**Лабораторна робота № 1**

**(Частина 1)**

**Створення складових Інтернету речей в середовищі AWS IoT Core**

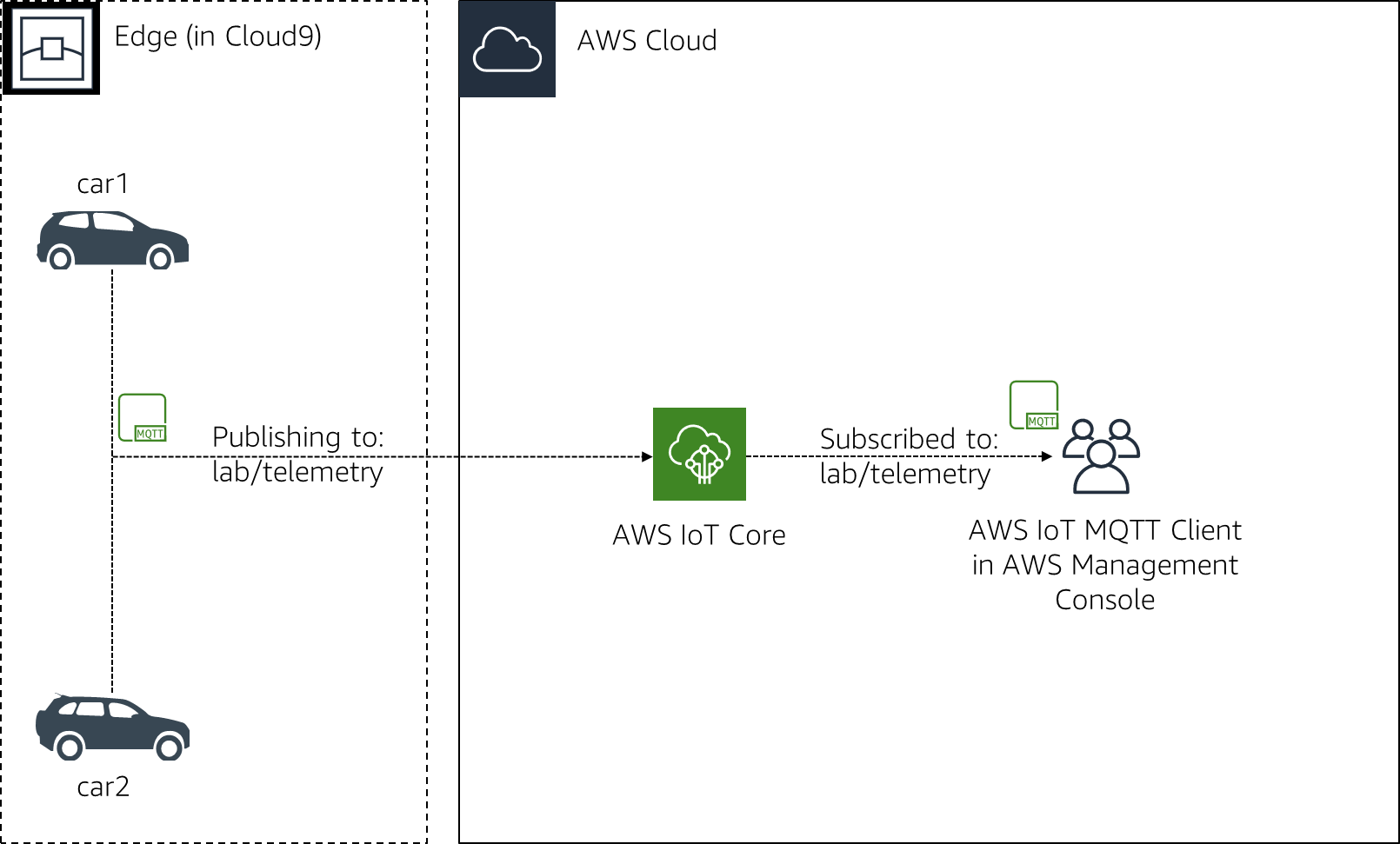
**УВАГА! Ви отримали доступ до ресурсів AWS в рамках AWS Academy, налаштованих у цьому середовищі, для вивчення даного курсу. Ваш бюджет обмежено (100 дол. США), тому вам слід бути обережними, щоб запобігти стягненням, які надто швидко вичерпають бюджет. Якщо ви перевищите свій бюджет, ви втратите доступ до свого середовища та всю свою роботу.**

**Завдання**

В лабораторній роботі необхідно створити дві речі Cars Things і підключити їх до сервісу AWS IoT Core так, щоб вони могли надсилати дані телеметрії в тему IoT. Щоб підключити автомобілі, необхідно створити річ IoT, сертифікат і політику. Річ буде представляти автомобіль. Сертифікат використовуватиметься для автентифікації в AWS IoT Core, а Політика визначатиме, що може робити автомобіль після автентифікації. Перший автомобіль буде створено за допомогою Консолі керування AWS, другий автомобіль – за допомогою інтерфейсу командного рядка AWS (CLI).

Для моделювання необхідно створити середовище Cloud9. Буде завантажено код автомобілів, їхні сертифікати IoT та запущено двигуни автомобілів. Буде використано клієнт AWS IoT MQTT в Консолі керування AWS, щоб підписатися на тему телеметрії IoT, щоб підтвердити, що автомобілі надсилають дані.

На схемі нижче показано ресурси та потік даних, які будуть створені під час виконання завдання.



**Хід роботи**

**1. Створити IAM Policy**

Для початку створіть користувача та політику IAM, специфічну для цього курсу.

У цьому розділі ви створите політику IAM. Політики, якими керує клієнт, забезпечують більш точний контроль над вашими політиками, ніж політики, якими керує AWS. Ця політика матиме дозволи, специфічні для ресурсів AWS, які вам потрібні для цього курсу.

1. В AWS Management Console, натисніть **Services**, а потім **IAM**, щоб відкрити панель IAM.
2. В меню навігації зліва, натисніть **Policies**.
3. Натисніть **Create policy**.
4. Перейдіть на вкладку **JSON**.
5. У текстовому полі редактора замініть зразок політики за замовчуванням на наступний:

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"greengrass:\*",

"iot:\*",

"iotanalytics:\*",

"cloud9:\*",

"lambda:\*",

"s3:\*",

"sns:\*",

"iam:\*",

"cognito-identity:\*",

"cognito-sync:\*",

"cognito-idp:\*",

"logs:\*",

"ec2:\*",

"cloudwatch:\*",

"kms:ListAliases",

"kms:DescribeKey",

"cloudformation:DescribeStackResources",

"tag:getResources"

],

"Resource": "\*"

}

]

}

1. Натисніть **Review Policy**.
2. В полі **Name**, введіть labIoTPolicy.
3. Натисніть **Create policy**.

Ви створили IAM policy.

**2. Створити користувача IAM та додати політику (у разі виникнення помилки під час виконання цього пункту, пропускайте його і виконуйте завдання далі).**

У цьому розділі ви створите користувача IAM і додасте до нього політику.

1. В AWS Management Console, натисніть **Services**, потім натисніть **IAM** для переходу в панель IAM.
2. В меню навігації зліва, натисніть **Users**.
3. Натисніть **Add user**.
4. В полі **User name** , введіть labIoTUser.
5. Для **Access type**, виберіть **AWS Management Console access**.
6. Для **Console password**, виберіть **Custom password** та введіть пароль на власний вибір. **Запам’ятайте пароль**.
7. Видаліть **відмітку** біля **User must create a new password at next sign-in**.
8. Натисніть **Next: Permissions**.
9. В розділі **Set permissions** , натисніть **Attach existing policies directly**.
10. В полі пошуку для **Filter**, введіть labIoTPolicy.
11. Додайте відмітку поряд з **labIoTPolicy** у відфільтрованому списку.
12. Натисніть **Next: Tags**.
13. Натисніть **Next: Review**.
14. Перегляньте інформацію та натисніть **Create user**. Повинно з’явитись повідомлення про успішне створення.
15. Зверніть увагу на **sign-in URL** в повідомленні. Це спеціальна URL-адреса для користувачів IAM, яка містить ідентифікатор облікового запису (account ID).
16. Натисніть **sign-in URL** в повідомленні. Це призведе до виходу з системи.
17. **Sign in** як користувач IAM labIoTUser .

**3. Створити середовище AWS Cloud9**

У цьому розділі ви створите середовище AWS Cloud9.

1. В AWS Management Console, натисніть **Services**, а потім натисніть **Cloud9**, щоб відкрити панель Cloud9.
2. Переконайтесь, що ви знаходитесь в одному з регіонів: **Frankfurt, Ireland, N. Virginia, Ohio, Oregon or Tokyo**. На разі, це єдині регіони з усіма сервісами, які будуть використовуватися: Amazon Cognito, Amazon EC2, Amazon S3, Amazon SNS, AWS Cloud9, AWS IAM, AWS IoT Analytics, AWS IoT Core, AWS IoT Greengrass, AWS Lambda. Оскільки всі ресурси повинні бути в одному регіоні, щоб завдання працювало, ви повинні використовувати регіон, де доступні всі ці служби.
3. Натисніть **Create environment** в правому верхньому кутку.
4. В полі **Name**, введіть IoTOnAWS.
5. Натисніть **Next step**.
6. На сторінці Configure Settings, залиште параметри за замовчуванням, та натисніть **Next step**.
7. Перегляньте параметри, та натисніть **Create environment**. Це запустить ваше середовище AWS Cloud9 протягом кількох хвилин.

Зверніть увагу, що цей екземпляр Cloud9 автоматично вимкнеться через 30 хвилин, якщо його не використовувати. Уся ваша робота буде збережена до повторного запуску вашого середовища.

**4. Налаштувати середовище Cloud9, завантажити код для речі Car та AWS IoT CA Public Cert**

У цьому розділі ви встановите пакет Node для роботи коду автомобіля, завантажте код автомобіля та налаштуєте структуру сховища для автомобілів.

1. Встановіть AWS IoT Device SDK Node запустивши відповідну команду в AWS Cloud9 **terminal**. Термінал знаходиться внизу сторінки. Це вкладка *bash*  із запитом *labIoTUser:~/environment: $*.

npm install aws-iot-device-sdk

Повідомлення про відсутність файлів package.json можна ігнорувати.

1. Створіть структуру сховища для застосунків автомобілів. Оскільки буде 2 автомобілі, ви створите 2 папки. Виконайте наступні команди у AWS Cloud9 **terminal**.

mkdir ~/environment/car1**;**

mkdir ~/environment/car2

1. Завантажте та скопіюйте код програми в кожну папку автомобіля, виконавши наведені нижче команди в AWS Cloud9 **terminal**:

cd ~/environment

wget https://aws-tc-largeobjects.s3.amazonaws.com/OTP-AWS\_D5-2019/v1.0/code/exercise-1.1.js

cp exercise-1.1.js car1/

cp exercise-1.1.js car2/

rm exercise-1.1.js

1. Завантажте AWS IoT Certificate Authority Public Certificate, який пізніше використовуватиметься в коді та підпише IoT Certificates, які ви створите в наступному розділі. Виконайте наступні команди в Cloud9 **terminal**:

cd ~/environment

wget -O root-CA.crt https://www.amazontrust.com/repository/AmazonRootCA1.pem

**5. Створити Car 1 IoT Thing, сертифікат та політику**

In this section you will use the AWS Management Console to create all of the resources required for your Car 1 to connect to AWS IoT. This includes the the Car Thing, the Certificate and the Policy. We could use the wizard to create many of these resources, but instead, you will do it manually to see how each of the components is attached to each other. You will finish by uploading the Certificate and Private Key that you generated as part of these steps to Cloud9.

У цьому розділі ви використовуватимете AWS Management Console, щоб створити всі ресурси, необхідні для підключення Car 1 до AWS IoT. Це включає в себе автомобіль, сертифікат і політику. Ви завершите завантаженням до Cloud9 сертифіката та приватного ключа, які ви згенеруєте в рамках цих кроків.

**5.1 Створити IoT річ**

У цьому розділі ви створите річ, що представляє автомобіль.

1. В AWS Management Console, натисніть **Services**, потім натисніть **IoT Core** для відкриття IoT Console.
2. Переконайтесь, що знаходитесь в тому ж **Region** , що і Cloud9.
3. Натисніть **Get started**.
4. Розгорніть **Manage** в лівому меню.
5. Натисніть **Things**.
6. Натисніть  **Register a thing**.
7. Натисніть  **Create a single thing** (будь-яка з двох кнопок з такою назвою працює).
8. В полі **Name**, введіть car1 та натисніть **Next**.
9. Натисніть **Create thing without certificate**, щоб уникнути створення сертифіката за допомогою майстра.

Річ car1 створена.

**5.2 Створити IoT Політику**

У цьому розділі ви створите політику для авторизації, яка використовуватиметься в наступному розділі.

1. Розгорніть **Secure** в лівому меню.
2. Натисніть **Policies**.
3. Натисніть **Create a policy**.
4. В полі **Name**, введіть labPolicy.
5. Натисніть **Advanced mode**.
6. **Замініть** зразок політики наведеною нижче політикою, яка дозволяє підключатися до вашої кінцевої точки AWS IoT Core, публікувати та підписуватися на тему IoT, отримувати повідомлення від AWS IoT після підписки та використовувати API Discover від Greengrass, який буде використано в наступних частинах лабораторної роботи.

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"iot:Connect",

"iot:Publish",

"iot:Subscribe",

"iot:Receive",

"greengrass:Discover"

],

"Resource": [

"\*"

]

}

]

}

1. Натисніть **Create**.

Ви створили політику для авторизації.

**5.3 Створити IoT Сертифікат**

У цьому розділі ви створите сертифікат, який буде використовуватися для аутентифікації.

1. Розгорніть **Certificates**.
2. Натисніть **Create a certificate**.
3. Натисніть **Create certificate,** щоб автоматично згенерувати сертифікат, відкритий ключ і закритий ключ за допомогою центру сертифікації AWS IoT, який ви потім завантажите. Для простоти використовуйте створення сертифіката в один клік.
4. Перш ніж перейти до наступного екрана, переконайтеся, що ви виконали наведені нижче дії, інакше вам доведеться створювати сертифікат заново.
   1. Натисніть кнопку **Activate**  для активації сертифіката для того, щоб його можна було використовувати пізніше для підключення до AWS IoT Core.
   2. Натисніть на посиланні **Download** поряд з **A certificate for this thing**.
   3. **Перейменуйте** файл на certificate.pem.crt.
   4. Натисніть на посиланні **Download** поряд з **A private key**. Зауважте, що відкритий ключ вам не знадобиться.
   5. **Перейменуйте** файл на private.pem.key.
5. Натисніть **Done**.

Тепер у вас є сертифікат і закритий ключ, які можна використовувати для підключення до кінцевої точки AWS IoT Core. Ви можете побачити, що сертифікат неактивний, якщо ви оновите сторінку, він повинен відображатися як активний.

**5.4 Приєднати політику та річ до сертифіката**

У цьому розділі ви додасте політику до свого сертифіката, щоб додати авторизації та приєднати річ car1, яка пов’язана з цим сертифікатом.

1. Натисніть на сертифікаті, який ви щойно створили.
2. Натисніть **Actions > Attach policy**.
3. Виберіть **labPolicy** та натисніть **Attach**.
4. Натисніть **Actions > Attach thing**.
5. Виберіть **car1** та натисніть **Attach**.

**5.5 Завантажити сертифікат і закритий ключ до Cloud9**

У цьому розділі ви завантажите сертифікат і закритий ключ, які тепер пов’язані з car1, в Cloud9 за допомогою функції завантаження.

1. В лівому меню консолі Cloud9, виберіть папку **car1**.
2. Натисніть **File > Upload Local Files...**.
3. Натисніть **Select files**.
4. **Завантажте** файли certificate.pem.crt та private.pem.key

Сертифікат car1 і закритий ключ тепер мають бути в папці car1.

**6. Створити Car 2 IoT Thing, сертифікат та політику**

У цьому розділі ви використовуватимете AWS CLI з терміналу Cloud9, щоб створити всі ресурси, необхідні для підключення car 2 до AWS IoT. Це включає річ та сертифікат. Політика вже створена на попередньому кроці, тому ви будете використовувати її повторно.

**6.1 Створити ресурси для car2 IoT Core за допомогою CLI**

1. В Cloud9 **terminal**, введіть наступні команди:

cd ~/environment/car2

aws iot create-thing --thing-name car2

1. Для створення сертифікату введіть наступну команду:

aws iot create-keys-and-certificate --set-as-active --certificate-pem-outfile certificate.pem.crt --private-key-outfile private.pem.key

Ця команда помістить сертифікат і закритий ключ у файли certificate.pem.crt і private.pem.key відповідно. Він також виведе certificateArn, який ви повторно використаєте в наступній команді.

1. Щоб прикріпити політику до сертифіката, виконайте наступну команду. **Замініть** certificateArn\_changeme значенням атрибута certificateArn з результату попередньої команди. Наприклад:  *arn:aws:iot:us-east-1:1234567890:cert/0f11db22dafacda87be0940dd5b2e010635916f541461ccf2d1c56ced0f343ee*

aws iot attach-policy --policy-name labPolicy --target certificateArn\_changeme

Команда не повинна повертати нічого, якщо спрацювала.

1. Щоб прикріпити річ car2 до сертифіката, виконайте наступну команду. **Замініть certificateArn\_changeme** на значення атрибута certificateArn з результатів команди створення сертифіката. Наприклад: *arn:aws:iot:region:1234567890:cert/0f11db22dafacda87be0940dd5b2e010635916f541461ccf2d1c56ced0f343ee*

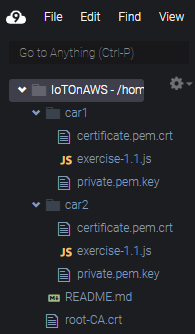
aws iot attach-thing-principal --thing-name car2 --principal certificateArn\_changeme

Команда не повинна повертати нічого, якщо спрацювала.

Ви створили річ car2, сертифікат і закритий ключ для аутентифікації та приєднали labPolicy для авторизації команд для подальшого виконання.

**7. Виконате код і перевірити телеметрію**

На цьому етапі у середовищі Cloud9 повинна бути структура каталогів, яка виглядає так:



У цьому розділі ви виконаєте код обох автомобілів і перевірите, що дані телеметрії надсилаються обома автомобілями за допомогою клієнта AWS IoT MQTT в AWS Management Console.

Код (exercise-1.1.js) потребує ще одного ресурсу для зв’язку з AWS IoT, і це конкретна кінцева точка AWS IoT. Він буде збережений у файлі, який використовуватиметься в наступних лабораторних роботах. Для автентифікації на кінцевій точці AWS IoT вона використовуватиме *certificate.pem.crt*  (сертифікат), *private.pem.key*  (закритий ключ) і *root-CA.crt*  (публічний сертифікат центру сертифікації). Потім він підключається до AWS IoT і починає публікувати випадкові дані телеметрії кожні 5 секунд у темі *lab/telemetry*.

**7.1 Виконати код**

У цьому розділі ви спочатку отримаєте свою конкретну кінцеву точку AWS IoT, яка була автоматично створена разом з річчю IoT. Потім необхідно відкрити два термінали Cloud9, щоб імітувати підключення до AWS IoT з обох автомобілів.

1. В Cloud9 **terminal**, виконайте команду, щоб отримати кінцеву точку AWS IoT, яка потім буде збережена у файлі endpoint.json:

aws iot describe-endpoint --endpoint-type iot:Data-ATS > ~/environment/endpoint.json

Команда не повинна повертати нічого, якщо спрацювала.

1. В Cloud9, створіть **New Terminal**.
2. В лівому терміналі виконайте наступні команди, щоб запустити код для car1. Сертифікат і закритий ключ Car1, кореневий центр сертифікації та кінцева точка AWS IoT будуть зчитуватися зі створених вами файлів.

cd ~/environment/car1

node exercise-1.1.js

Результат повинене бути наступним:

Connected to AWS IoT

Sending car telemetry data to AWS IoT for car1

Sending car telemetry data to AWS IoT for car1

...

1. В правому терміналі виконайте наступні команди, щоб запустити код для car2. Сертифікат і закритий ключ Car2, кореневий центр сертифікації та кінцева точка AWS IoT будуть зчитуватися зі створених вами файлів.

cd ~/environment/car2

node exercise-1.1.js

Результат повинене бути наступним:

Connected to AWS IoT

Sending car telemetry data to AWS IoT for car2

Sending car telemetry data to AWS IoT for car2

...

Зараз обидві машини успішно надсилають телеметричні дані до теми *lab/telemetry*.

**7.2 Підписатися на тему lab/telemetry**

В цьому розділі використовується AWS IoT MQTT Client в AWS Management Console, щоб підписатися на тему *lab/telemetry* . Під час підключення до MQTT Client використовуються хвилини підключення. Не забудьте відключитися, після завершення використання, вийшовши з цієї сторінки.

1. В AWS Management Console, натисніть **Services**, потім натисніть **IoT Core,**  щоб відкрити IoT Core console.
2. Натисніть **Test** в лівому меню. Відкриється AWS IoT MQTT Client, де можна побачити всі теми, до яких ви маєте доступ. Цей Client автоматично підключатиметься до кінцевої точки.
3. В полі **Subscription topic**, введіть lab/telemetry.
4. Натисніть **Subscribe to topic**.

У наступні 5 секунд ви повинні побачити дані, які публікуються обома автомобілями в інтерфейсі. Ви можете побачити, який автомобіль надсилає дані, за атрибутом  *device* .

**Знімок цього екрану = звіт про виконану роботу**

**8. Зупинити використання створених ресурсів**

Поки до служби IoT не передаються з’єднання та дані, плата за використання цієї служби не стягується.

Середовище Cloud9 використовує екземпляр t2.micro EC2 і том Elastic Block Storage (EBS) об’ємом 8 Гб, за який з вас стягується плата за використання Cloud9. Ви будете оплачувати час роботи екземпляра EC2 для екземпляра t2.micro і плату за том EBS, поки існує ваше середовище.

В усіх наступних лабораторних роботах використовуватимуться ресурси, створені в ході цієї роботи. Рекомендується зберегти всі створені ресурси і дозволити Cloud9 зупинитися самостійно через 30 хвилин бездіяльності, щоб ви могли зберегти середовище.

Якщо ви вирішите не зберігати ці ресурси, вам доведеться виконати всі кроки ще раз перед іншими завданнями.

**8.1 Зупинити cars**

1. **Натисніть Ctrl-c** к кожному вікні Cloud9 **terminal,** щоб зупинити взаємодію із AWS IoT.

**8.2 Зупинити MQTT Client**

1. **Вийдіть зі сторінки** **AWS IoT MQTT Client,** щоб від’єднатися від клієнта.

**8.3 Зупинити середовище Cloud9**

Середовище Cloud9 автоматично вимкнеться після 30 хвилин бездіяльності. Щоб середовище Cloud9 вважалося неактивним, потрібно закрити вкладку браузера. Усі налаштування будуть збережені.

1. Закрийте вкладку браузера, в якій запушено середовище.

Оскільки операційною системою є Amazon Linux, виставляється рахунок за секунду протягом 30 хвилин бездіяльності. Бажано зупинити екземпляр EC2, який запускає середовище Cloud9. Це не вплине на майбутні завдання.

1. В AWS Management Console, натисніть **Services**, потім натисніть **EC2,** щоб відкрити EC2 console.
2. Натиснути **Instances** в лівому меню.
3. Виберіть екземпляр EC2, який називається **aws-cloud9-IoTOnAWS**.
4. Натисніть **Actions > Instance State > Stop instance**