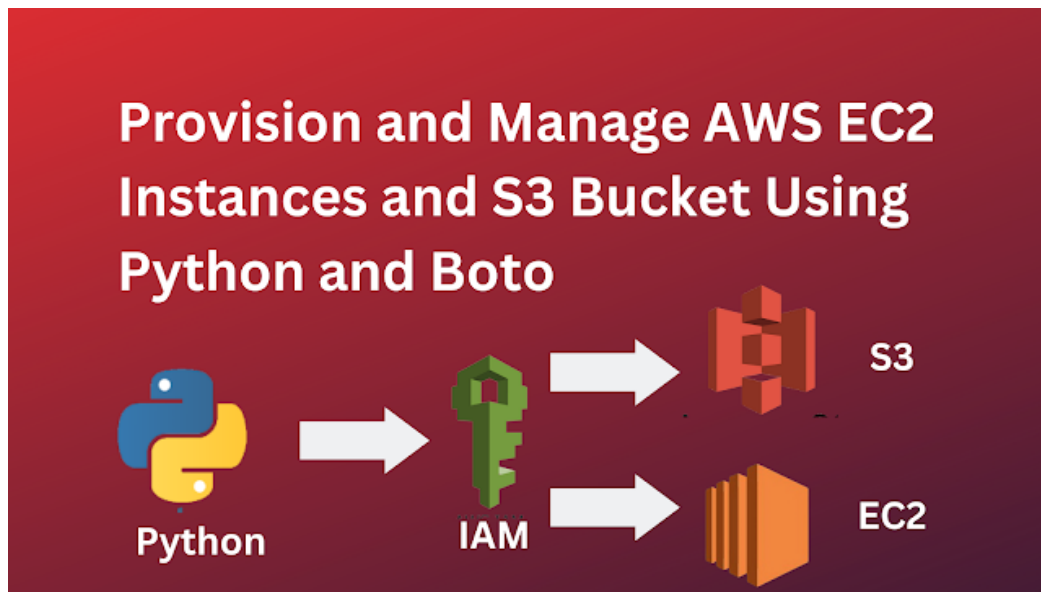


Provision and Manage AWS EC2 Instances and S3 Bucket Using Python

April 17, 2023



Amazon Web Services (AWS) is a cloud computing platform that provides a wide range of services and features to help organizations and individuals run their applications in a secure and scalable manner. Among the most widely used services are Elastic Compute Cloud (EC2), which enables users to launch virtual machines (instances) in the cloud.

In this blog post, we will learn how to provision and manage EC2 instances using Python with Boto, a Python library for working with AWS.

Provisioning EC2 Instances

various ways, including setting environment variables, using the AWS CLI, or specifying them directly in your code. For more information on how to configure AWS credentials, refer to the official AWS documentation.

Provisioning an EC2 instance using Boto requires the following steps:

Install Boto:

To install Boto, run the following command in your terminal:

pip install boto3

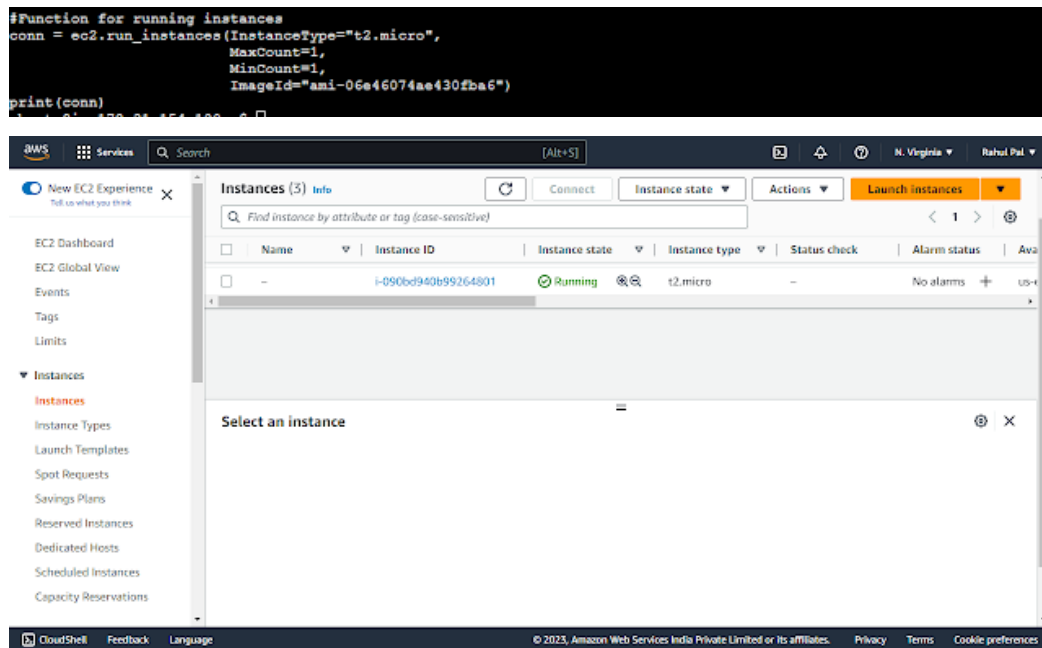
Configure AWS credentials: Before you can use Boto to interact with AWS, you need to configure your AWS credentials. This can be done in various ways, including setting environment variables, using the AWS CLI, or specifying them directly in your code. For more information on how to configure AWS credentials, refer to the official AWS documentation.

Create an EC2 client:

To create an EC2 client using Boto, run the following code:

Launch an instance: To launch an EC2 instance using Boto, you need to specify various parameters, such as the AMI ID, instance type, security group, and key pair. Here's an example code that launches an EC2 instance with Ubuntu 18.04 AMI:

Brand2Cloud - Cloud DevOps & Branding Culture



Manage instances:

Once you have launched an EC2 instance, you can manage it using Boto. For example, you can start, stop, terminate, or reboot an instance using the `start_instances`, `stop_instances`, `terminate_instances`, or `reboot_instances` methods, respectively. Here's an example code that stops an EC2 instance:

```
import boto3
```

```
# Connect to the EC2 service
```

```
ec2 = boto3.client('ec2',
```

```
    'us-east-1',
```

```
    aws_access_key_id='AKIATGIZZJYAWCT63BCL',
```

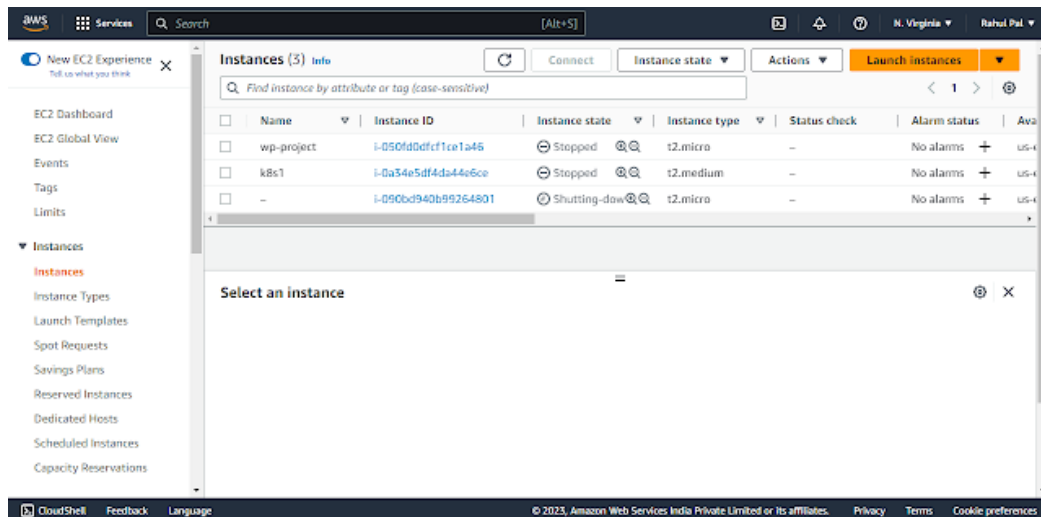
```
    aws_secret_access_key='NLQ8gEsFYhFPgKy5+Pcma1x4lYUCK2UWqtYoeyMP')  
ec2.stop_instances(InstanceIds=['i-090bd940b99264801'])
```

Brand2Cloud - Cloud DevOps & Branding Culture

```
response = ec2.terminate_instances(instance_ids=[i-090bd940b99264801'])
```

Print the response

```
print(response)
```



Managing S3 Buckets

S3 is a scalable object storage service provided by AWS, which allows users to store and retrieve data of any size, in any format. To manage S3 buckets using Boto, follow these steps:

Create an S3 client: To create an S3 client using Boto, run the following code:

```
ubuntu@ip-172-31-154-100:~/python-files$ cat packages.py
import boto3

s3 = boto3.resource('s3')
c = 0
for bucket in s3.buckets.all():
    print(bucket)
    print(bucket.name)
    c+=1
print(c)

s3.Bucket("rahul-bucket-s3").upload_file("linux.py", "linux.py")
```

Create a bucket: To create an S3 bucket using Boto, you need to specify the bucket name and the region in which the bucket should be

In this code, `s3.upload_fileobj` is used to upload a file to an S3 bucket. The file object is opened in binary mode ('rb') and passed as the first argument to `upload_fileobj`. The second argument is the name of the bucket (`bucket_name`), and the third argument is the object key (`object_key`).

```
ubuntu@ip-172-31-154-100:~/python-files$ python3 packages.py
s3.Bucket(name='abhishekshar')
abhishekshar
s3.Bucket(name='rahul-bucket-s3')
rahul-bucket-s3
0
ubuntu@ip-172-31-154-100:~/python-files$
```

Note that the S3 bucket needs to exist before you can upload a file to it. You can create an S3 bucket using the `create_bucket` method, as shown in the previous section.

Conclusion

In this blog post, we learned how to provision and manage EC2 & S3 instances using Python with Boto and also accessed a upload feature. We covered the steps required to launch an instance, as well as how to manage its state. With Boto, you can easily automate the deployment and management of EC2 instances, making it an essential tool for anyone working with AWS.

DEVOPS



Enter comment