



# Examen Selim Bouhassatine

🕒 Created time	@6 décembre 2022 15:16
↗ Course	
📁 Type	
📎 Files	
↗ Tasks	
↗ Project	
↗ Related to Student Flashcards (Property)	
<input checked="" type="checkbox"/> Archive	<input type="checkbox"/>
☰ Ressource	

Question 1 :

Screenshot du terraform plan :

```

    ]
    + 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.

Note: You didn't use the `-out` option to save this plan, so Terraform can't guarantee to take exactly these actions if you run `"terraform apply"` now.

→ **terraform** █

Screenshoot de l'instance lancé depuis la console aws

EC2 > Instances > i-054985614664d670c

**Résumé de l'instance pour i-054985614664d670c (selim.bouhassatine@etu.u-pec.fr)** Informations

Mis à jour il y a less than a minute

Se connecter État de l'instance Actions

<p>ID d'instance i-054985614664d670c (selim.bouhassatine@etu.u-pec.fr)</p> <p>Adresse IPv6 -</p> <p>Type de nom d'hôte Nom de l'adresse IP: ip-172-31-15-186.eu-north-1.compute.internal</p> <p>Réponse à un nom DNS de ressource privée -</p> <p>Adresse IP attribuée automatiquement 13.50.107.134 [IP publique]</p> <p>Rôle IAM -</p>	<p>Adresse IPv4 publique 13.50.107.134   <a href="#">adresse ouverte</a></p> <p>État de l'instance En cours d'exécution</p> <p>Nom DNS de l'IP privé (IPv4 uniquement) ip-172-31-15-186.eu-north-1.compute.internal</p> <p>Type d'instance t3.micro</p> <p>ID de VPC vpc-0286dde36b8046607</p> <p>ID de sous-réseau subnet-09cf5f9ea19f046b5</p>	<p>Adresses IPv4 privées 172.31.15.186</p> <p>DNS IPv4 public ec2-13-50-107-134.eu-north-1.compute.amazonaws.com   <a href="#">adresse ouverte</a></p> <p>Adresses IP élastiques -</p> <p>Recherche d'AWS Compute Optimizer <a href="#">Inscrivez-vous à AWS Compute Optimizer pour obtenir des recommandations.</a> <a href="#">En savoir plus</a></p> <p>Nom du groupe Auto Scaling -</p>
--	--	---

Détails Sécurité Mise en réseau Stockage Vérifications de statut Surveillance Balises

▼ Détails de l'instance Informations

<p>Plateforme Red Hat (dédit)</p> <p>Informations sur la plateforme Red Hat Enterprise Linux</p> <p>Protection contre l'arrêt Désactivé</p> <p>Récupération automatique de l'instance Par défaut</p>	<p>ID AMI ami-0286da901b5df8024</p> <p>Nom de l'AMI RHEL-9.1.0_HVM-20221101-x86_64-2-Hourly2-GP2</p> <p>Heure de lancement Tue Dec 06 2022 15:30:37 GMT+0100 (heure normale d'Europe centrale) (1 minute)</p> <p>Cycle de vie normal</p>	<p>Surveillance désactivé</p> <p>Protection de la résilience Désactivé</p> <p>Emplacement de l'AMI amazon/RHEL-9.1.0_HVM-20221101-x86_64-2-Hourly2-GP2</p> <p>Comportement Arrêt - Mise en veille prolongée désactivé</p>
--	--	---

Question 2 :

Connexion à la VM depuis ma machine avec un terminal

```
ssh -i "mykey.pem" ec2-user@ec2-13-50-107-134.eu-north-1.compute.amazonaws.com
```

```

Verifying      : python3-libs-3.9.14-1.el9_1.1.x86_64          5/6
Verifying      : python3-libs-3.9.14-1.el9.x86_64             6/6
Installed products updated.

Upgraded:
  python-unversioned-command-3.9.14-1.el9_1.1.noarch          python3-3.9.14-1.el9_1.1.x86_64          python3-libs-3.9.14-1.el9_1.1.x86_64

Complete!
[ec2-user@ip-172-31-15-186 ~]$ sudo yum install -y pip
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Last metadata expiration check: 0:00:50 ago on Tue 06 Dec 2022 02:38:36 PM UTC.
Dependencies resolved.
=====
Package                Architecture      Version           Repository          Size
=====
Installing:
python3-pip            noarch            21.2.3-6.el9      rhel-9-appstream-rhui-rpms 2.0 M
=====
Transaction Summary
=====
Install 1 Package

Total download size: 2.0 M
Installed size: 8.7 M
Downloading Packages:
python3-pip-21.2.3-6.el9.noarch.rpm                                25 MB/s | 2.0 MB  00:00
-----
Total                                                                19 MB/s | 2.0 MB  00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                                             1/1
  Installing     : python3-pip-21.2.3-6.el9.noarch            1/1
  Running scriptlet: python3-pip-21.2.3-6.el9.noarch            1/1
  Verifying      : python3-pip-21.2.3-6.el9.noarch            1/1
Installed products updated.

Installed:
  python3-pip-21.2.3-6.el9.noarch

Complete!
[ec2-user@ip-172-31-15-186 ~]$

```

Installation réussi

```

[ec2-user@ip-172-31-15-186 ~]$ python --version
Python 3.9.14
[ec2-user@ip-172-31-15-186 ~]$ pip --version
pip 21.2.3 from /usr/lib/python3.9/site-packages/pip (python 3.9)
[ec2-user@ip-172-31-15-186 ~]$

```

Voilà un screen de la sortie du programme python, on remarque bien l'ajout de ticker 'SEBO'

```
{'event_time': '2022-12-06T14:45:00.174642', 'ticker': 'SEBO', 'price': 5.28}
{'event_time': '2022-12-06T14:45:00.174671', 'ticker': 'BNB', 'price': 38.66}
{'event_time': '2022-12-06T14:45:00.174690', 'ticker': 'DOGE', 'price': 19.97}
{'event_time': '2022-12-06T14:45:00.174708', 'ticker': 'ETH', 'price': 86.91}
{'event_time': '2022-12-06T14:45:00.174723', 'ticker': 'ETH', 'price': 97.76}
{'event_time': '2022-12-06T14:45:00.174739', 'ticker': 'SEBO', 'price': 85.87}
{'event_time': '2022-12-06T14:45:00.174754', 'ticker': 'SEBO', 'price': 64.41}
{'event_time': '2022-12-06T14:45:00.174769', 'ticker': 'ETH', 'price': 32.85}
{'event_time': '2022-12-06T14:45:00.174784', 'ticker': 'SEBO', 'price': 92.05}
{'event_time': '2022-12-06T14:45:00.174813', 'ticker': 'BTC', 'price': 88.71}
{'event_time': '2022-12-06T14:45:00.174824', 'ticker': 'ETH', 'price': 55.68}
{'event_time': '2022-12-06T14:45:00.174834', 'ticker': 'SEBO', 'price': 57.03}
{'event_time': '2022-12-06T14:45:00.174844', 'ticker': 'SEBO', 'price': 79.03}
{'event_time': '2022-12-06T14:45:00.174853', 'ticker': 'XRP', 'price': 9.02}
{'event_time': '2022-12-06T14:45:00.174863', 'ticker': 'DOGE', 'price': 83.16}
{'event_time': '2022-12-06T14:45:00.174872', 'ticker': 'BTC', 'price': 41.19}
{'event_time': '2022-12-06T14:45:00.174882', 'ticker': 'DOGE', 'price': 55.64}
{'event_time': '2022-12-06T14:45:00.174892', 'ticker': 'DOGE', 'price': 99.33}
{'event_time': '2022-12-06T14:45:00.174901', 'ticker': 'DOGE', 'price': 27.65}
{'event_time': '2022-12-06T14:45:00.174911', 'ticker': 'ETH', 'price': 34.51}
{'event_time': '2022-12-06T14:45:00.174920', 'ticker': 'ETH', 'price': 31.68}
{'event_time': '2022-12-06T14:45:00.174930', 'ticker': 'SEBO', 'price': 0.66}
{'event_time': '2022-12-06T14:45:00.174956', 'ticker': 'SEBO', 'price': 98.49}
{'event_time': '2022-12-06T14:45:00.174966', 'ticker': 'DOGE', 'price': 86.59}
{'event_time': '2022-12-06T14:45:00.174975', 'ticker': 'BTC', 'price': 44.61}
{'event_time': '2022-12-06T14:45:00.174985', 'ticker': 'SEBO', 'price': 23.19}
{'event_time': '2022-12-06T14:45:00.174994', 'ticker': 'ETH', 'price': 79.82}
{'event_time': '2022-12-06T14:45:00.175004', 'ticker': 'SEBO', 'price': 48.44}
{'event_time': '2022-12-06T14:45:00.175013', 'ticker': 'XRP', 'price': 0.21}
{'event_time': '2022-12-06T14:45:00.175023', 'ticker': 'XRP', 'price': 14.85}
{'event_time': '2022-12-06T14:45:00.175033', 'ticker': 'ETH', 'price': 64.76}
{'event_time': '2022-12-06T14:45:00.175043', 'ticker': 'XRP', 'price': 71.43}
{'event_time': '2022-12-06T14:45:00.175053', 'ticker': 'BNB', 'price': 7.81}
{'event_time': '2022-12-06T14:45:00.175062', 'ticker': 'ETH', 'price': 55.39}
{'event_time': '2022-12-06T14:45:00.175072', 'ticker': 'BTC', 'price': 41.46}
{'event_time': '2022-12-06T14:45:00.175082', 'ticker': 'DOGE', 'price': 69.09}
{'event_time': '2022-12-06T14:45:00.175091', 'ticker': 'BTC', 'price': 87.84}
{'event_time': '2022-12-06T14:45:00.175118', 'ticker': 'XRP', 'price': 9.17}
{'event_time': '2022-12-06T14:45:00.175136', 'ticker': 'ETH', 'price': 9.42}
{'event_time': '2022-12-06T14:45:00.175152', 'ticker': 'SEBO', 'price': 40.44}
{'event_time': '2022-12-06T14:45:00.175168', 'ticker': 'BNB', 'price': 45.66}
{'event_time': '2022-12-06T14:45:00.175184', 'ticker': 'ETH', 'price': 47.39}
{'event_time': '2022-12-06T14:45:00.175200', 'ticker': 'DOGE', 'price': 37.15}
{'event_time': '2022-12-06T14:45:00.175216', 'ticker': 'SEBO', 'price': 40.44}
```

### Question 3

Création depuis la console aws

Flux de données selim-bouhassatine-stock-input-stream créé avec succès.

Amazon Kinesis > Flux de données > selim-bouhassatine-stock-input-stream

### selim-bouhassatine-stock-input-stream [Info](#)

[Supprimer](#)

#### Résumé du flux de données

Statut ✓ Actif	Mode de capacité À la demande	ARN arn:aws:kinesis:eu-north-1:145740120431:stream/selim-bouhassatine-stock-input-stream	Heure de création 06 décembre 2022 à 15:50 UTC+1
Période de rétention des données 1 jour			

[Applications](#) | [Surveillance](#) | [Configuration](#) | [Visionneuse de données](#) | [Diffusion améliorée \(0\)](#)

#### Producteurs [Info](#)

## Question 4

### 4.1

Voici la commande pour supprimer

```
→ ~ aws kinesis delete-stream --stream-name selim-bouhassatine-stock-input-stream
→ ~
```

Sur le tableau de bord d'AWS il n'y a plus de data-stream

Amazon Kinesis > Flux de données

**Nouveau mode à la demande pour Kinesis Data Streams**  
Le mode à la demande élimine la nécessité d'allouer et de mettre à l'échelle manuellement vos flux de données. Avec ce mode, vos flux de données mettent automatiquement à l'échelle leur capacité d'écriture jusqu'à 200 MiB/seconde. [En savoir plus](#)

Flux de données (0) [Info](#)

Traiter les données en temps réel | Créer un flux de diffusion Firehose | Actions | [Créer un flux de données](#)

Rechercher des flux de données

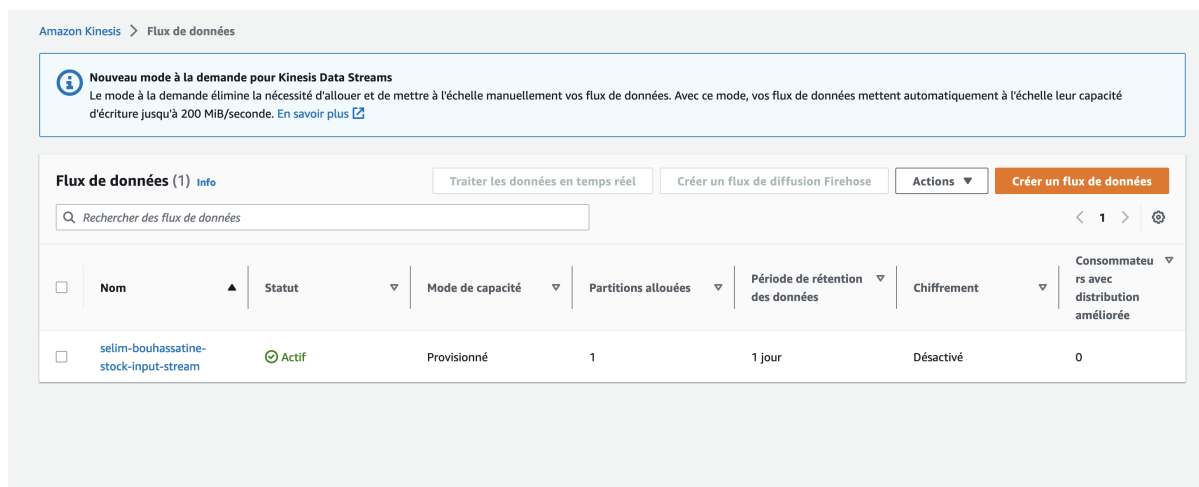
	Nom	Statut	Mode de capacité	Partitions allouées	Période de rétention des données	Chiffrement	Consommateurs avec distribution améliorée
Aucun flux de données							

### 4.2

Voici la commande pour créer un data-stream depuis AWS-CLI

```
→ ~ aws kinesis create-stream --stream-name selim-bouhassatine-stock-input-stream --shard-count 1
→ ~
```

On remarque bien la création du data-stream dans la console aws kinesis



## Question 5

### 5.1

Le message d'erreur "NoCredentialsError: Unable to locate credentials" indique que la bibliothèque `boto3` utilisée dans votre code Python ne parvient pas à trouver les informations d'identification nécessaires pour accéder aux ressources AWS.

### 5.2

Pour résoudre cette erreur :

Il existe plusieurs façons de résoudre ce problème. La première consiste à utiliser la commande `aws configure` pour configurer les informations d'identification d'un compte AWS sur votre ordinateur. Cette commande vous demandera votre identifiant d'accès, votre clé d'accès secrète et le nom de votre région par défaut. Une fois que vous avez entré ces informations, elles seront enregistrées dans un fichier de configuration sur votre ordinateur et `boto3` pourra les utiliser pour accéder aux ressources AWS.

Si vous ne souhaitez pas enregistrer ces informations d'identification sur votre ordinateur, vous pouvez également les fournir directement à l'objet `client` de `boto3` lorsque vous le créez. Voici un exemple de code montrant comment faire cela :

### 5.3

Il faut modifier `kinesis_client` et y ajouter les accès aws. Après correction le programme devient : On ajoute les accès Aws et modifie la région pour retrouver notre data-stream

```

import datetime
import json
import random
import boto3

STREAM_NAME = "input-stream"
REGION = "eu-north-1"

def get_data():
    return {
        'event_time': datetime.datetime.now().isoformat(),
        'ticker': random.choice(["BTC", "ETH", "BNB", "XRP", "DOGE", "SEBO"]),
        '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="partitionkey")

if __name__ == '__main__':
    generate('selim-bouhassatine-stock-input-stream', boto3.client('kinesis',
aws_access_key_id="AKI*****0Z",
aws_secret_access_key="qpUgT6X*****LSd",
region_name=REGION))

```

On remarque bien que les données arrive par le data-stream depuis l'onglet visionnaliser les flux de données



selim-bounassatine-stock-input-stream Info Supprimer

Résumé du flux de données

Statut	Mode de capacité	ARN	Heure de création
			06 décembre 2022 à 16:02

**Enregistrer les données** ✕

Numéro de séquence 49635854155105551618260520991983173308767951756898336770

ID de partition shardId-000000000000

Données brutes

JSON

Copier

```
{"event_time": "2022-12-06T15:40:37.241138", "ticker": "BTC", "price": 83.69}
```

Fermer

partitionkey	partitionkey	partitionkey	partitionkey
["event_time": "2022-12-06T15:...	06 décembre 2022 à 16:40:37 U...	4963585415510555161826052...	
["event_time": "2022-12-06T15:...	06 décembre 2022 à 16:40:37 U...	4963585415510555161826052...	

On retrouve les données dans la shard 1 que nous avons créer

Examen Selim Bouhassatine

9

selim-bouhassatine-stock-input-stream
[Info](#)

Supprimer

Résumé du flux de données

Statut  
Actif

Mode de capacité  
Provisionné

ARN  
arn:aws:kinesis:eu-north-1:145740120431:stream/selim-bouhassatine-stock-input-stream

Heure de création  
06 décembre 2022 à 16:02 UTC+1

Applications
Surveillance
Configuration
Visionneuse de données
Diffusion améliorée (0)

Partition
shardId-000000000000

Position de départ
[Info](#)
Dernier

Obtenir des enregistrements

Enregistrements (4)
Prochains enregistrements

Partition : shardId-000000000000
Position de départ : Dernier

Clé de parti...	Données	Horodatage approximatif de...	Numéro de séquence
partitionkey	{"event_time": "2022-12-06T15:..."}	06 décembre 2022 à 16:40:37 U...	4963585415510555161826052...
partitionkey	{"event_time": "2022-12-06T15:..."}	06 décembre 2022 à 16:40:37 U...	4963585415510555161826052...
partitionkey	{"event_time": "2022-12-06T15:..."}	06 décembre 2022 à 16:40:37 U...	4963585415510555161826052...
partitionkey	{"event_time": "2022-12-06T15:..."}	06 décembre 2022 à 16:40:37 U...	4963585415510555161826052...

## Question 6

Voici le [main.tf](#) à écrire pour créer un data-stream depuis terraform

```
# Définissez les variables d'entrée
variable "stream_name" {
    type = string
}

variable "partition_count" {
    type = number
    default = 1
}

# Définissez le type de ressource à créer
resource "aws_kinesis_stream" "selim-bouhassatine-stock-input-stream" {
    name           = var.stream_name
    shard_count    = var.partition_count
    tags = {
        Name = "selim.bouhassatine@etu.u-pec.fr"
        Owner = "selim_dev"
    }
}
```

```
}
}
```

Flux de données (1) info

Traiter les données en temps réel

Créer un flux de diffusion Firehose

Actions ▾

Créer un flux de données

Rechercher des flux de données

< 1 > ⚙

<input type="checkbox"/>	Nom ▲	Statut ▾	Mode de capacité ▾	Partitions allouées ▾	Période de rétention des données ▾	Chiffrement ▾	Consommateurs avec distribution améliorée ▾
<input type="checkbox"/>	<a href="#">selim-bouhassatine-stock-input-stream</a>	✔ Actif	Provisionné	1	1 jour	Désactivé	0

## Question 7

### 7.1

On ajoute ce que nous avons écrit pour la première question dans le [main.tf](#)

```
# Définissez les variables d'entrée
variable "stream_name" {
    type = string
}

variable "partition_count" {
    type = number
    default = 1
}

# Définissez le type de ressource à créer
resource "aws_kinesis_stream" "selim-bouhassatine-stock-input-stream" {
    name           = var.stream_name
    shard_count    = var.partition_count
    tags = {
        Name = "selim.bouhassatine@etu.u-pec.fr"
        Owner = "selim_dev"
    }
}

resource "aws_instance" "ec2_vm" {
    ami           = "ami-02384a901b5df8024"
    instance_type = "t3.micro"
    key_name       = "mykey"
    tags = {
        Name = "selim.bouhassatine@etu.u-pec.fr"
        Owner = "selim_dev"
    }
    vpc_security_group_ids = ["${aws_security_group.default.id}"]
}
```

```
# Default ec2 user is "ec2-user"
```

Par faute de temps pour faire la 7.2 il faut utiliser votre fichier main.tf associez au main.tf de la question 6

Le probleme est que le scp ne fonctionne pas, il faudrait utiliser un wget et mettre le fichier dans un serveur accessible en public.