Повний посібник з AWS

Загальний огляд AWS та основи

Amazon Web Services (AWS) — це набір хмарних сервісів від компанії Amazon, що надає платформи для зберігання, обробки та управління даними через інтернет. AWS пропонує широкий спектр послуг, включаючи обчислювальні потужності, зберігання даних, бази даних, аналітику, мережеві рішення, інструменти розробки, штучний інтелект та багато іншого.

З AWS ви отримуєте:

- **Гнучкість:** Запускайте сервери, бази даних та інші сервіси за лічені хвилини.
- **Масштабованість:** Збільшуйте чи зменшуйте потужності автоматично, залежно від навантаження.
- **Оплата за використання:** Ви платите лише за те, що споживаєте, без великих початкових інвестицій.

Основні поняття

- **Регіони (Regions):** Географічні розташування з дата-центрами. Обирайте найближчий до ваших користувачів (наприклад, eu-central-1 у Франкфурті).
- **Зони доступності (Availability Zones):** Ізольовані дата-центри в межах регіону. Використання кількох зон забезпечує відмовостійкість.

Основні сервіси AWS

- **EC2 (Elastic Compute Cloud):** обчислювальні потужності на вимогу.
- **S3 (Simple Storage Service):** масштабоване хмарне сховище.
- **RDS (Relational Database Service):** управління реляційними базами даних.
- **Lambda:** виконання коду без управління серверами.
- **DynamoDB:** масштабована NoSQL база даних.
- **VPC (Virtual Private Cloud):** ізольовані мережеві ресурси.

Де використовують AWS?

AWS використовують у різних галузях і сферах для різних цілей, таких як:

- **Інформаційні технології:** для хостингу веб-сайтів, управління базами даних, обчислень та аналізу даних.
- **Фінанси:** для обробки транзакцій, управління даними клієнтів та ризиками.
- **Охорона здоров'я:** для зберігання медичних даних, аналізу медичних зображень та досліджень.
- **Медіа та розваги:** для стрімінгу відео, зберігання контенту та обробки великих обсягів даних.

Основні знання для роботи з AWS

Для ефективної роботи з AWS важливо мати базові знання та навички у наступних областях:

- **Хмарні обчислення:** розуміння основ хмарних технологій та моделей хмарних послуг (IaaS, PaaS, SaaS).
- **Безпека:** управління доступами і правами (IAM), шифрування даних, моніторинг та аудит безпеки.
- **Інструменти розробки:** використання AWS SDK, CLI, CloudFormation для автоматизації та управління ресурсами.

Частина 1: Початок роботи

1.1 Створення безкоштовного облікового запису

AWS пропонує безкоштовний рівень (**AWS Free Tier**), що дозволяє вам досліджувати та випробовувати їхні послуги.

Як зареєструватися:

1. Перейдіть на офіційний сайт AWS.
2. Натисніть кнопку **"Create a Free Account"**.
3. Завершіть кроки реєстрації, ввівши вашу електронну пошту та пароль. AWS вимагає дані кредитної картки для верифікації, але в рамках Free Tier ви не будете платити.
4. Після завершення ви будете перенаправлені на вітальну сторінку.

1.2 Налаштування сповіщень про витрати (Billing Alarms)

Це критично важливий крок, який допоможе уникнути неочікуваних рахунків.

1. У Консолі управління AWS перейдіть до сервісу **CloudWatch**.
2. У лівому меню оберіть **"Alarms"** -> **"All alarms"**.
3. Натисніть **"Create alarm"**.
4. Виберіть метрику: Сервіс: **"Billing"**, Метрика: **"EstimatedCharges"**.
5. Встановіть поріг: **"Greater"** (Більше ніж) **1.0** (1 долар).
6. Налаштуйте сповіщення, вказавши свою електронну пошту. AWS надішле вам лист для підтвердження.
7. Збережіть сигналізацію.

Тепер ви отримаєте лист, щойно ваші витрати перевищать $1.

1.3 Швидкий старт з AWS CLI та Boto3 (Python SDK)

Для автоматизації та програмної взаємодії з AWS використовуйте AWS CLI або AWS SDK (наприклад, Boto3 для Python).

Встановлення та налаштування AWS CLI

1. Встановіть AWS CLI за допомогою pip:

```
pip install awscli
```

2. Налаштуйте облікові дані. Для цього вам потрібні Access Key ID та Secret Access Key, які можна згенерувати в сервісі IAM.

```
aws configure
```

Виконавши цю команду, ви по черзі введете ваш ключ, секретний ключ, регіон за замовчуванням та формат виводу.

Встановлення та налаштування Boto3

Boto3 – це офіційний AWS SDK для Python.

1. Встановіть Boto3 через pip:

```
pip install boto3
```

2. Boto3 автоматично використовує облікові дані, налаштовані через aws configure.

Частина 2: Основні сервіси та робота з ними

2.1 Робота з Amazon S3 (Хмарне сховище)

Amazon S3 (Simple Storage Service) — це об'єктне сховище, яке ідеально підходить для зберігання будь-яких файлів: зображень, відео, документів, бекапів.

Що таке Бакет?
Бакет (bucket) - це основний контейнер для зберігання даних у S3. Уявіть його як папку, що може зберігати будь-які типи даних. Імена бакетів повинні бути унікальними глобально в межах всієї системи Amazon S3.

Взаємодія з S3 через AWS CLI

- **Створення бакету:**

```
aws s3 mb s3://моя-нова-унікальна-назва-бакету
```

- **Завантаження файлу:**

```
aws s3 cp моїй-файл.txt s3://моя-нова-унікальна-назва-бакету/
```

- **Перегляд вмісту бакету:**

```
aws s3 ls s3://моя-нова-унікальна-назва-бакету
```

- **Видалення файлу:**

```
aws s3 rm s3://моя-нова-унікальна-назва-бакету/моїй-файл.txt
```

Взаємодія з S3 через Boto3 (Python)

Ось приклади використання Boto3 для роботи з S3.

Приклад 1: Створення бакету та завантаження файлу

```python
import boto3

# Ініціалізація клієнта S3
s3 = boto3.client('s3')

# Створення бакету
s3.create_bucket(Bucket='my-example-bucket')

# Завантаження файлу в бакет
s3.upload_file('path/to/localfile.txt', 'my-example-bucket',
'uploadedfile.txt')
```

Приклад 2: Завантаження файлу з бакету

```
import boto3

# Ініціалізація клієнта S3
s3 = boto3.client('s3')

# Завантаження файлу з бакету
s3.download_file('my-example-bucket', 'uploadedfile.txt',
'path/to/localfile.txt')
```

Приклад 3: Перегляд вмісту бакету

```
import boto3

# Ініціалізація клієнта S3
s3 = boto3.client('s3')

# Отримання списку об'єктів у бакеті
response = s3.list_objects_v2(Bucket='my-example-bucket')
for obj in response.get('Contents', []):
    print(obj['Key'])
```

2.2 Робота з Amazon EC2 (Віртуальні сервери)

EC2 (Elastic Compute Cloud) надає вам віртуальний комп'ютер, який ви можете використовувати для запуску програм, вебсайтів тощо. Інстанс — це назва віртуального сервера в AWS.

Основні компоненти EC2:

- **Інстанси:** Віртуальні сервери.
- **AMI (Amazon Machine Image):** Шаблон, що містить ОС та програмне забезпечення.
- **Ключові пари (Key Pairs):** Використовуються для безпечного підключення до інстансів через SSH.

Запуск найдешевшого інстансу (Free Tier)

1. У Консолі AWS перейдіть до сервісу **EC2**.
2. Натисніть **"Launch instances"**.

3. Оберіть AMI з позначкою **"Free tier eligible"** (наприклад, Ubuntu Server).
4. Виберіть тип інстансу **t2.micro**.
5. Створіть нову пару ключів (.pem файл) і завантажте її. **Це ваш єдиний спосіб доступу!**
6. Налаштуйте **Security Group**, дозволивши трафік **SSH** та **HTTP**.
7. Запустіть інстанс.

Підключення до EC2 інстансу через термінал

1. Зробіть ваш файл ключа приватним:

```
chmod 400 my-ec2-key.pem
```

2. Підключіться до інстансу, використовуючи його публічну IP-адресу:
   ssh -i "my-ec2-key.pem" ubuntu@ВАША_ПУБЛІЧНА_IP_АДРЕСА

Взаємодія з EC2 через Boto3 (Python)

Приклад 1: Створення інстансу

```
import boto3

# Ініціалізація клієнта EC2
ec2 = boto3.resource('ec2')
instance = ec2.create_instances(
    ImageId='ami-0abcdef1234567890', # Замініть на реальний AMI ID
    MinCount=1,
    MaxCount=1,
    InstanceType='t2.micro',
    KeyName='my-key-pair'
)
print(instance[0].id)
```

Приклад 2: Запуск і зупинка інстансів

```
import boto3

ec2 = boto3.resource('ec2')
```

```
# Зупинити інстанс за його ID
ec2.Instance('i-1234567890abcdef0').stop()

# Запустити інстанс за його ID
ec2.Instance('i-1234567890abcdef0').start()
```

2.3 Робота з Amazon RDS (Бази даних)

Amazon RDS (Relational Database Service) — це керована служба для реляційних баз даних (MySQL, PostgreSQL, SQL Server тощо). RDS автоматизує рутинні завдання, такі як резервне копіювання, оновлення та масштабування.

Взаємодія з RDS через Boto3 (Python)

Приклад 1: Створення бази даних

```
import boto3

# Ініціалізація клієнта RDS
rds = boto3.client('rds')

# Створення інстансу бази даних MySQL
response = rds.create_db_instance(
    DBInstanceIdentifier='mydbinstance',
    MasterUsername='masteruser',
    MasterUserPassword='masterpassword',
    DBInstanceClass='db.t2.micro',
    Engine='mysql',
    AllocatedStorage=20
)
print(response)
```

Приклад 2: Підключення до бази даних

```
import pymysql

# Параметри підключення (замініть на ваші реальні дані)
host = 'mydbinstance.c9akciq32.rds.amazonaws.com'
```

```
user = 'masteruser'
password = 'masterpassword'
database = 'mydatabase'

# Підключення до бази даних
connection = pymysql.connect(
    host=host,
    user=user,
    password=password,
    database=database
)

cursor = connection.cursor()
cursor.execute('SELECT DATABASE()')
db = cursor.fetchone()
print(f"Connected to: {db}")
```

2.4 Робота з Amazon DynamoDB та Lambda

Amazon DynamoDB — це масштабована NoSQL база даних. **Amazon Lambda** дозволяє запускати код без управління серверами.

Приклад: Створення таблиці DynamoDB через Boto3

```
import boto3

dynamodb = boto3.resource('dynamodb')
table = dynamodb.create_table(
    TableName='Users',
    KeySchema=[
        {'AttributeName': 'username', 'KeyType': 'HASH'},  # Partition key
        {'AttributeName': 'last_name', 'KeyType': 'RANGE'}  # Sort key
    ],
    AttributeDefinitions=[
        {'AttributeName': 'username', 'AttributeType': 'S'},
        {'AttributeName': 'last_name', 'AttributeType': 'S'}
    ],
    ProvisionedThroughput={
        'ReadCapacityUnits': 5,
        'WriteCapacityUnits': 5
```

```
    }
)
print("Table status:", table.table_status)
```

Приклад: Створення Lambda-функції через Boto3

```
import boto3

lambda_client = boto3.client('lambda')
response = lambda_client.create_function(
    FunctionName='MyLambdaFunction',
    Runtime='python3.8',
    Role='arn:aws:iam::account-id:role/execution_role', # Замініть на вашу роль
    Handler='lambda_function.lambda_handler',
    Code={
        'ZipFile': open('function.zip', 'rb').read(), # Шлях до вашого zip-файлу з кодом
    },
)
print("Lambda function created:", response)
```

Частина 3: Практичні проекти

Проєкт 1: Веб-додаток на EC2 з базою даних RDS

1. **Запуск EC2-інстансу** для розміщення вашого веб-додатка (наприклад, на Flask або Django).
2. **Створення RDS-інстансу** (наприклад, MySQL) для зберігання даних.
3. **Підключення** веб-додатка на EC2 до бази даних на RDS.

Проєкт 2: Система повідомлень з використанням SQS та SNS

1. **Створення черги SQS** для зберігання повідомлень від вашого додатка.
2. **Налаштування теми SNS** для відправки push-сповіщень.
3. **Інтеграція** вашого додатка з цими сервісами для обробки та відправки повідомлень.

Проєкт 3: Обробка файлів за допомогою S3 та Lambda

1. **Створення S3-бакету** для завантаження файлів.

2. **Створення Lambda-функції**, яка буде обробляти файли (наприклад, змінювати розмір зображень) після їх завантаження.
3. **Налаштування тригера** в S3, щоб він викликав Lambda-функцію щоразу, коли новий файл завантажується в бакет.

Ресурси для подальшого навчання

- **Офіційна документація AWS:** AWS Documentation
- **Онлайн-курси та відеоуроки:** Coursera, Udemy, edX та YouTube-канал AWS.
- **Спільноти та форуми:** Stack Overflow та офіційний форум AWS.