XCCELERATED

proudly part of Xebia

ACCELERATING GROWTH

» AWS Cloud Engineering Assessment

Assessment

Introduction

This assignment is part of the Xccelerated assessment procedure for hiring new cloud engineers.

We'd like to give you a chance to demonstrate your skills, knowledge, and ability to learn.
It's hard to evaluate every person right, but we believe that this assignment can be a good start for a conversation.

The following aspects are considered important for the evaluation of your work and solution to this assignment:

- If you have any questions, do ask these. If there are areas of the assignment that are unclear, discuss these.
- Apply and demonstrate your current skills. We want to understand what you already can deliver and we are not seeking for latest greatest technology, you will learn that in our program later.
- The assignment is a minimum of specifications to help get this part of the assessment started. Feel free to discuss any enhancements to the specification.
- Make sure you include essential parts of your implementation to help demonstrate what you think of a high-quality solution to a problem.

If you have any questions or want clarifications on the requirements/assignment you can reach out to Matthijs Brouns, at mbrouns@xccelerated.io

The Assignment

Context

A team you work with asked for your help with automating backups of their virtual machines running on Amazon Web Services. They want to use AWS’s snapshot functionality to make regular snapshots of the Elastic Block Storage disks attached to their virtual machines, based on the labels of those virtual machines.

You are asked to automate this task using Typescript with the AWS SDK for JavaScript (https://docs.aws.amazon.com/AWSJavaScriptSDK/v3/latest/clients/client-ec2/index.html) or with Python and the boto3 library (https://boto3.amazonaws.com/v1/documentation/api/latest/index.html). You should get an email with an invitation to our AWS environment. In order to authenticate and work with AWS you have to install the AWS CLI (https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html). Afterwards you should be able to run `aws configure` and by following the directions authenticate with the account created for you.

The project has been seeded with two virtual machines. A single snapshot already exists for one of the machines. Typically, such a project would need to be deployed to the cloud environment to run periodically, but this is left **out of scope** for this assessment due to time limitations. The application only needs to run interactively on your local machine. There is some sample code available at the end of this document to help you get started.

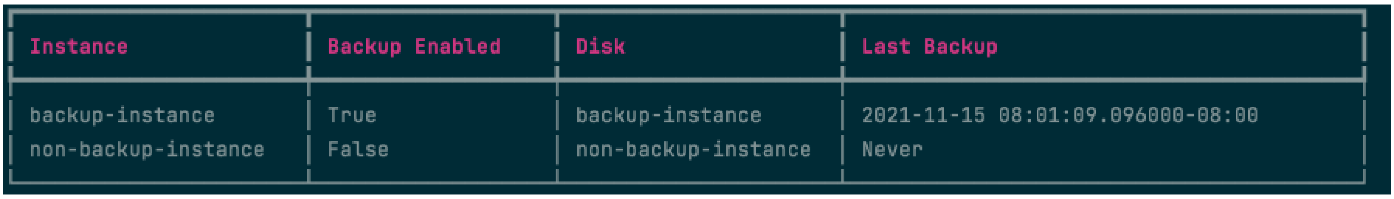
The team provided you with the following user stories:

BACKUP-1

As an engineer, I want a program to list the virtual machines in our account, the VolumeId of their first disk, whether backups are enabled and if so, when their latest backup happened. Virtual Machines on AWS are called EC2 Instances. The EC2 instances we provisioned for you are in the ‘us-east-1’ region

Backups are considered to be enabled if the virtual machine has a tag ‘backup’ with the value ‘true’.

Example output of your program (you can ignore the formatting):



```
| Instance            | Backup Enabled | Disk                | Last Backup                      |
|---------------------|----------------|---------------------|----------------------------------|
| backup-instance     | True           | backup-instance     | 2021-11-15 08:01:09.096000-08:00 |
| non-backup-instance | False          | non-backup-instance | Never                            |
```

BACKUP-2

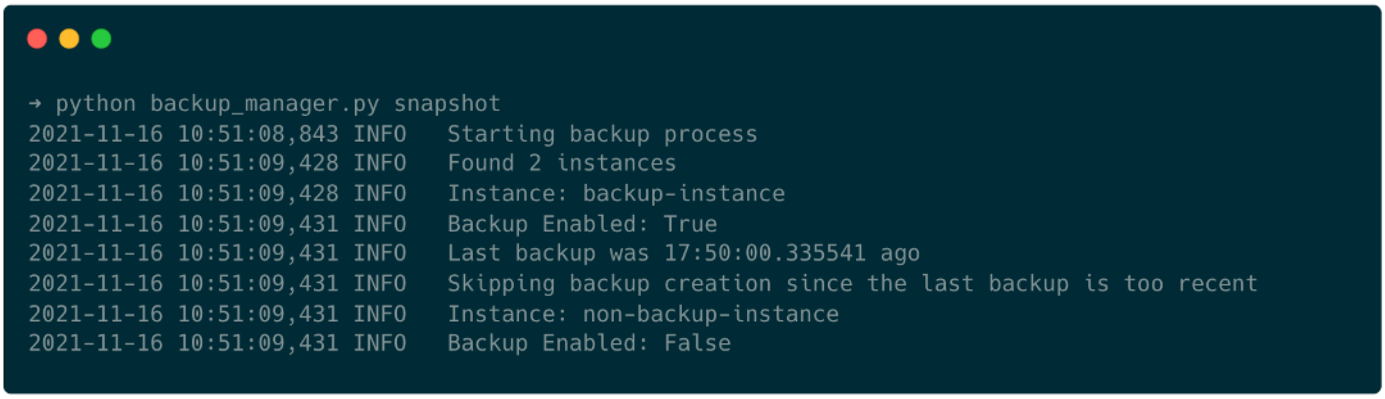
As an engineer, I want to be able to run a program that creates a snapshot for all primary disks of virtual machines with label ‘backup’ set to ‘true.

The intended backup schedule is to create daily backups, meaning that the program should only create new snapshots if the previous snapshot wasn’t made today.

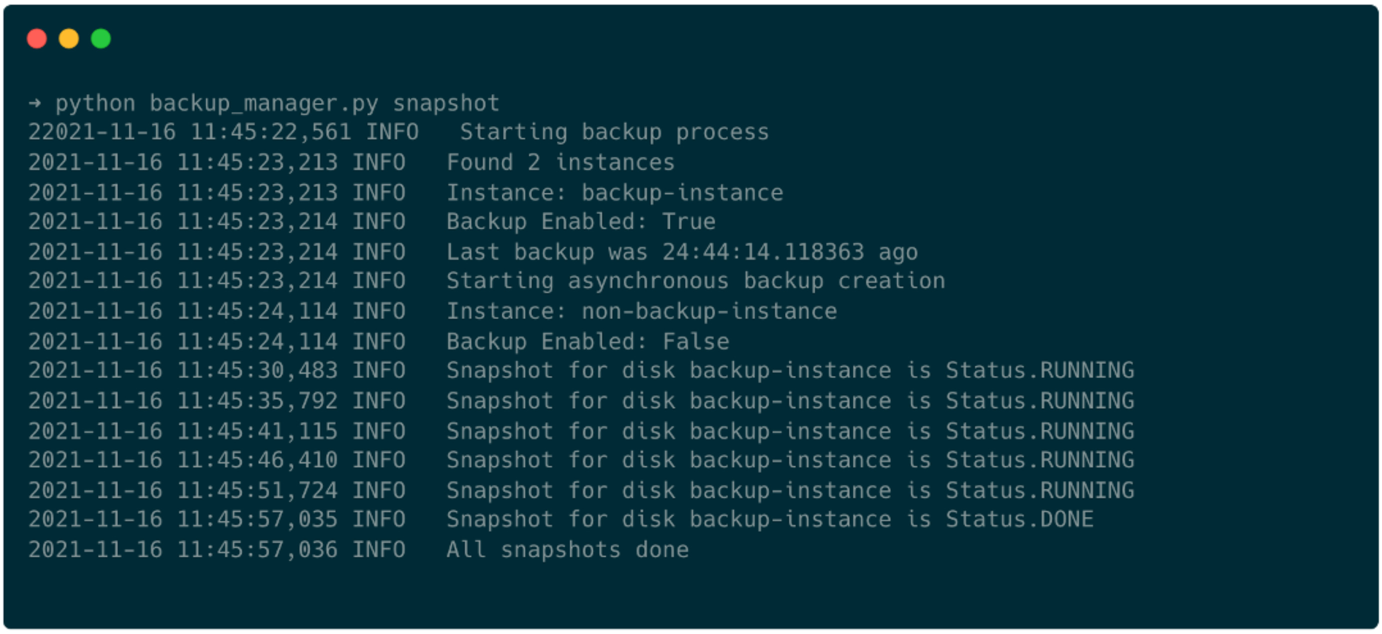
Hints:

- Creating snapshots of disks in AWS happens asynchronously.
- Think about if, how and where to wait for the snapshot operations to complete

Example output:



```
➜ python backup_manager.py snapshot
2021-11-16 10:51:08,843 INFO   Starting backup process
2021-11-16 10:51:09,428 INFO   Found 2 instances
2021-11-16 10:51:09,428 INFO   Instance: backup-instance
2021-11-16 10:51:09,431 INFO   Backup Enabled: True
2021-11-16 10:51:09,431 INFO   Last backup was 17:50:00.335541 ago
2021-11-16 10:51:09,431 INFO   Skipping backup creation since the last backup is too recent
2021-11-16 10:51:09,431 INFO   Instance: non-backup-instance
2021-11-16 10:51:09,431 INFO   Backup Enabled: False
```



```
➜ python backup_manager.py snapshot
22021-11-16 11:45:22,561 INFO   Starting backup process
2021-11-16 11:45:23,213 INFO   Found 2 instances
2021-11-16 11:45:23,213 INFO   Instance: backup-instance
2021-11-16 11:45:23,214 INFO   Backup Enabled: True
2021-11-16 11:45:23,214 INFO   Last backup was 24:44:14.118363 ago
2021-11-16 11:45:23,214 INFO   Starting asynchronous backup creation
2021-11-16 11:45:24,114 INFO   Instance: non-backup-instance
2021-11-16 11:45:24,114 INFO   Backup Enabled: False
2021-11-16 11:45:30,483 INFO   Snapshot for disk backup-instance is Status.RUNNING
2021-11-16 11:45:35,792 INFO   Snapshot for disk backup-instance is Status.RUNNING
2021-11-16 11:45:41,115 INFO   Snapshot for disk backup-instance is Status.RUNNING
2021-11-16 11:45:46,410 INFO   Snapshot for disk backup-instance is Status.RUNNING
2021-11-16 11:45:51,724 INFO   Snapshot for disk backup-instance is Status.RUNNING
2021-11-16 11:45:57,035 INFO   Snapshot for disk backup-instance is Status.DONE
2021-11-16 11:45:57,036 INFO   All snapshots done
```

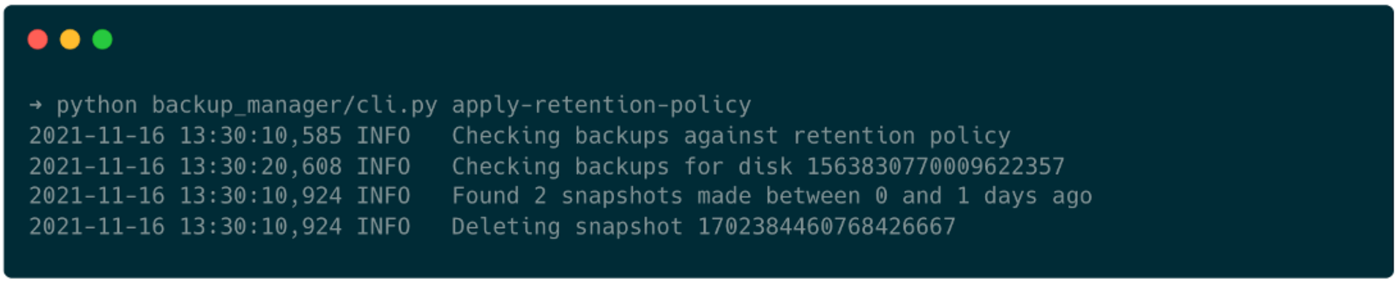
BACKUP-3

As an engineer, we want old backups to be removed according to the following retention policy:

No more than one backup per day should be kept for backups made in the last 7 days
No more than one backup per week should be kept for backups made prior to the last 7 days

When removing backups that don’t match the retention policy, keep the most recent backup that still fits.

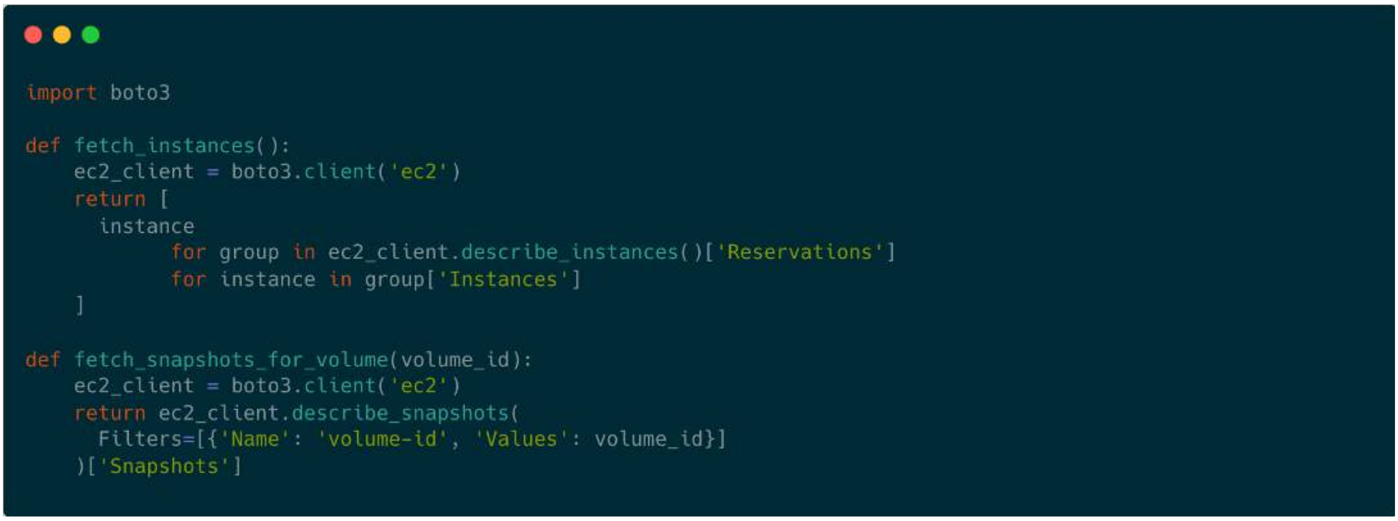
Example output



```
→ python backup_manager/cli.py apply-retention-policy
2021-11-16 13:30:10,585 INFO   Checking backups against retention policy
2021-11-16 13:30:20,608 INFO   Checking backups for disk 1563830770009622357
2021-11-16 13:30:10,924 INFO   Found 2 snapshots made between 0 and 1 days ago
2021-11-16 13:30:10,924 INFO   Deleting snapshot 1702384460768426667
```

Hints

In order to help you along, we’ve provided two functions. These functions create a client object that can be used to manipulate AWS objects. You can use these as inspiration, or as a way to get started with the assessment.



```python
import boto3

def fetch_instances():
    ec2_client = boto3.client('ec2')
    return [
      instance
            for group in ec2_client.describe_instances()['Reservations']
            for instance in group['Instances']
    ]

def fetch_snapshots_for_volume(volume_id):
    ec2_client = boto3.client('ec2')
    return ec2_client.describe_snapshots(
      Filters=[{'Name': 'volume-id', 'Values': volume_id}]
    )['Snapshots']
```

Timeframe

We expect that you should spend around 8 hours on this task. Don’t forget to spend some time on testing and documentation. If you can make it in 10 hours of work that is a good sign. If you need a little bit more that’s ok too. However, do not spend more than 14 hours. It’s better to deliver something simple than nothing at all.

Also, let us know if you feel that the time estimate is inaccurate. This feedback is valuable for us.

Matthijs Brouns
CTO
✉ mbrouns@xccelerated.io

XCCELERATED

proudly part of Xebia

Xccelerated B.V.
Sarphatistraat 640 - 642
1018 AV Amsterdam

www.xccelerated.io
hello@xccelerated.io
+31 (0) 20 76 09 842