Добрый день, уважаемые студенты! Я рад вас видеть. Мы с вами добрались до третьего модуля и сегодня поговорим про глобальную инфраструктуру AWS. Сегодняшняя лекция намного короче, чем предыдущие и состоит из двух частей. В первой части мы проговорим про глобальную инфраструктуру AWS, а во второй части поближе познакомимся с некоторыми группами сервисов и сервисами, которые в нее входят. Эти же сервисы вы можете встретить на реальном экзамене AWS Certified Cloud Practitioner.

Глобальная инфраструктура AWS, что же это такое? Это специально продуманная и построенная, гибкая, масштабируемая, безопасная и надежная среда для облачных вычислений, которая доступна для вас по всему земному шару в 22 различных регионах. Когда мы говорим про регионы, мы подразумеваем определенную географическую локацию. Каждый регион AWS – это изолированная часть глобальной инфраструктуры AWS, которая может работать самостоятельно. Внутри региона зачастую два или более Availability zone, так называемые зоны доступности. Когда мы говорим про зоны доступности, каждая из них может содержать несколько дата-центров. Таким образом обеспечивается высокая доступность каждого региона. Если у вас в рамках вашего бизнеса есть необходимость присутствовать в нескольких регионах, на нескольких локациях, то задачи по репликации, по переносу, либо копированию и клонированию ваших данных, вашей инфраструктуры на другой регион, производится вашими силами, является вашей ответственностью перед клиентами. Для всего этого есть уже разработанные инструменты, которые помогут упростить этот процесс. Также следует отметить, что помимо стандартных регионов, которые разбросаны по всему земному шару, существуют три отдельных региона. Первые два – это регионы, которые находятся в Китае. Они обособленны от основной части, от глобальной инфраструктуры, а также появления новых features, в целом развитие AWS в этом регионе на некоторое количество шагов отстает. Таким образом, все те новинки, которые появились в прошлом году, или в этом году, маловероятно, что вы сможете использовать в регионах, которые находятся в Китае. Поэтому если у вас бизнес такой, который требует от вас создание инфраструктуры в Китае, следует учитывать эти моменты. Другой регион, который также следует выделить, это AWS GovCloud, это специальный изолированный регион в США, предназначенный для правительственных проектов, т.е. все сайты, которые правительственные и которые хостятся в облаке, они хостятся именно в этом регионе.

У вас может возникнуть достаточно правильный вопрос: «У нас есть достаточно много регионов по всему земному шару и какой же выбрать?» Самый первый момент, который часто упоминается, это так называемые Data governance и Legal requirements, т.е. официальные документы, требования внутри вашей страны, которые регулируют построение IT-архитектур, обработку данных, обработку персональных данных, а также хранение, трансграничную передачу, и другие моменты, которые могут в каком-то смысле усложнить или создать дополнительные сложности при построении IT-инфраструктуры в облаке AWS. Поэтому, когда вы начинаете большой проект, обязательно посмотрите эти официальные документы в вашей стране и постарайтесь это учесть. Со стороны AWS тоже есть некоторые активности, которые помогают клиентам, т.е. нам. Они выпускают так называемые Whitepapers – это такой некий документ различного объёма в зависимости от темы. Он может быть посвящен на различные темы, цель которого дать некоторые рекомендации от самого AWS, о том, как правильно, рационально, безопасно, оптимально построить некоторые решения в облаке AWS. Какая-то часть этих whitepapers предназначена как раз для освещения этих моментов, вопросов по Data governance и Legal requirements для каждой отдельно взятой страны. Следует также проверить, этот список постоянно пополняется, обновляется и, если есть подобный документ для вашей страны, следует непременно с ним ознакомиться.

Следующим аспектом, который следует учитывать, это близость к вашим клиентам. Если клиенты находятся по большей части в определенном регионе, то наиболее рационально выбрать регион, ближайший к этой локации. Следует также учесть, что географически одинаковое расстояние от вашей локации до двух разных регионов не говорит о том, что скорость доставки контента, так называемое Latency, будет одинаковым. Поэтому для точного подбора нужно выбрать несколько ближайших регионов и провести несколько тестов, для того чтобы точно измерить среднее значение Latency между вами и вашими клиентами. Также здесь хотелось бы отметить, я не буду приводить примеры, это можно быстро найти одним запросом Гугла, есть специальные веб-сайты, которые позволяют делать это автоматически, т.е. вы нажимаете старт и по нажатию этой кнопки начинается ping в различные регионы AWS и сохранятся значения этого ping, и таким образом, через несколько этапов прохода этой проверки скорости вы получаете усредненное значение от вашего местоположения, где был открыт ваш сайт, т.е. с вашего локального интернет-провайдера до всех регионов AWS. Это может удобно.

Следующий момент, который надо учитывать, это доступность сервисов в определенном регионе, который вы хотите выбрать. Это я говорю к тому, что есть регион North Virginia, в котором выходят первыми все новинки, это считается основным регионом среди стандартных регионов и через какое-то время, как оно появилось там, это обновление перетекает во все остальные регионы с той или иной скоростью, и новый сервис либо новый функционал старого сервиса становится доступным в других регионах. Поэтому может получится так, что определенный сервис или определенный функционал сервиса критичен для вас, для вашего бизнеса, в таком случае надо обязательно проверить доступность этой функции либо сервиса в выбранном вами регионе. В целом, если нет абсолютной разницы с точки зрения Latency между всеми регионами до вас, это могут быть какие-то изолированные расчеты, которые не требуют близости к вашим клиентам, происходят внутри вашей инфраструктуры и на выходе отдается лишь какой-то вывод, который не важно с какой скоростью передается, главное доходит, то в этом случае рекомендуется выбирать главный регион North Virginia, где появляются все обновления. Есть некоторые нюансы, которые упрощают работу с облаком при выборе North Virginia, но это уже Advanced топики, которые даже не придут на AWS Certified Cloud Practitioner экзамене, и эти нюансы мы пройдем на следующих курсах.

Самый последний момент, который надо учитывать, это – цена, т.е. по всему земному шару, даже если все везде одинаковое, все стандартизировано, дата-центры одинаковые, то уровень жизни отличается в разных локациях, соответственно это отражается на цене сервисов. Какой-то сервис, который находится в определенном регионе, может стоить дороже или дешевле, чем в другом регионе. Еще один момент, который может повлиять на цену, это то, что старые регионы, которые давно работают, достаточно популярные, что позволяет им увеличивать количество дата-центров, расширяться, таким образом они выигрывают от масштабирования. Зачастую в таких крупных регионах цены ниже, по сравнению с регионами, которые новые или непопулярные. Поэтому этот момент следует учитывать, но в момент сравнения цен вы и так все это увидите.

Следующим вложением внутри каждого AWS региона, находятся Availability zones. Availability zone – тоже изолированная, самодостаточная сеть между несколькими дата-центрами, которая может работать самостоятельно и предоставляет высокую доступность всем сервисам, IT-ресурсам в облаке AWS. Следует запомнить некоторые физические расстояния между структурами AWS. Начнем с ближайших. Мы говорили ранее, что внутри Availability zone есть несколько дата-центров. Каждый дата-центр – это такое здание, которое может состоять из сотен тысяч серверов. Можете представить масштабы? Каждый дата-центр расположен от другого дата-центра в рамках одной Availability zone на десятки километров, а когда мы говорим про расстояния между Availability zones, то мы говорим о расстоянии до 100 км, и теперь следующий порядок, это несколько сотен км, это расстояние между регионами. Подобного рода вопросы приходят на экзаменах. От вас не будет требоваться знание определенных цифр, но вы должны запомнить порядок цифр. Это десятки км между дата-центрами, если до 100 км – это расстояние между Availability zones, если больше 100 км – это расстояние между регионами. Обычно это несколько сотен км. Когда мы говорим про Naming, т.е. наименование, у каждого региона есть свое кодовое название. Как пример, возьмем первый западноевропейский регион eu-west-1, вы видите его на слайде, и naming availability zone в рамках этого региона отличается тем, что в конце добавляется буква, начиная с a, b, c, d и т.д. Здесь есть одна определенная фишка, т.е. из разряда «все гениальное – просто». Все, скажем так, клиенты, мы и вы, подсознательно стараемся выбирать и создавать наши ресурсы в первой availability zone, т.е. «а», стремление к тому, чтобы быть первыми. Для того чтобы распределить равномерно между несколькими availability zones, так чтобы каждый мог выбрать «а», был сделан некоторый трюк. Вы можете сейчас остановить видео и подумать, попробуйте угадать. Сообщите куратору. Я надеюсь, что кто-то угадал, как это было решено. Давайте расскажу. На самом деле, все гениальное – просто. Как это делается? Когда вы регистрируете AWS аккаунт, представим, что в одном регионе есть три availability zones. Назовем их первая availability zone «а» в первом городе, вторая availability zone «b» во втором городе, третья availability zone «с» в третьем городе. При регистрации, например, я зарегистрировался и случайным образом перемешиваются названия availability zones с реальными availability zones. Например, первая availability zone «а» может быть привязана к первому городу, вторая availability zone «b» ко второму городу, третья availability zone «с» к третьему городу. Когда регистрируется кто-то другой, этот порядок случайным образом меняется и таким образом для него первая availability zone «а» фактически будет означать хостинг в третьем городе. Вторым пусть будет первый город, и третьим будет второй город. Таким образом, все выбирают и могут выбрать availability zone «а», но фактически рандомно на фоне, они были распределены равномерно и нагрузка на все availability zones примерно одинакова. Надеюсь, кто-то угадал, если угадали, молодцы. Идем дальше.

Мы с вами уже начали говорить про дата-центры. Еще пару моментов, которые хотелось бы добавить, это то, что задача AWS как облачного провайдера касательно дата-центров – это предоставить полную физическую безопасность, и чтобы легче это обеспечить, не разглашается даже расположение дата-центров. В лучшем случае вы можете знать страну расположения дата-центров и таким образом, это помогает лучше защитить дата-центры от физического обнаружения и попыток проникновения. Другой момент, это то, что используются самые новейшие технологии, самые новейшие оборудования и железо. Оно специально изначально разрабатывается избыточным, это так называемый Redundant дизайн. Что такое избыточная архитектура? Представим, что вам надо три сервера для корректной работы, а вы делаете и настраиваете пять серверов. Таким образом, вы гарантируете, что, если что-то ломается, один из серверов, это абсолютно никак не повлияет на работу того, что обрабатывают эти сервера. Вы безопасно отключаете одну, для корректной работы вам достаточно три, а там еще четыре. Таким образом, работа не останавливается, и в момент, когда находится по мониторингу, что вот это сервер вышел из строя, он выводится из балансирования трафика и заменяется новым. Касательно железа, следует упомянуть, что закупка оборудования происходит, как правило, от оригинальных производителей. Зачастую те продукты, то сетевое оборудование, будь то другое железо, мы покупаем уже с определенным брендом, мы переплачиваем за бренд. В случае с AWS, у них есть прямой выход на производителей с несколькими альтернативами одного и того же физического оборудования, и они могут максимально выгодно приобрести нужное и правильное железо, для того чтобы обеспечивать нам высокую доступность.

Мы с вами проговорили, что такое дата-центры, и теперь давайте поговорим о таком понятии, как Points of Presence, т.е. точки присутствия. Другими словами, это та технология, позволяющая на ближайших к пользователям серверах кэшировать контент, который был запрошен и часто запрашивается другими пользователями, для того чтобы сократить расстояние до клиентов и уменьшить время, когда пользователи добираются до контента, будь то медиаконтент, текстовая информация, веб-сайты и т.д. В случае, когда мы говорим про AWS, у него есть собственный CDN (Content Delivery Network), это сеть серверов по доставке контента, называется Amazon CloudFront, она распределена по всему миру и используется для сокращения Latency. Есть также сервис Amazon Route 53, это сервис для регистрации доменных имен, т.е. вы хотите купить доменное имя, вы можете это сделать в одном из сервисов AWS – Amazon Route 53. Далее, все сервисы, так или иначе работающие с доставкой контента, связаны с Amazon CloudFront, что такое Amazon CloudFront? Это сеть из серверов, которые называются Edge Location – эти сервера расположены к определенной группе пользователей, чтобы обеспечивать им быстрый доступ к данным. По всему миру, на момент записи видео, они расположены в 69 крупных городах в 30 странах по всему миру. Этим обеспечивается высокоскоростной и бесперебойный доступ к данным.

Помимо Edge Locations, существует так называемый Regional Edge Cashes. Отличие в том, что Edge Loсations ближе к пользователям и хранит самые часто запрашиваемые объекты, файлы и т.д. Когда мы говорим про Regional Edge Cashes, он агрегирует данные из нескольких Edge Locations, и находится, скажем так, чуть «дальше от клиентов» и там хранятся данные, которые менее популярные и не поместились в Edge Locations.

Мы уже достаточно проговорили про глобальную инфраструктуру AWS, о ее компонентах и основная идея в том, что благодаря этой инфраструктуре вы можете построить свою IT-инфраструктуру в облаке, которая достаточно гибкая, легко масштабируется, отказоустойчивая и высокодоступная. Давайте пройдемся по основным моментам, пройденным в первой части нашей сессии. Мы узнали, что такое Availibality zones, что такое регионы, что такое дата-центры. Дальше мы перешли к CDN, т.е. Content Delivery Network, Edge Locations, Regional Edge Cashes, и в целом это вся необходимая информация касательно глобальной инфраструктуры AWS для успешной сдачи экзамена AWS Certified Cloud Practitioner.

Мы с вами переходим ко второй, последней части нашей сегодняшней сессии и сделаем некий обзор основных групп сервисов и сервисов внутри этих групп, которые мы будем проходить на нашем курсе. Как уже упоминалось на предыдущей лекции, глобальная структура AWS состоит из нескольких уровней. Самый базовый уровень – это инфраструктура, в которой есть наши регионы, Availability zones, Edge Locations под которыми сидят наши дата-центры. Если идем выше, абстрагируемся, есть набор Foundational сервисов, основополагающие сервисы, а именно Compute – вычислительные мощности, это сервис Amazon EC2; далее Networking – все, что связано с сетями, начиная от сервиса AWS VPC; и Storage – это хранение, основные сервисы Amazon S3, Amazon EBS (Elastic Block Storage) и т.д. Следующий уровень – Platform services, это все те сервисы, которые базируются на основных сервисах, и предоставляют свой сервис по модели либо Platform as a Service, либо Software as a Service. Все то, что поверх, уже последний уровень, это наши приложения, которые пишут наши разработчики, после чего они доходят до наших клиентов.

На этом слайде перечислены 23 различные категории сервисов внутри AWS, этот список постоянно меняется и дополняется. На нашем курсе мы рассмотрим следующие 7, которые могут потенциально прийти на реальном экзамене AWS. Поэтому внутри каждой группы мы рассмотрим основные сервисы более подробно, чтобы полноценно подготовиться к экзамену.

На следующих слайдах мы подробно остановимся на сервисах внутри каждой из 7 групп сервисов и вкратце расскажем для чего нужен этот сервис. На следующих слайдах мы подробно остановимся на каждом из сервисов, но здесь некий широкий обзор того, что мы будем делать, что к чему относится, и если вы конспектируете, самое время записывать.

Первая группа – это группа сервисов хранения. В рамках нашего курса и для экзамена AWS Certified Cloud Practitioner нам необходимо знать 4 сервиса: Amazon Simple Storage Service (Amazon S3) – сервис объектного хранения данных. Следующий сервис – Amazon Elastic Block Store (Amazon EBS), этот сервис используется в связке с нашими виртуальными серверами Amazon EC2. Почему, объясняю. Я специально перепрыгнул на второй сервис, чтобы дать описание каждой и сейчас я приведу сравнение и описание каждого из типа хранения. Когда мы храним файлы в виде объектов, объектно, то файл воспринимается как единое целое и вне зависимости от его размера оно не должно меняться, т.е. мы его загрузили, и в тот момент, когда в нем что-то меняется, неважно, большая часть файла или один байт, этот файл уже другой. Он обновлен, и для того, чтобы обновить файл в объектном хранилище, его надо загрузить заново, весь файл целиком. А когда мы говорим про Amazon EBS – это блочное хранение. То же самое, что жесткие диски на вашем компьютере или ноутбуке. Когда мы храним файлы или данные блоками. Представим большой файл, для того чтобы хранить такой большой файл, вы занимаете десятки или даже сотни тысяч блоков для хранения этого файла. Таким образом, благодаря этому блочному хранению, мы имеем возможность при малейшем изменении этого большого файла, достаточно найти и обновить этот блок и файл может быть доступен в новой версии. Теперь вопрос. Я вам рекомендую остановить видео и подумать. В каком из видов хранения, в объектном или блочном, обычно устанавливается операционная система и почему? Надеюсь, мы услышали много интересных и правильных ответов. На самом деле ответ был дан еще до того, как был задан. Операционная система устанавливается на блочном типе хранения. Почему? Когда мы работаем с операционной системой, она может состоять из миллионов маленьких файлов, и намного проще сохранить эти файлы в одном блоке, чем вести учет каждого маленького файла в объектном типе хранилища. Таким образом, хранение занимает меньше места и учет как таковой этих маленьких файлов нам не нужен. Другой момент, это то, что у нас операционная система, это некий живой организм, постоянно меняющийся, постоянно какие-то файлы обновляются, и говоря про объектный тип, у нас есть разные файлы внутри нашей операционной системы. Если что-то обновилось, чтобы достучаться до новой версии, нужно весь этот файл загрузить в объектное хранилище. А если размер файла большой, а обновилась лишь малая часть, то мы тратим фактически больше времени, чтобы загрузить его туда, поэтому нерационально устанавливать операционную систему в объектное хранилище, а правильнее в блочное.

Идем дальше. Следующий сервис – это сервис Amazon EFS (Amazon Elastic File System) это также блочное хранилище, файловая система. В отличие от Amazon EBS, она может быть разделена, расшарена между несколькими Amazon EC2 инстансами. Amazon EFS не может работать как операционная система, чтобы установить операционную систему и ваш сервер запустился, вам надо привязать к нему Amazon EBS, после запуска, монтируется другой диск, который может быть Amazon EFS. Таким образом, можно будет, например, если есть задача в проведении каких-то расчетов и необходимо место для обмена данными между несколькими инстансами, то Amazon EFS может быть достаточно хорошим и правильным вариантом.

Четвертый сервис, о котором мы поговорим – это сервис в сервисе, т.е. у нас есть Amazon S3, внутри есть другой сервис для объектного хранения – Amazon S3 Glacier, возле него еще один сервис – Amazon S3 Glacier Archive. В чем отличие и идея? Говоря об Amazon S3, он состоит из нескольких вариантов хранения, о них мы поговорим на следующих сервисах, и глобальное отличие S3 классов хранения от классов Glacier и Glacier Archive, в том, что файлы, хранящиеся объектно, записываются один раз и очень редко, маловероятно запрашиваются обратно. Зная поведение ваших данных, например, есть данные, которые записаны один раз и они не используются, но их надо хранить по определенным нормативным документам. К примеру, все финансовые операции в банках, их логи, должны храниться 5 лет. После 5 лет вся эта информация о транзакциях может быть удалена. Подобного рода данные достаточно часто встречаются, поэтому был создан специальный сервис для хранения подобных данных. Их можно загрузить туда и хранить очень дешево, зная, что мы их никогда не запросим. Отличие Glacier от Glacier Archive в том, что Glacier Archive еще дешевле. Там хранятся только те данные, про которые мы с 99% уверенностью знаем, что мы их не запросим. Если мы их запросим, то оплата за запрошенные данные может достаточно дорогой. В момент, когда вы выбираете класс хранения, будь то Glacier или любой из классов S3, следует учитывать стоимость и особенности каждого из классов хранения.

Переходим к следующей группе сервисов – группа сервисов Compute (вычисления). Здесь будет рассмотрено большее количество сервисов. Пройдемся вкратце по каждому из них. Самый первый и основной – это Amazon EC2 (Elastic Compute Cloud), это сервис, я буду стараться говорить простым языком, чтобы вы уловили основной смысл и при необходимости смогли найти сложные описания, сложные дефиниции самостоятельно в документации. Работать с документацией вы уже умеете. Amazon Elastic Compute Cloud – это когда вам нужно запросить виртуальный сервер, чистый виртуальный сервер, с которым вы дальше можете что-то настраивать, что-то делать.

Далее, это Amazon EC2 Auto Scaling, это сервис, настраиваемый поверх Amazon EC2, ранее я рассказывал, что есть такое понятие, как автомасштабирование, и можно настроить инфраструктуру таким образом, что в зависимости от нагрузки она увеличивается или уменьшается, некий абстрактный живой организм. Одним из сервисов, реализующих это, является Amazon EC2 Auto Scaling. Вы создаете некоторую группу, к ней привязываете Amazon EC2 инстансы, передаете различные параметры, и далее, в зависимости от определенных условий, эти сервера автоматически отключаются либо дополняются при необходимости. Условия могут быть разными, в т.ч. и показатели нагрузки на ваши сервера.

Следующий сервис – это Amazon Elastic Container Service (Amazon ECS). Это сервис контейнерной оркестрации, т.е. используем этот сервис, только когда нам надо работать с докер-контейнерами, и связанный сервис Amazon ECR (Elastic Container Registry), это когда нужно найти некий образ докер-контейнера, и как раз то место, где все хранится или можно сохранить, это как раз Amazon ECR.

Далее, это сервис AWS Elastic Beanstalk, он достаточно активно продвигается самим AWS, но так как есть ряд других сервисов, предоставляющих больше контроля и свободы, AWS Elastic Beanstalk на текущий момент не является самым популярным при создании приложений. Но идея такая, что если у вас есть патчи либо Microsoft IIS приложения, вы пишете приложения, оформляете код, архивируете код, дополняете конфигурационными файлами для Beanstalk, в которых расписываете необходимые для вас параметры и этот архив передаете в сервис AWS Elastic Beanstalk. На этом все, после этого у вас автоматически это приложение, код разархивириуется, компилируется и deployed, т.е. устанавливается на Production Instance, Production Environment среду, в котором она сразу становится доступной. Это некий сервис, облегчающий жизнь разработчикам, которые не очень сильны в системном администрировании, и этот сервис будет очень кстати.

Следующий мой любимый сервис, это AWS Lambda, представитель модели FaaS, это следующая модель предоставления сервисов, Function as a service. В чем идея? Представим, что вы разработчик, знаете один язык программирования Python, но вы хотите что-то большее сделать, какой-то стартап. В этом случае нужно знать что-то еще из системного администрирования. Иногда это напрягает, кто-то боится, кто-то не хочет, кому-то лень это изучать. В случае с AWS Lambda все просто, вы пишете код, он деплоится внутри этой функции и все! Магия происходит! В момент, когда функция нужна, она вызывается и все то, что находится под этим, полностью обслуживание вот этого выполнения кода, обходит вас стороной. Таким образом, обновление версий Python, обновления операционной системы, патчи безопасности, администрирование самим виртуальным сервером, это все обходит вас стороной. Вы пишете код и оплачиваете только за гигабайт-секунды. Это такая мера измерения, помогающая считать отработку AWS Lambda, чтобы как-то оценивать и за что оплачивать. Гигабайт-секунды, идея в том, что AWS Lambda работает какое-то количество времени, максимум 15 мин. Сама AWS Lambda запускается в маленькой виртуальной среде с определенным количеством операционной памяти. Например, выделяете для нее 512 МБ, если она запускается за 2 сек, если перемножить, то получится 1 ГБ\сек потратила эта Lambda, для исполнения определенного кода. За 1 ГБ\сек, в зависимости от цен со страницы Pricing, будет оплата. AWS Lambda – одна из самых основных сервисов подхода Serverless, т.е. безсерверной архитектуры. Про это мы будем очень много говорить, когда будет возможность, я буду рассказывать интересные моменты, что-то из своего опыта работы. Мы к этому вернемся еще не раз.

Следующий сервис – Amazon Elastic Kubernetes Service (Amazon EKS). Ранее мы говорили про Amazon ECS для работы с докер-образами. Докер-образы – это отдельная технология, а умение совладать с этим количеством докер-образов и правильно их оркестрировать, это другая технология, называется Kubernetes, или K8S. AWS для Kubernetes создал отельный специальный сервис Amazon EKS, для предоставления Kubernetes as a service. Вам не нужно его настраивать, разбираться в его администрировании, поднимать и выделять сервер. Вы просто выводите определенные параметры для работы вашего Kubernetes, указываете пути до ваших докер-образов, дополнительные параметры работы. Она запускается и производится оплата согласно прайсингу за этот сервис. Фактически, вы избавляетесь от большой административной части работы. Естественно, насколько это возможно сделана тюнинг-оптимизация, подходящая для большинства задач. Если у вас такая задача, это идеальный вариант. Если же у вас какая-то специфичная задача, нужна супероптимизация, какие-то тонкие настройки, в этом случае, возможно, правильным вариантом будет самостоятельно хостить Kubernetes на виртуальных серверах, но это требует дополнительных умений, знаний, специалиста как минимум и может обойтись дороже, чем платить за этот сервис через Pay-as-you-go-pricing, когда вы платите столько, сколько вы использовали.

Последнее, это Amazon Fargate – вычислительный движок для Amazon ECS, помогающий вам запускать и управлять контейнерами, это вроде аналога Kubernetes, но без Kubernetes, сервис оркестрации контейнеров. Он также предоставляется по модели Software as a service, т.е. готовый, достаточно ввести определенные настройки и вы запускаетесь. На этом мы закончили с группой compute-сервисов. Идем дальше.

Следующая группа сервисов – это сервисы, связанные с базами данных. Так как у меня есть некоторый опыт работы как разработчика баз данных, я надеюсь, что смогу параллельно со знаниями AWS, также передать структурированные знания по базам данных, которые помогут вам лучше понимать вам сервисы AWS именно по направлению баз данных. Это поставит вас на уровень выше по сравнению с другими студентами и другими кандидатами при устройстве на работу. Надеюсь, пригодится.

Первый сервис – Amazon Relational Database Service, реляционная база данных, которая предоставляется как сервис. Один из самых популярных сервисов, в котором можно выбрать из достаточно немаленького списка существующих движков систем управления базами данных, таких, как Postgres, MySQL, Oracle, Microsoft Server и др.

Далее идет Amazon Aurora. Amazon RDS и Amazon Aurora всегда идут рука об руку, потому что Amazon Aurora точно такой же сервис, как и Amazon RDS, но отличается тем, что специальная команда AWS взяла два самых популярных open source СУБД, это MySQL и Postgres и оптимизировала его настолько, внедрила и интегрировала настолько, что за те же деньги оно дает намного больше функционала и возможностей, нежели поставить чистый MySQL или Postgres через Amazon RDS. Но и здесь есть «ложка дегтя в бочке меда». Amazon Aurora – чисто AWS решение, поэтому, как только вы начинаете использовать ее, с учетом того, что MySQL и Postgres достаточно сильно модифицированы, где-то может синтаксис сильно отличаться при написании SQL-запросов. Это говорит о том, что вы доходите до Vendor lock-in, это когда вы привязываетесь к одному производителю, к какому-то приложению или программному обеспечению. В этом плане все сильно зависит от ваших запросов и возможностей, а также планов на будущее, будете ли вы дальше двигаться с AWS или потенциально может произойти что-то, что вам нужно будет перепрыгнуть от одного облачного провайдера к другому. Если такие риски есть, то нужно продолжать с Amazon RDS, потому как вы привяжетесь в начале, все приложение будет интегрировано с Amazon Aurora, так как даже подобного функционала, как у Amazon Aurora, не будет у другого облачного провайдера, даже если у него будет своя Amazon Aurora, скажем так, то она будет очень сильно отличаться. Даже простой перенос от Amazon Aurora с движком MySQL на другой чистый, не привязанный к Aurora MySQL, потребует значительных усилий и затрат ресурсов и времени. Оно может того не стоить. Поэтому надо семь раз подумать, в тот момент, когда вы решите продолжать с AWS до конца по тем или иным причинам, можно выбрать Amazon Aurora и двигаться с ней и наслаждаться тем, что она дает. Если нет, то понимать и знать, что вы в любой момент можете перепрыгнуть в тот момент, когда условия работы с AWS будет не устраивать, миграция это никогда не легко, но относительно легче и менее болезненно перенестись от Amazon RDS с чистым MySQL, Postgres либо любым другим СУБД на сервис другого аналогичного облачного провайдера.

Далее, идет Amazon Redshift. Говоря про первые два сервиса, Amazon RDS и Amazon Aurora, имеем в виду реляционные базы данных транзакционного типа. Это значит, мы работаем транзакциями, пишем много, но можем мало читать. Мы делаем CRUD-операции, аббревиатура составлена из первых букв названий основных операций в СУБД: C – create, R – read, U – update, D – delete. Это фактически описание жизненного цикла данных в базе данных. В CRUD-операциях, имеется в виду каждая из этих операций легковесная, и делает что-то одно простое: записал, считал, обновил, удалил. Для работы приложения достаточно транзакционных типов отношений с базами данных, когда вам нужно, к примеру, список пользователей, вы делаете простой селект, который считывает нужные данные из нужной таблицы. Может будет сложный запрос, где присутствуют несколько таблиц, но не более. Здесь упор делается на скорость. Когда говорим про Amazon Redshift, это нетранзакционная, это аналитическая база данных, ее отличие в том, что вы не пишите туда транзакции, вы пишете один раз и много раз будете считывать. Много считывать потому, что она аналитическая и подразумевается, что данные крутятся по-разному и вы пытаетесь получить некий вид этих данных, чтобы извлечь полезную информацию для вашего бизнеса и принять какие-то решения. Еще одно отличие, когда мы работаем с Amazon RDS или Amazon Aurora, обычно эти базы так используют приложения, а приложениями пользуются пользователи. Когда мы говорим про Amazon Redshift, обычно загружается большой объем информации, данные готовятся заранее в нужном виде, готовятся сложнейшие запросы, чтобы крутить данные по-разному. Вызываются вручную обычно высшим менеджментом, не обязательно технический специалист или IT-директор, это может быть CEO или еще кто-то, который нажимает на кнопку и получает сложный запрос, который для него приготовили и он видит выжимку всего того большого объема данных, он может достигать тера- и петабайтов, и на основе этой выжимки принимает какое-то важное бизнес-решение.

Идем дальше. Четвертый сервис, который важно знать – это Amazon DynamoDB. Когда мы говорим про Amazon RDS или Amazon Aurora, имеем в виду SQL-базы данных. В последние годы стали уже популярны NoSQL базы данных и Amazon DynamoDB – один из ярких представителей NoSQL базы данных. Чем они отличаются и историю борьбы этих двух типов баз данных я расскажу чуть позже. Сейчас вам важно запомнить, что для некоторых приложений нет необходимости содержать достаточно сложную структуру из таблиц внутри реляционной базы данных, они должны быть между собой связанными, целостными и есть определенные ограничения при работе с реляционными базами данных. Работая с Amazon DynamoDB, это не связанные между собой, не структурированный набор записанных данных, и в нужном месте, зная, как их вытащить, вы запрашиваете эти данные для целей приложения, чтобы выдать дальше пользователю. Так вот, ввиду некоторых особенностей популярных в последние годы приложений, Amazon DynamoDB тоже является достаточно интересным решением и может дать некоторые преимущества. Но это опять же зависит от ваших бизнес-требований. Здесь следует отметить, что Amazon DynamoDB также является serverless-приложением. Вам не нужно ничего поддерживать, он дается как сервис. Оно часто используется в связке с AWS Lambda, Amazon DynamoDB и ряд других сервисов, которые мы рассмотрим на следующих слайдах.

Идем дальше, следующая группа сервисов – это сеть и доставка контента. Долго останавливаться не будем, потому что с основными сервисами мы уже пересекались на предыдущих лекциях. Вкратце пройдемся. Первая – Amazon VPC, идея в том, что допустим в вашей корпоративной сети или сети университета, есть развернутая локальная сеть. Через нее проходят разные перемещения. Например, можно скачать общую папку, где лежат электронные версии книг и учебников, может организованы дистрибутивы различных приложений для студентов на бесплатной основе и т.д. вы подключаетесь и работаете с ними через вашу локальную сеть. Бывает так, что в целях безопасности, в университетах во время, скажем, рубежного контроля или экзаменов, отключают выход в интернет, но остается доступ к вашему внутреннему университетскому сайту для сдачи экзаменов, увидеть оценки и т.д. Ваш сайт, если хостится локально, т.к. вы сидите в одной локальной сети, вы напрямую можно обращаться к нему. Если он хостится отдельно, то ваш университетский сайт заносится в так называемый «белый список» и, соответственно, вы можете загрузить его, но не другие сайты.

Следом идет Elastic Load Balancing, это второй из сервисов, помогающих «оживить» нашу инфраструктуру и стать автомасштабируемым, наряду с Amazon EC2 Auto Scaling. Оба используются с Amazon EC2, т.е. у нас поднимается виртуальный сервер; далее идет Auto Scaling, который в нужный момент увеличивает или уменьшает количество серверов в группе; и следом Elastic Load Balancing, балансирующий (распределяющий) нагрузку на группу.

Сервис Amazon CloudFront – мы уже проговаривали о нем, Content Delivery Network, сети Amazon для доставки контента.

AWS Transit Gateway, здесь я перескочу немного вперед, AWS Direct Connect и AWS VPN, эти сервисы помогают вам ваш локальный дата-центр связать с вашей облачной инфраструктурой. Они отличаются своими особенностями, плюсами\минусами и в зависимости от ваших бизнес-требований вы можете выбрать тот или иной вариант. Следует упомянуть, что может быть несколько подходящих вариантов, поэтому намеренно выбираются два наиболее подходящих из них, чтобы был резервный канал связи между локальным дата-центром, либо локальным оффлайн офисом до облачной инфраструктуры. Это часто практикуется.

Последний сервис – это Amazon Route 53, мы про него упоминали, это domain name system (DNS), через него можно покупать или арендовать домены.

Следующая группа сервисов – сервисы, связанные с безопасностью. Большинство из них мы уже проходили. Вкратце напомню. AWS IAM и AWS Organizations вам знакомы. Amazon Cognito – сервис, позволяющий вам регистрироваться или логиниться вашим пользователям внутри вашего приложения или веб-сайта нативно в AWS. Хотел бы отметить, что Amazon Cognito, также, как и Amazon Aurora, полностью сервис AWS, подобного вы не найдете больше нигде, это Vendor lock. Поэтому для ведения учета ваших пользователей существуют различные приложения, поддерживающие не только AWS, но и другие облачные провайдеры и в случае, если вы переживаете за Vendor lock, следует также рассмотреть те варианты.

Следующая – это AWS Artifact. Этот сервис является больше сборником документов, доступных в режиме чтения. AWS проходит различные проверки от регулирующих органов на получение сертификатов соответствия, сертификатов безопасности на соответствие требованиям. И вам как компании, которая хостит свою инфраструктуру в облаке AWS, также могут пригодиться некоторые подтверждающие документы по запросу госорганов в вашей стране. Если вам придет такой запрос, вы знаете куда обратиться для получения всех необходимых документов.

Следующий очень важный сервис – AWS Key Management Service, очень важный сервис, кому-то этот сервис может показаться сложным для понимания. Но как только вы его поймете, вы освоите одну из самых сложных тем в AWS, с этим сервисом мы еще столкнёмся. Он позволяет нам создавать и управлять ключами шифрования. Эти ключи позволяют шифровать весь поток данных, как в процессе хранения, так и в процессе передачи. Оно интегрировано с большим количеством сервисов AWS и является одним из основных наряду с сервисом AWS IAM.

Последний сервис, связанный с безопасностью, это AWS Shield, это SaaS сервис, т.е. managed сервис. Отличие от обычных сервисов в том, что они предоставляются по модели SaaS, который позволяет максимально сконцентрироваться на основном продукте, а этот сервис управляется самим AWS. Вам достаточно передать определенные настройки управления. AWS Shield помогает в защите от DDoS-атак. Что это такое, вы можете узнать одним кликом в интернете.

Предпоследняя группа сервисов, о них мы уже проговаривали, поэтому остановимся на них максимально кратко. Это сервисы, управляющие вашими затратами в облаке – AWS Cost and Usage Report, AWS Budgets и AWS Cost Explorer. Подробно о них мы с вами говорили на предыдущей лекции, поэтому двигаемся дальше.

Последняя группа сервисов – AWS management and governance, т.е. сервисы управления, облегчающие работу в облаке. Первый – это AWS Management Console, в какой-то мере можно назвать тоже сервис, дает возможность добраться до других сервисов. Про него будет отдельное Демо, там мы его рассмотрим подробнее. AWS Config – у вас есть IT-инфраструктура и в ней много разных ресурсов, каждый из них может так или иначе меняться, и AWS Config позволяет отслеживать эти изменения и реагировать на них. Amazon CloudWatch – важный сервис, там хранятся все логи ваших приложений. AWS Auto Scaling мы проходили уже, также, как и AWS Command Line Interface. AWS Trusted Advisor помогает вам оптимизировать безопасность и так называемый «performance», т.е. производительность вашей IT-инфраструктуры. Схожий с ним сервис, это AWS Well-Architected Tool, про его теоретические моменты мы еще проговорим подробно. В целом это некоторый набор рекомендаций, помогающий лучше выстроить архитектуру в вашем облаке. Последний, но немаловажный сервис, это AWS CloudTrail. Когда ваше приложение пишет логи, оно не изменяет никакие IT-ресурсы, допустим, у вас поднят виртуальный сервер, там есть приложение, когда пользователь отправляет ему запрос, происходит изменение в данных, связанных с этим пользователем, происходят какие-то изменения в самом приложении, он может по-другому отображаться, но сам виртуальный сервер никуда не уходит, и изменения в инфраструктуре не происходят. Так вот, есть отдельный вид операций, это именно User Activity и API usage. API имеется в виду, ранее говорилось, что все сервисы AWS, а это, наверное, самый большой в мире набор API (Application Programming Interface) т.е. язык, на котором сервисы (машины) общаются между собой по определенному протоколу. При запросе одних API ничего не меняется, но при запросе других меняется инфраструктура. Приведу пример: при создании новых инстансов вы сделали API-запрос, неважно как, через AWS SDK, AWS CLI или AWS Management Console, все это запускает API-вызов в отношении AWS и поднимается ваш сервер; или же поднимаете инстанс базы данных и т.д. Эти API-действия привязаны к пользователю или к роли внутри AWS, который это запросил, сохраняются в AWS CloudTrial. Его можно сравнить с «черным ящиком» в самолетах, в котором непрерывно записываются все данные, показания приборов, параметры и т.д. в момент полета. Если в облаке происходит какая-то ситуация или попытка несанкционированного проникновения, через AWS IAM отключаются все доступы и идет обращение в AWS CloudTrial для выяснения причин произошедшего в рамках вашей инфраструктуры.

На этом мы заканчиваем сегодняшнюю сессию и подведем небольшой итог. В первой части мы рассмотрели регионы, availibality zones, edge locations, т.е. глобальную инфраструктуру AWS. Вторая часть чуть больше, там рассмотрены категории и виды сервисов, их определения, для чего они существуют. Они могут прийти в вопросах на реальном AWS экзамене. На этом слайде представлены ссылки, где можно получить более подробную информацию по данной теме. На этом мы заканчиваем, всем большое спасибо за внимание и до встречи на следующем занятии!