Добрый день, уважаемые студенты! Я рад вас видеть на очередной сессии разбора лабораторной работы. Тема лабораторной работы – AWS Lambda. Итак, начинаем. Для начала работы нам необходимо добраться до AWS Management Console. Для этого на основной странице лабораторной работы нажимаем на кнопку AWS.

После этого мы попадаем на основную страницу AWS Management Console. При этом можно увидеть, что вверху справа отображено “временный пользователь”. Теперь в строке поиска сервисов вводим Lambda и в выдаче вы увидите сервис, на который нам необходимо перейти. После перехода на основную страницу AWS Lambda, выйдет следующее сообщение об ошибке, но мы его игнорируем, т.к. на реальном AWS аккаунте подобного сообщения не будет. Далее необходимо в левом навигационном меню выбрать “Functions” и на основной странице нажать кнопку “Create Function”. В рамках нашей лабораторной работы мы создаем новую функцию, поэтому выбираем опцию Author From Scratch. Ниже вводим название функции, она будет называться myStopinator. Как runtime, т.е. среда исполнения, выбираем Python версии 3.8. Еще ниже находятся настройки роли для этой функции, мы будем использовать существующую роль, т.е. из списка выбираем myStopinatorRole. Это роль создана специально для нас, она доступна для Lambda, у нее есть права для остановки EC2 инстансов. После введения всех необходимых настроек мы нажимаем внизу справа на кнопку Create Function. Нас перенаправит на страницу с нашей функцией и появится соответствующее сообщение о том, что функция успешно создана.

Следующий наш шаг – создание триггер-ивента. Нажав кнопку “Add Trigger”, мы будем направлены на страницу, где необходимо выбрать источник. Источником, т.е. триггером, будет выступать Event Rule, следовательно, начинаем вводить event, и в выпадающем меню выбираем опцию EventBridge (CloudWatch Events). У нас появится возможность выбрать существующий ивент, либо создать новый. Выбираем опцию создать новый ивент (Create a new rule) и назовем его “everyMinute”. Как тип выберем Schedule Expression, т.е. повторяющиеся по расписанию события. В нашем случае, это будет rate (1 minute). Это нужно для того, чтобы этот ивент триггерил нашу лямбду каждую минуту. После введения всех настроек необходимо нажать на кнопку Add внизу чуть справа. Как только вы ее нажмете, вас перенаправит обратно на страницу с нашей функцией, там будет сообщение, что наш триггер успешно создан и у нас на странице появился один существующий триггер, EventBridge, который триггерит нашу лямбда-функцию.

Теперь необходимо изменить код нашей функции. Для этого прокрутим ниже и во вкладке “Code” откроем специальный редактор кода. Здесь выбираем lambda\_function.py и открываем код этого файла. Необходимо очистить все то, что там есть и вставить шаблон кода из нашей лабораторной работы. Здесь необходимо будет внести некоторые изменения в код, а именно заменить регион. В нашем случае, это North Virginia, поэтому вставляем системное название “us-east-1”. Далее необходимо вставить ID EC2 инстанса. Переходим на этот сервис, затем в левом навигационном меню переходим на вкладку Instances. Следует удостовериться, что вы находитесь в нужном регионе, а именно North Virginia. После этого вы увидите в списке инстансов один или два инстанса, нам нужен тот, который называется instance1. Обратите внимание на его статус, он должен быть в состоянии running. Далее выбираем этот инстанс, в дополнительной информации переходим во вкладку Details и копируем Instance ID. После того, как мы скопировали его, возвращаемся обратно к нашей лямбда-функции и на третьей строке вставляем значение Instance ID, который мы скопировали на предыдущей странице. После этого необходимо сохранить код. Это можно сделать, нажав комбинацию клавиш Ctrl+S, либо выбрать опцию Save в пункте меню File. Чтобы новый код лямбда-функции начал работать, необходимо его задеплоить путем нажатия кнопки Deploy и через несколько секунд появится сообщение о том, что код функции успешно обновлен.

Пройдемся вкратце по коду, который мы сохранили. В первой строке мы импортируем наш SDK. SDK для Python – это Boto3, поэтому здесь будет import boto3. Далее, вторая и третья строки это объявление переменных, а именно регион и информация о наших инстансах. В первом случае – это текстовый тип данных, а во втором случае – это массив из string-ов. Таким образом, если необходимо останавливать не один, а несколько инстансов, можно передавать дополнительные значения в этом массиве. Далее, в четвертой строке, мы инициируем клиент Boto3 для сервиса EC2, т.е. мы вызываем Boto3.client, указываем какой сервис нам нужен, в нашем случае это EC2, и дополнительно указываем название нашего региона. Все это мы присваиваем переменной EC2. Далее идет стандартная конструкция для Python кода внутри AWS, а именно lambda\_handler функция. На вход она принимает event и context. Это две переменные, которые заполняются системно, т. е. в момент, когда что-то вызывает нашу лямбда-функцию, то при передаче некоторых данных эти данные будут находиться в переменной event. Также есть некоторая мета информация по исполнению этого кода, вся эта информация хранится в переменной context. Внутри этой функции в седьмой строке мы используем наш клиент ec2.stop\_instances, мы вызываем этот action. Мы помним, что для совершения некоторых действий в AWS можно использовать три варианта. Первый – это использовать саму консоль, т.е. можем через консоль пойти и остановить самостоятельно. Во-вторых, мы можем воспользоваться AWS CLI, т.е. командная строка, в ней вызываем команду Stop Instances и этой командой мы останавливаем наши инстансы. И третий вариант – это использование SDK, это написание кода более высокого уровня, которое производит некоторые действия в облаке. В нашем случае мы останавливаем инстансы. Внутри в скобках передаются дополнительные параметры, в нашем случае параметр instanceIDs, который равен нашей переменной instances, т.е. мы передаем на вход список instanceID. Далее в восьмой строке мы делаем вывод сообщения с информацией о наших инстансах.

Это был самый простейший пример кода Лямбды и все, что вы будете писать в будущем, будет плюс-минус похоже на этот пример. Наиболее часто вы будете использовать SDK, это Boto3 в случае использования языка программирования Python, и на самом деле это не так сложно. Более того, мы только что подготовили код, который будет запускаться в облаке и при этом вам не пришлось настраивать отдельный сервер или настраивать что-либо дополнительно. Мы лишь загрузили наш код, задеплоили его, сохранили и теперь этот код будет запускаться каждую минуту.

Теперь перейдем на вкладку “Monitor”, где можно посмотреть, какая информация доступна касательно запусков этой лямбды. Мы остановимся только на трех графиках. Первый – это Invocations, количество запусков нашей лямбда-функции во времени. Здесь видно небольшую линию, это говорит о том, что наша лямбда запускалась два раза. Далее, второй график Duration показывает, какое количество миллисекунд запускалась наша функция, в нашем случае это 18 миллисекунд. Третий график – Error count and success rate, он показывает процентное соотношение удачных и неудачных совершений исполнения лямбда-кода. В нашем случае красный индикатор находится на уровне 0, т.е. все наши запуски прошли успешно, а зеленый на уровне 1, т.е. 100%. Отлично! Теперь вернемся на страницу с нашими инстансами. Мы убедились, что наш код уже несколько раз запустился, теперь попробуем обновить список наших инстансов. Нажав на иконку обновить, вы увидите, что состояние инстанса instance1 изменилось с Running на Stopped, т.е. наш код отработал и успешно остановил наш инстанс.

На этом наша лабораторная работа завершается. Я вам напоминаю, что нам необходимо правильно выйти со всех систем. В первую очередь нам необходимо выйти с AWS Management Console. Для этого вверху справа нажимаем на название вашего пользователя и в выпадающем меню кнопку “Sign Out”. Далее, уже на странице с лабораторной работой надо остановить лабораторную работу, чтобы наш AWS аккаунт был удален. Для этого нажимаем кнопку End Lab, подтверждаем, что мы завершаем лабораторную работу и дожидаемся сообщения “You may close this message box now”. После этого вы можете закрыть все окна и успешно выйти из этих систем.

На этом мы завершаем сессию разбора лабораторной работы. Надеюсь, вы узнали что-то новое и получили более полное представление о сервисе AWS Lambda. Спасибо, увидимся на следующих наших активностях!