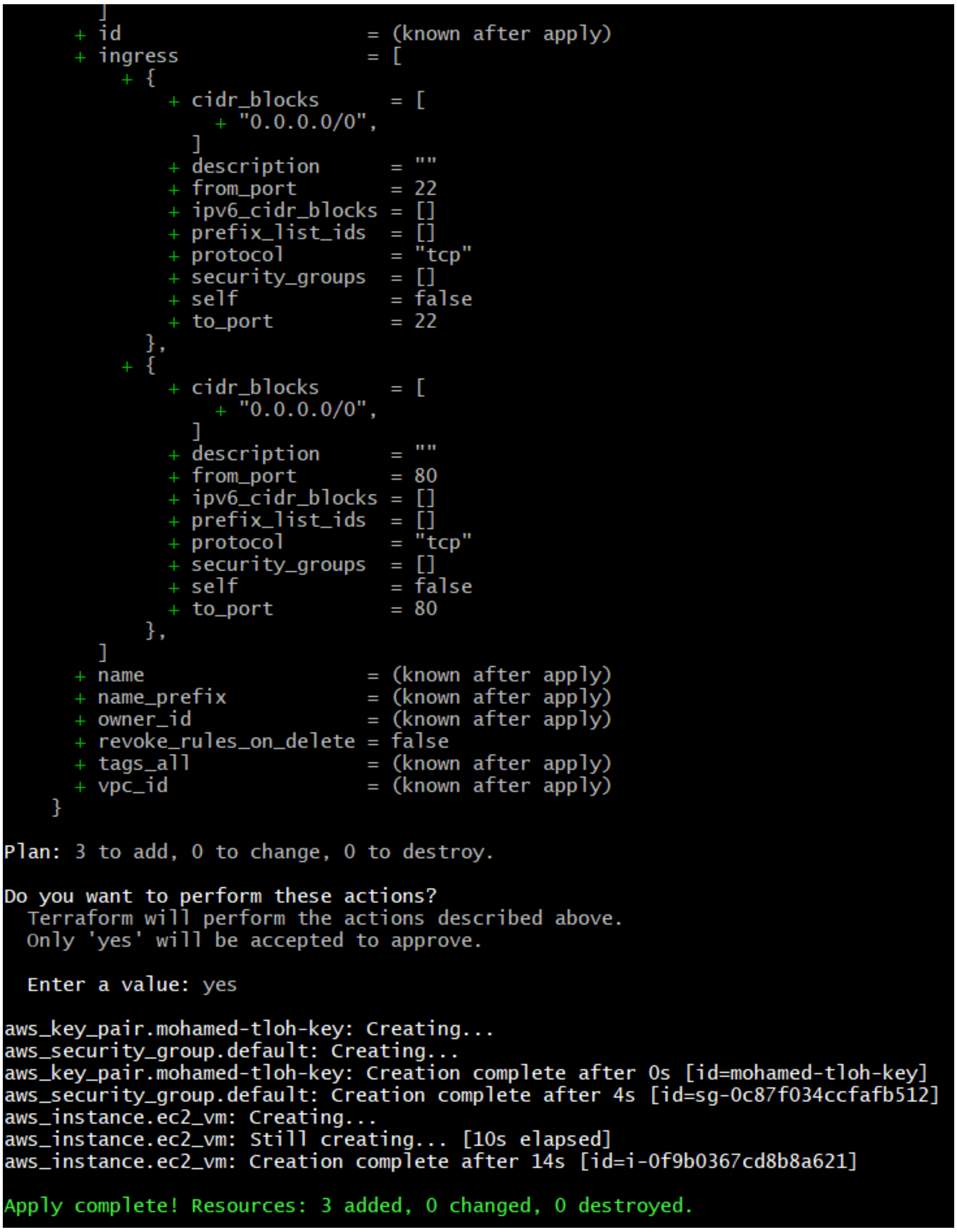
Tloh

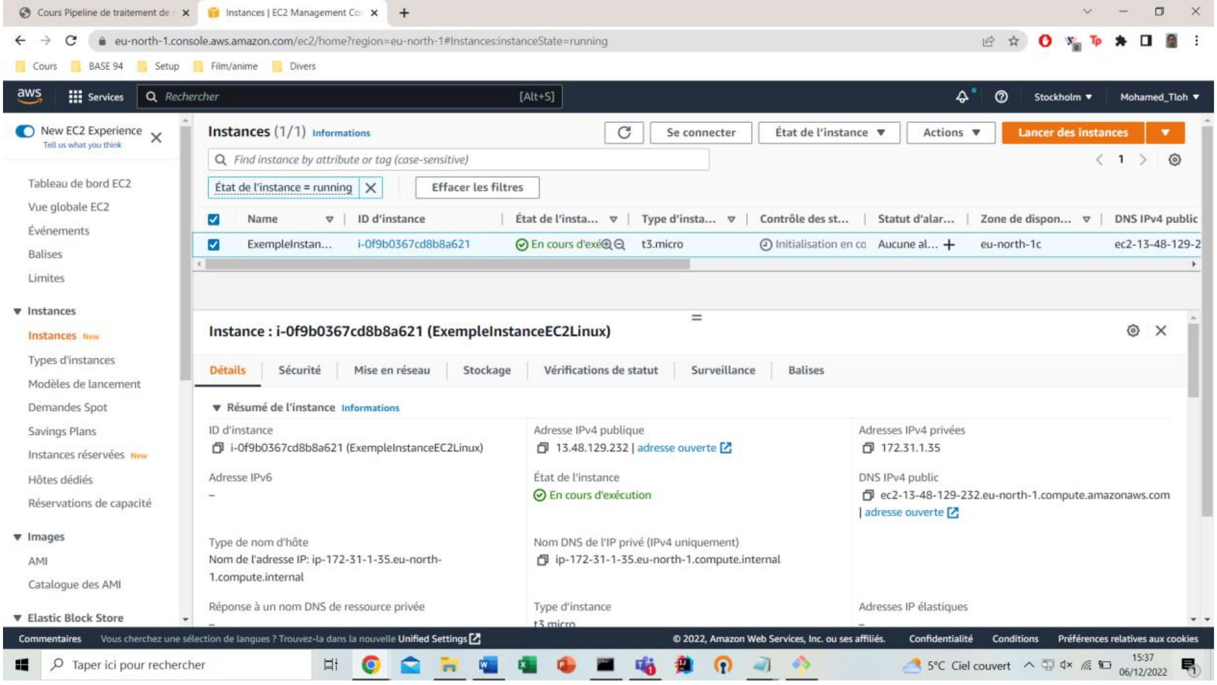
Mohamed

ING3 FISE

Question 1



```
        ]
      + id                     = (known after apply)
      + ingress                = [
          + {
              + cidr_blocks      = [
                  + "0.0.0.0/0",
                ]
              + description      = ""
              + from_port        = 22
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "tcp"
              + security_groups  = []
              + self             = false
              + to_port          = 22
            },
          + {
              + cidr_blocks      = [
                  + "0.0.0.0/0",
                ]
              + description      = ""
              + from_port        = 80
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "tcp"
              + security_groups  = []
              + self             = false
              + to_port          = 80
            },
        ]
      + name                   = (known after apply)
      + name_prefix            = (known after apply)
      + owner_id               = (known after apply)
      + revoke_rules_on_delete = false
      + tags_all               = (known after apply)
      + vpc_id                 = (known after apply)
    }

Plan: 3 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_key_pair.mohamed-tloh-key: Creating...
aws_security_group.default: Creating...
aws_key_pair.mohamed-tloh-key: Creation complete after 0s [id=mohamed-tloh-key]
aws_security_group.default: Creation complete after 4s [id=sg-0c87f034ccfafb512]
aws_instance.ec2_vm: Creating...
aws_instance.ec2_vm: Still creating... [10s elapsed]
aws_instance.ec2_vm: Creation complete after 14s [id=i-0f9b0367cd8b8a621]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
```



Instances (1/1) Informations

État de l'instance = running

☑	Name	ID d'instance	État de l'insta...	Type d'insta...	Contrôle des st...	Statut d'alar...	Zone de dispon...	DNS IPv4 public
☑	ExempleInstan...	i-0f9b0367cd8b8a621	En cours d'exé	t3.micro	Initialisation en co	Aucune al... +	eu-north-1c	ec2-13-48-129-2

Instance : i-0f9b0367cd8b8a621 (ExempleInstanceEC2Linux)

▼ Résumé de l'instance Informations

ID d'instance
i-0f9b0367cd8b8a621 (ExempleInstanceEC2Linux)

Adresse IPv4 publique
13.48.129.232 | adresse ouverte

Adresses IPv4 privées
172.31.1.35

Adresse IPv6
–

État de l'instance
En cours d'exécution

DNS IPv4 public
ec2-13-48-129-232.eu-north-1.compute.amazonaws.com | adresse ouverte

Type de nom d'hôte
Nom de l'adresse IP: ip-172-31-1-35.eu-north-1.compute.internal

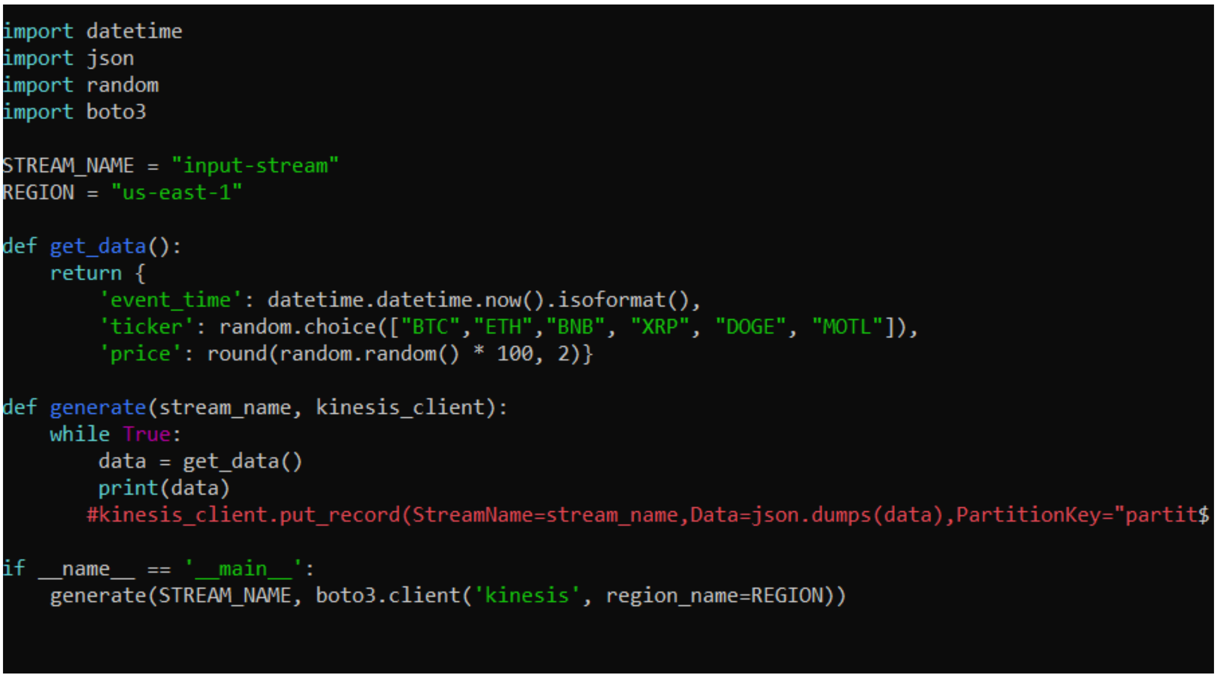
Nom DNS de l'IP privé (IPv4 uniquement)
ip-172-31-1-35.eu-north-1.compute.internal

Réponse à un nom DNS de ressource privée
–

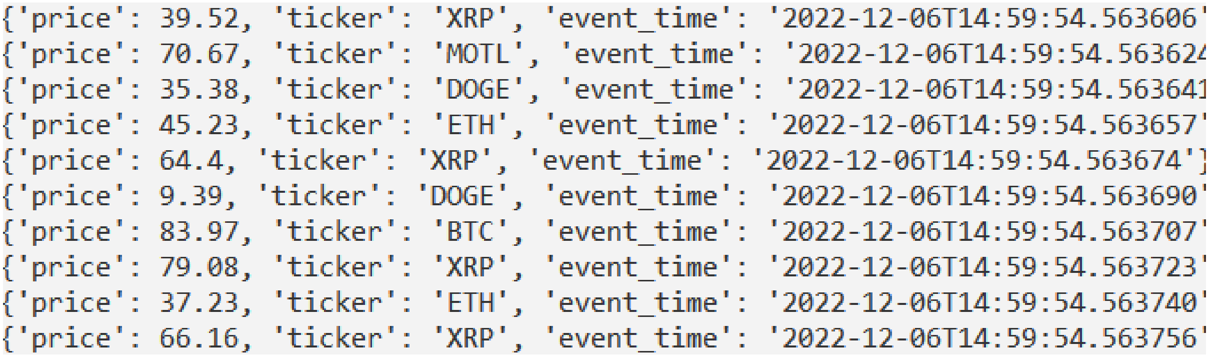
Type d'instance
t3.micro

Adresses IP élastiques
–

Question 2

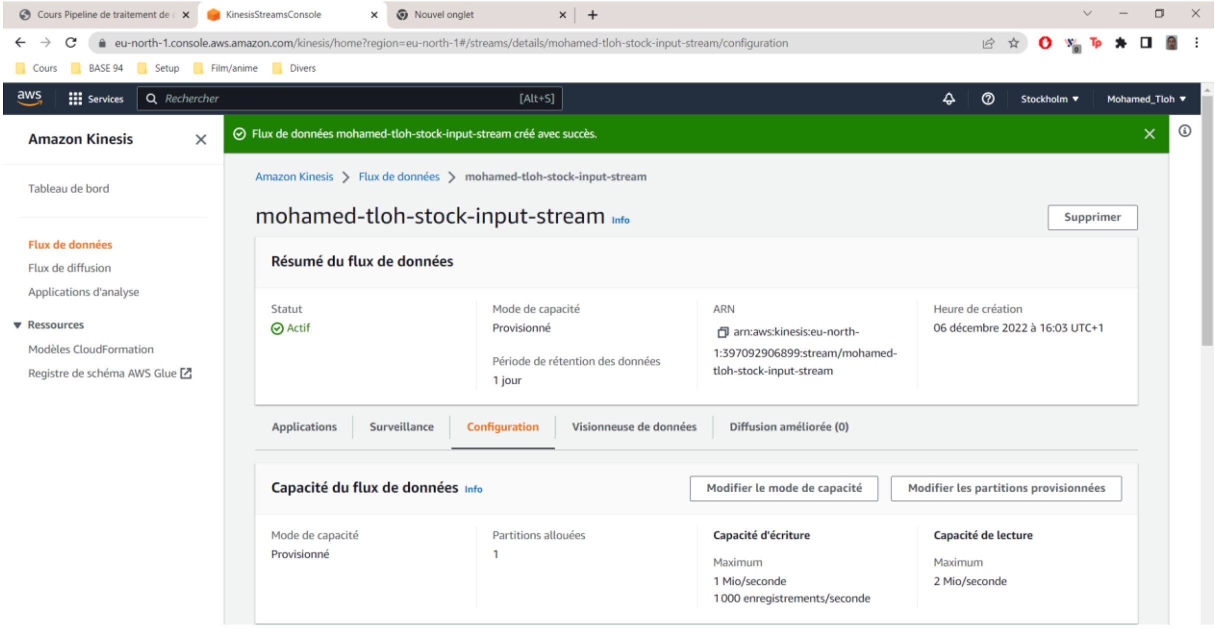


```
import datetime
import json
import random
import boto3

STREAM_NAME = "input-stream"
REGION = "us-east-1"

def get_data():
    return {
        'event_time': datetime.datetime.now().isoformat(),
        'ticker': random.choice(["BTC","ETH","BNB", "XRP", "DOGE", "MOTL"]),
        'price': round(random.random() * 100, 2)}

def generate(stream_name, kinesis_client):
    while True:
        data = get_data()
        print(data)
       #kinesis_client.put_record(StreamName=stream_name,Data=json.dumps(data),PartitionKey="partit$

if __name__ == '__main__':
    generate(STREAM_NAME, boto3.client('kinesis', region_name=REGION))
```



```
{'price': 39.52, 'ticker': 'XRP', 'event_time': '2022-12-06T14:59:54.563606'
{'price': 70.67, 'ticker': 'MOTL', 'event_time': '2022-12-06T14:59:54.563624
{'price': 35.38, 'ticker': 'DOGE', 'event_time': '2022-12-06T14:59:54.563641
{'price': 45.23, 'ticker': 'ETH', 'event_time': '2022-12-06T14:59:54.563657'
{'price': 64.4, 'ticker': 'XRP', 'event_time': '2022-12-06T14:59:54.563674'}
{'price': 9.39, 'ticker': 'DOGE', 'event_time': '2022-12-06T14:59:54.563690'
{'price': 83.97, 'ticker': 'BTC', 'event_time': '2022-12-06T14:59:54.563707'
{'price': 79.08, 'ticker': 'XRP', 'event_time': '2022-12-06T14:59:54.563723'
{'price': 37.23, 'ticker': 'ETH', 'event_time': '2022-12-06T14:59:54.563740'
{'price': 66.16, 'ticker': 'XRP', 'event_time': '2022-12-06T14:59:54.563756'
```

Question 3



eu-north-1.console.aws.amazon.com/kinesis/home?region=eu-north-1#/streams/details/mohamed-tloh-stock-input-stream/configuration

Flux de données mohamed-tloh-stock-input-stream créé avec succès.

Amazon Kinesis > Flux de données > mohamed-tloh-stock-input-stream

mohamed-tloh-stock-input-stream

Résumé du flux de données

Statut	Mode de capacité	ARN	Heure de création
Actif	Provisionné Période de rétention des données 1 jour	arn:aws:kinesis:eu-north-1:397092906899:stream/mohamed-tloh-stock-input-stream	06 décembre 2022 à 16:03 UTC+1

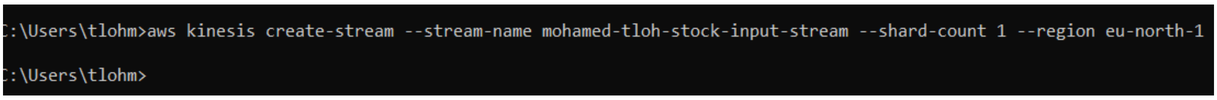
Capacité du flux de données

Mode de capacité	Partitions allouées	Capacité d'écriture	Capacité de lecture
Provisionné	1	Maximum 1 Mio/seconde 1 000 enregistrements/seconde	Maximum 2 Mio/seconde

Question 4.1

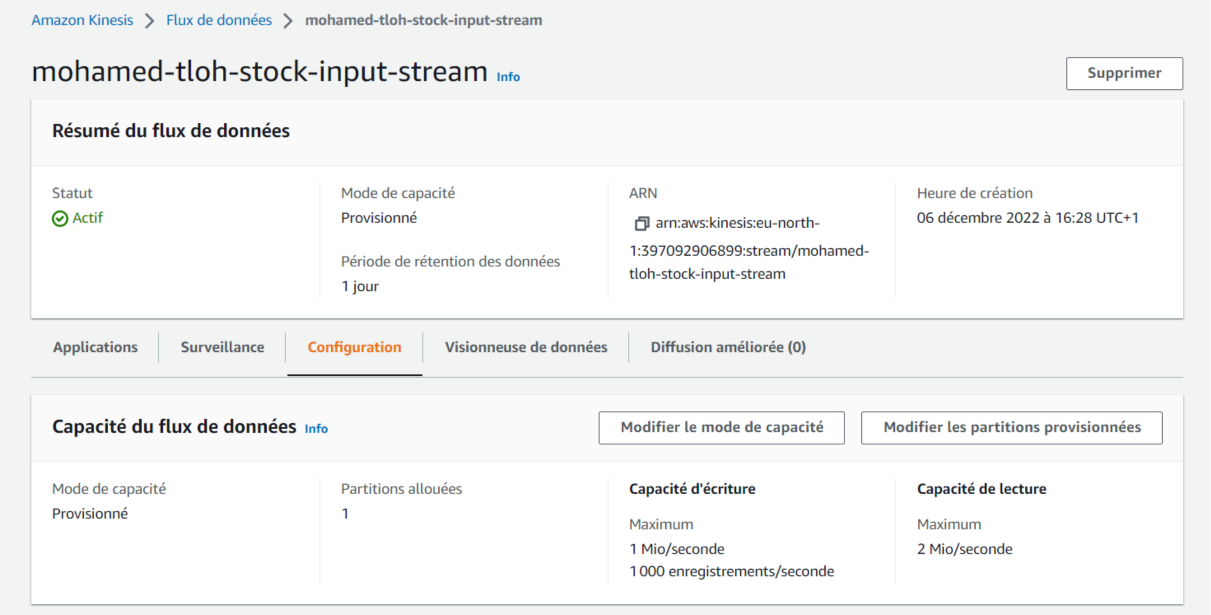
aws kinesis delete-stream --stream-name "mohamed-tloh-stock-input-stream"

Question 4.2



```
:\Users\tlohm>aws kinesis create-stream --stream-name mohamed-tloh-stock-input-stream --shard-count 1 --region eu-north-1

:\Users\tlohm>
```

aws kinesis create-stream --stream-name mohamed-tloh-stock-input-stream --shard-count 1 --region eu-north-1



Amazon Kinesis > Flux de données > mohamed-tloh-stock-input-stream

mohamed-tloh-stock-input-stream

Résumé du flux de données

Statut
Actif

Mode de capacité
Provisionné

Période de rétention des données
1 jour

ARN
arn:aws:kinesis:eu-north-1:397092906899:stream/mohamed-tloh-stock-input-stream

Heure de création
06 décembre 2022 à 16:28 UTC+1

Applications
Surveillance
Configuration
Visionneuse de données
Diffusion améliorée (0)

Capacité du flux de données

Modifier le mode de capacité
Modifier les partitions provisionnées

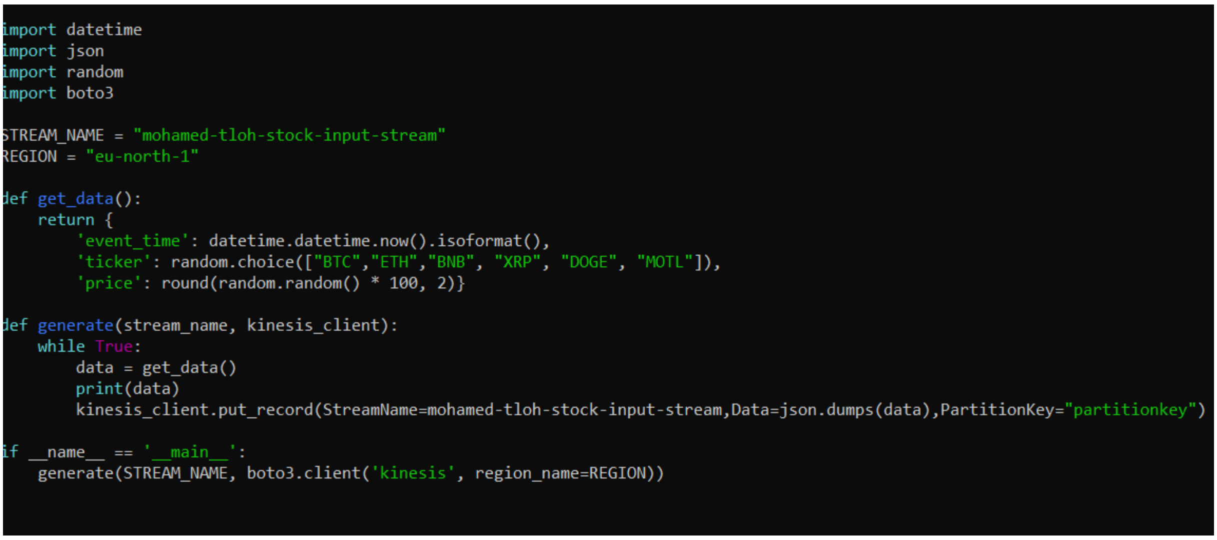
Mode de capacité
Provisionné

Partitions allouées
1

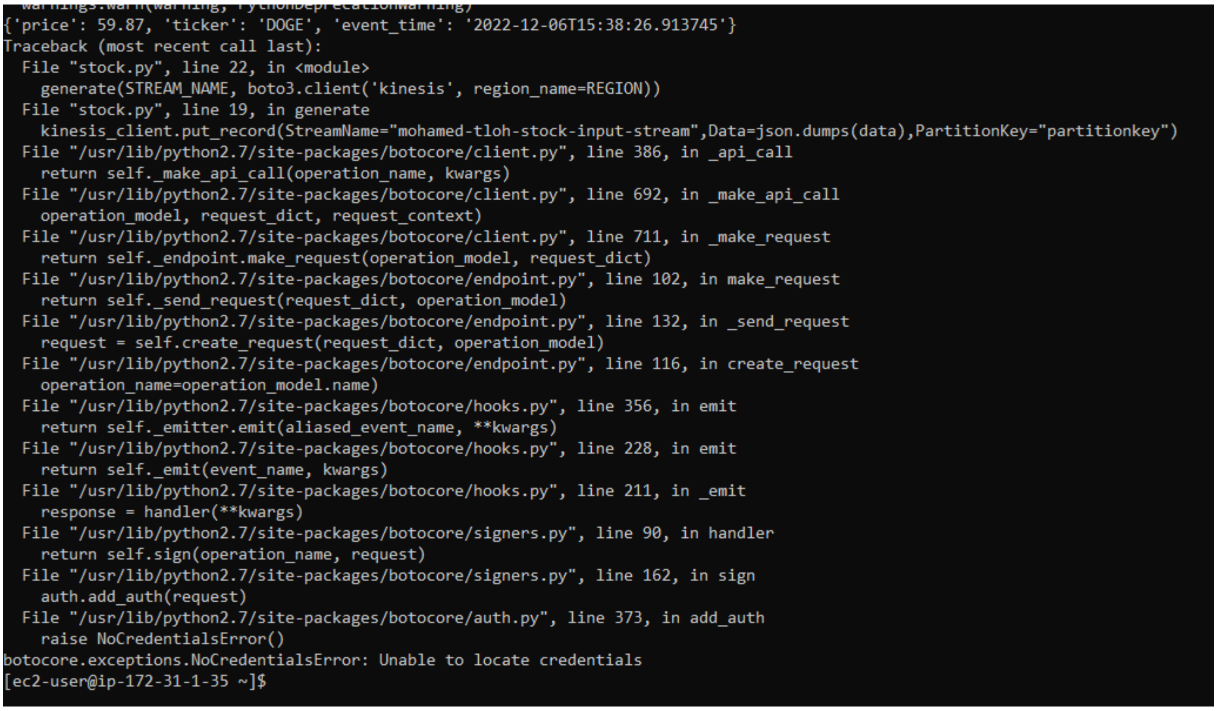
Capacité d'écriture
Maximum
1 Mio/seconde
1 000 enregistrements/seconde

Capacité de lecture
Maximum
2 Mio/seconde

5.1



```
import datetime
import json
import random
import boto3

STREAM_NAME = "mohamed-tloh-stock-input-stream"
REGION = "eu-north-1"

def get_data():
    return {
        'event_time': datetime.datetime.now().isoformat(),
        'ticker': random.choice(["BTC","ETH","BNB", "XRP", "DOGE", "MOTL"]),
        'price': round(random.random() * 100, 2)}

def generate(stream_name, kinesis_client):
    while True:
        data = get_data()
        print(data)
        kinesis_client.put_record(StreamName=mohamed-tloh-stock-input-stream,Data=json.dumps(data),PartitionKey="partitionkey")

if __name__ == '__main__':
    generate(STREAM_NAME, boto3.client('kinesis', region_name=REGION))
```



```
  warnings.warn(warning, PythonDeprecationWarning)
{'price': 59.87, 'ticker': 'DOGE', 'event_time': '2022-12-06T15:38:26.913745'}
Traceback (most recent call last):
  File "stock.py", line 22, in <module>
    generate(STREAM_NAME, boto3.client('kinesis', region_name=REGION))
  File "stock.py", line 19, in generate
    kinesis_client.put_record(StreamName="mohamed-tloh-stock-input-stream",Data=json.dumps(data),PartitionKey="partitionkey")
  File "/usr/lib/python2.7/site-packages/botocore/client.py", line 386, in _api_call
    return self._make_api_call(operation_name, kwargs)
  File "/usr/lib/python2.7/site-packages/botocore/client.py", line 692, in _make_api_call
    operation_model, request_dict, request_context)
  File "/usr/lib/python2.7/site-packages/botocore/client.py", line 711, in _make_request
    return self._endpoint.make_request(operation_model, request_dict)
  File "/usr/lib/python2.7/site-packages/botocore/endpoint.py", line 102, in make_request
    return self._send_request(request_dict, operation_model)
  File "/usr/lib/python2.7/site-packages/botocore/endpoint.py", line 132, in _send_request
    request = self.create_request(request_dict, operation_model)
  File "/usr/lib/python2.7/site-packages/botocore/endpoint.py", line 116, in create_request
    operation_name=operation_model.name)
  File "/usr/lib/python2.7/site-packages/botocore/hooks.py", line 356, in emit
    return self._emitter.emit(aliased_event_name, **kwargs)
  File "/usr/lib/python2.7/site-packages/botocore/hooks.py", line 228, in emit
    return self._emit(event_name, kwargs)
  File "/usr/lib/python2.7/site-packages/botocore/hooks.py", line 211, in _emit
    response = handler(**kwargs)
  File "/usr/lib/python2.7/site-packages/botocore/signers.py", line 90, in handler
    return self.sign(operation_name, request)
  File "/usr/lib/python2.7/site-packages/botocore/signers.py", line 162, in sign
    auth.add_auth(request)
  File "/usr/lib/python2.7/site-packages/botocore/auth.py", line 373, in add_auth
    raise NoCredentialsError()
botocore.exceptions.NoCredentialsError: Unable to locate credentials
[ec2-user@ip-172-31-1-35 ~]$
```

5.2 Par défauts, les instances refusent les flux intrants, il faudrait configurer le data stream pour que l’instance accepte de recevoir des données.

6.