WEBVTT

00:00:00.000 --> 00:00:04.600

Добрый день, уважаемые студенты! Я рад вас всех видеть на очередной лекции.

00:00:04.600 --> 00:00:07.400

Тема сегодняшней лекции - Connecting Networks.

00:00:07.400 --> 00:00:14.920

Ранее мы с вами говорили о том, из чего состоит VPC, какие у нее есть компоненты.

00:00:14.920 --> 00:00:31.000

Теперь же мы поднимаемся на уровень выше и посмотрим, как мы можем соединять между собой сети в облаке AWS, а также сеть в облаке AWS с нашей локальной сетью, которая не находится в облаке.

00:00:31.000 --> 00:00:33.000

Итак, давайте начнем.

00:00:33.000 --> 00:00:38.000

В рамках нашей сегодняшней лекции мы рассмотрим 6 тем.

00:00:38.000 --> 00:00:43.200

Первая тема касается наших лабораторных работ и посмотрим, какие компоненты добавятся.

00:00:43.200 --> 00:00:51.000

Далее, во второй и третьей части мы рассмотрим вопросы, как связать локальную инфраструктуру с облаком AWS.

00:00:51.000 --> 00:01:01.600

Далее, в четвертой и пятой части мы рассмотрим, каким образом мы можем соединять VPC между собой в облаке AWS.

00:01:01.600 --> 00:01:07.400

И в самом конце рассмотрим дополнительные сервисы, которые нативно интегрируются с VPC.

00:01:07.400 --> 00:01:17.200

Мы с вами начинаем первую часть нашей сегодняшней лекции и здесь рассмотрим, какие задачи предстоят в рамках лабораторных работ.

00:01:17.200 --> 00:01:29.200

Согласно бизнес-кейсу, кафе запустила акцию. В рамках акции, если посетитель покупает 10 и более одинаковых позиций, то на подарок дается некоторый десерт.

00:01:29.200 --> 00:01:36.000

Но для того, чтобы отслеживать прогресс посетителей, необходимо сохранять некоторую персональную информацию.

00:01:36.000 --> 00:01:39.400

Это email-адрес и номер кредитной карты.

00:01:39.400 --> 00:01:54.400

Кафе не может хранить эту информацию в облаке AWS, ввиду некоторых ограничений, связанных с хранением данных в конкретной стране, где находится София и другие главные герои.

00:01:54.400 --> 00:02:10.400

Поэтому было решено создать локально базу данных, которая хранит данные пользователей, но при этом транзакционные данные и вся инфраструктура остается в облаке AWS.

00:02:10.400 --> 00:02:23.400

В тот момент, когда нужны эти данные, из облака AWS приходит прямой запрос на локальную инфраструктуру, а именно в нужную нам базу данных и подтягивает необходимые данные.

00:02:23.400 --> 00:02:28.400

Другой момент связан с VPC.

00:02:28.400 --> 00:02:36.400

Мы с вами помним, что у нас есть Dev среда, а также Prod среда. И тот и другой находится в одном VPC.

00:02:36.400 --> 00:02:49.400

София, посоветовавшись с Оливией, решает, что необходимо их разделить между собой. То есть Dev среда находится в отдельной изолированной VPC, а Prod среда находится в другой VPC.

00:02:49.400 --> 00:03:02.400

Но, тем не менее, Оливия дополнительно советует Софии связать их между собой, так чтобы из Dev среды можно было обратиться к ресурсам на стороне Prod environment.

00:03:02.400 --> 00:03:19.400

Как вы видите, в рамках нашей лабораторной работы будем работать с несколькими VPC в облаке AWS, а также выполним задачу, связанную с соединением VPC в облаке AWS и локальной инфраструктурой.

00:03:19.400 --> 00:03:30.400

Мы с вами добрались до второй части нашей сегодняшней лекции. И подробнее здесь остановимся на сервисе AWS Site-to-Site VPN.

00:03:30.400 --> 00:03:44.400

Идея этого сервиса в том, чтобы через интернет прокинуть VPN канал, используя протокол IPSec, и подключить VPC в облаке AWS с нашей локальной сетью, которая не находится в облаке.

00:03:44.400 --> 00:03:50.400

Оплата за этот сервис происходит на каждый connection по часовой основе.

00:03:50.400 --> 00:03:56.400

В рамках этого сервиса доступны два варианта routing. Это Dynamic Routing и Static Routing.

00:03:56.400 --> 00:04:06.400

При использовании Dynamic Routing обязательное требование поддержки устройствами протокола Border Gateway Protocol, то есть BGP.

00:04:06.400 --> 00:04:12.400

Если этот протокол не поддерживается, то остается второй вариант, это Static Routing.

00:04:12.400 --> 00:04:22.400

Чем отличается Dynamic Routing от Static Routing? Так это тем, что в Dynamic Routing встроена нативная поддержка проверки состояния connection.

00:04:22.400 --> 00:04:29.400

И можно настроить достаточно надежное переключение на резервный канал.

00:04:29.400 --> 00:04:35.400

Когда мы говорим про Static Routing, нативно это не поддерживается, но настроить это возможно.

00:04:35.400 --> 00:04:43.400

Как происходит подключение локальной сети к облаку AWS?

00:04:43.400 --> 00:04:52.400

На стороне AWS создается ресурс Virtual Private Gateway, а на стороне локальной сети Customer Gateway.

00:04:52.400 --> 00:04:59.400

Она привязывается к соответствующему оборудованию и между ними настраивается VPN соединение.

00:04:59.400 --> 00:05:07.400

Следует обратить внимание также, что к одному Virtual Private Gateway можно подключить несколько Customer Gateway.

00:05:07.400 --> 00:05:15.400

Таким образом предоставить Highly Availability на стороне On-premise Network.

00:05:15.400 --> 00:05:21.400

На этом мы подошли к концу нашей второй части сегодняшней лекции.

00:05:21.400 --> 00:05:23.400

Давайте остановимся на основных моментах.

00:05:23.400 --> 00:05:39.400

Первое это то, что есть сервис AWS Site-to-Site VPN, который позволяет соединить локальную сеть с VPC на стороне облака AWS через создание VPN подключения поверх интернета.

00:05:39.400 --> 00:05:49.400

Сервис AWS Site-to-Site VPN поддерживает два варианта роутинга Static и Dynamic в зависимости от того, какие протоколы поддерживает ваше оборудование.

00:05:49.400 --> 00:06:06.400

И третье это то, что на стороне On-premise можно задействовать несколько Customer Gateway и все эти подключения направить к одному Virtual Private Gateway.

00:06:06.400 --> 00:06:16.400

Мы с вами добрались до третьей части нашей сегодняшней лекции и здесь подробнее рассмотрим сервис AWS Direct Connect.

00:06:16.400 --> 00:06:19.400

Также ее сокращенно называют AWS DX.

00:06:19.400 --> 00:06:27.400

Идея этого сервиса такая же, она помогает нам локальную сеть подключить с облаком AWS, но делает он это иначе.

00:06:27.400 --> 00:06:37.400

Когда мы говорили про сервис AWS Site-to-Site VPN, мы пробрасывали VPN канал поверх интернет подключения.

00:06:37.400 --> 00:06:46.400

И если у вас интернет подключения ненадежные либо слабые, то в этом случае VPN работает в рамках этих ограничений.

00:06:46.400 --> 00:06:53.400

В случае, если интернета не будет, то VPN подключения до вашего облака пропадает.

00:06:53.400 --> 00:06:59.400

Когда мы говорим про Direct Connect, здесь происходит качественно другое подключение.

00:06:59.400 --> 00:07:10.400

Мы с вами помним, есть глобальная сеть AWS и мы знаем, что есть партнеры AWS, которые распространены по всему миру.

00:07:10.400 --> 00:07:30.400

И они при настройке Direct Connect с глобальной инфраструктурой AWS подтягивают до своих дата-центров прямое подключение без выхода в интернет, что предоставляет большие возможности.

00:07:30.400 --> 00:07:46.400

То есть это подключение фактически равноценно тому, что глобальная инфраструктура AWS и ресурсы, находящиеся в дата-центрах партнера, могут общаться как будто бы в одной локальной сети.

00:07:46.400 --> 00:07:58.400

И это дает возможность создать канал больше пропускной способности. Предоставляется два варианта. Это 1 ГБит в секунду и 10 ГБит в секунду.

00:07:58.400 --> 00:08:06.400

Также при подключении через Direct Connect не происходит выхода в интернет, то есть мы независимы от интернета.

00:08:06.400 --> 00:08:11.400

Если интернет отключается, то ваше подключение к облаку AWS остается.

00:08:11.400 --> 00:08:26.400

Таким образом, для различных use-кейсов использование сервиса Direct Connect может быть более подходящим, чем AWS Site-to-Site VPN.

00:08:26.400 --> 00:08:37.400

Другой важный момент это то, что Direct Connect оплачивается индивидуально и может стоить в несколько раз дороже, чем AWS Site-to-Site VPN.

00:08:37.400 --> 00:08:48.400

Поэтому необходимо взвесить все аргументы за и против и после уже принимать решение о переходе к сервису AWS Direct Connect.

00:08:48.400 --> 00:09:03.400

Какие есть популярные use-кейсы? Первые - это гибридные инфраструктуры, то есть когда у вас часть инфраструктуры находится локально, а другая часть находится в облаке AWS.

00:09:03.400 --> 00:09:13.400

Вы не планируете мигрировать в облако либо обратно в локальный дата-центр, а продолжаете жить в таком режиме.

00:09:13.400 --> 00:09:26.400

В этом случае у вас может передаваться достаточно большой объем информации между облаком и локальной инфраструктурой, поэтому Direct Connect может быть удачным решением.

00:09:26.400 --> 00:09:43.400

Другой вариант - это случаи, когда передается большой объем информации и другие два варианта - это когда нам необходимо обеспечить более стабильную работу подключения в зависимости от вашей бизнес задачи.

00:09:43.400 --> 00:10:04.400

И также бывают некоторые регуляторные требования, которые не разрушают передачу некоторых чувствительных персональных данных передавать по сети интернет. В этом случае с Direct Connect мы не выходим в интернет и не нарушаем эти регуляторные требования.

00:10:04.400 --> 00:10:14.400

Давайте с вами теперь на примере этой диаграммы рассмотрим подключение локальной инфраструктуры с облаком AWS через сервис AWS Direct Connect.

00:10:14.400 --> 00:10:49.400

Посередине вы видите некоторые DX location, в котором находится ресурс DX endpoint. Этот DX endpoint подключается напрямую к глобальной инфраструктуре AWS с одной стороны, с другой стороны до него необходимо произвести подключение по двум каналам, приватному и публичному от вашего локального дата-центра до DX endpoint. Как только вы это проделываете вы через DX endpoint получаете возможность подключаться к любому VPC в любом регионе AWS.

00:10:49.400 --> 00:11:01.400

Почему? Потому что DX endpoint подключается напрямую к глобальной инфраструктуре AWS, а все VPC и так находятся там,поэтому появляется такая возможность.

00:11:01.400 --> 00:11:13.400

Также через DX endpoint вы можете через этот расширенный канал, выходя в интернет, обращаться к публичным сервисам AWS, например Amazon S3.

00:11:13.400 --> 00:11:37.400

Почему такой вариант может быть интересен? Это связано с небольшим каналом подключения в интернет от вашего дата центра. То есть если у вас канал небольшой, то через DX endpoint вы можете этот канал фактически расширить и передавать больше объема информации с большей скоростью.

00:11:37.400 --> 00:12:19.400

Давайте теперь рассмотрим следующий пример реализации высокой доступности подключения локального дата-центра с облаком AWS. В этом случае мы через партнера AWS используя сервис AWS Direct Connect подключили как основной канал локального дата-центра с облаком AWS и как резервный использовали AWS Site to Site VPN и подключили через интернет VPN соединение. Здесь хотелось бы отдельно напомнить, что в случае с Direct Connect есть два варианта канала. Это 1 ГБит в секунду и 10 ГБит в секунду.

00:12:19.400 --> 00:12:47.400

Когда мы говорим подключение через VPN, максимально оно может достигать 1.25 ГБит в секунду, но все зависит от вашего интернет соединения. Если сам канал не такой широкий, то поверх него быстрее этого канала вы передавать не сможете. Это и есть основное ограничение вашего VPN подключения.

00:12:47.400 --> 00:13:09.400

В другой случае представим, что вы международная компания, у вас есть несколько дата-центров, где вы хостите часть вашего приложения. Другая часть вашего приложения находится в облаке AWS и вам необходимо для корректного функционирования вашего приложения обеспечить постоянные соединения между облаком и локальными дата-центрами.

00:13:09.400 --> 00:13:28.400

В этом случае рекомендуется от каждого дата-центра настроить подключение в облако AWS через AWS Direct Connect. Таким образом, в случае неисправности одного из дата-центров у вас останется достаточно широкий канал, прямое подключение к облаку AWS и приложение продолжит работать.

00:13:28.400 --> 00:13:44.400

Мы с вами добрались до конца третьей части нашей сегодняшней лекции. Давайте вспомним самые основные моменты. Первое - это то, что AWS Direct Connect использует открытые стандарты для подключения напрямую к глобальной инфраструктуре AWS.

00:13:44.400 --> 00:14:04.400

При подключении через AWS Direct Connect у вас появляется возможность напрямую обратиться к любому VPC, к любому публичному AWS сервису, через него находящимся в любом регионе AWS. Это логично, так как все ресурсы AWS находятся в одной глобальной инфраструктуре.

00:14:04.400 --> 00:14:20.400

Поэтому подключившись к нему один раз, вы получаете доступ ко всем ресурсам. Также мы рассмотрели два примера настройки высокодоступного подключения из локального дата-центра в облако AWS.

00:14:20.400 --> 00:14:35.400

В первом случае используется Direct Connect и VPN соединение как резервные, а во втором случае рекомендуется использовать Direct Connect для каждого из дата-центров. Это позволит предоставить высокую доступность ваших приложений.

00:14:35.400 --> 00:14:55.400

Но здесь также хотелось бы напомнить, Direct Connect может стоить несколько раз дороже, чем Site-to-Site VPN, поэтому необходимо все взвесить и в случае, если вас этот сервис устраивает, то вы ее используете для подключения.

00:14:55.400 --> 00:15:08.400

Мы с вами добрались до четвертой части нашей сегодняшней лекции и здесь мы подробнее разберем подключение VPC в облаке и AWS между собой, используя функционал VPC Peering.

00:15:08.400 --> 00:15:48.400

Представим случай, у вас есть приложение, приложение растет и развивается и когда-то все ваши environment, то есть среды, находились внутри одной VPC, но с увеличением и развитием вашего приложения все большее количество ресурсов создается, все большее количество AWS лимитов вы достигаете, поэтому логичным решением с точки зрения управления, видимости и изолирования друг от друга разных нагрузок, как решение вы создаете разные VPC и к каждой среде привязываете определенные VPC.

00:15:48.400 --> 00:15:55.400

То есть в одном VPC будет находиться Dev среда, в другом VPC тестовая среда и в последней основная Production среда.

00:15:55.400 --> 00:16:08.400

Но в силу различных факторов у вас остается необходимость подключения от одной среды к другой для выполнения тех или иных задач.

00:16:08.400 --> 00:16:18.400

И в этом случае к нам на помощь приходит функционал VPC Peering, который дает возможность подключить один VPC к другому VPC напрямую.

00:16:18.400 --> 00:16:23.400

В этом случае не создается никаких VPN соединений, никакие Gateway не нужно создавать.

00:16:23.400 --> 00:16:40.400

Почему? Потому что обе VPC находятся в глобальной инфраструктуре AWS и вы фактически две локальные сети в рамках одной корпоративной сети соединяете между собой через локальную сеть.

00:16:40.400 --> 00:16:54.400

И в этом случае у вас нет выхода в интернет, вы остаетесь в той же глобальной сети и даже само подключение, оно имеет больший канал, больше данных можете передавать с большей скоростью.

00:16:54.400 --> 00:17:09.400

Таким образом, при таком подключении вы остаетесь в пространстве приватных IP адресов и с публичными IP адресами вам работать нет необходимости.

00:17:09.400 --> 00:17:18.400

Давайте рассмотрим на примере этой диаграммы, как происходит подключение двух VPC через VPC Peering.

00:17:18.400 --> 00:17:35.400

Самым первым делом создается Peering Connection, это некоторый ресурс. Как оно создается? На стороне AWS Management Console вы создаете Peering Connection и указываете VPC, который делает запрос на подключение на Peering.

00:17:35.400 --> 00:17:51.400

И вторым VPC указываете тот VPC, который вы хотите подключить. Здесь я бы хотел особо отметить, что VPC Peering вы можете подключать различные VPC между собой, находящиеся не только в одном регионе, но и в различных регионах.

00:17:51.400 --> 00:18:00.400

Более того, вы можете подключать VPC между собой не только в рамках одного AWS аккаунта, а подключиться к VPC в другом AWS аккаунте.

00:18:00.400 --> 00:18:09.400

Самый важный момент это то, что при создании Peering Connection отправляется приглашение подключения к соответствующему VPC.

00:18:09.400 --> 00:18:16.400

Если она находится в том же AWS аккаунте, то в том же AWS Management Console вы увидите приглашение.

00:18:16.400 --> 00:18:22.400

После того как вы ее принимаете, создается Peering Connection и две VPC между собой связываются.

00:18:22.400 --> 00:18:35.400

В случае, если второй VPC находится в другом AWS аккаунте, то либо у вас должен быть доступ к этому аккаунту, либо должна быть договоренность, что владелец этого AWS аккаунта примет приглашение.

00:18:35.400 --> 00:18:46.400

Как только вы это сделаете, останется несколько дополнительных шагов. Это настройка Routing Tables, Route Tables.

00:18:46.400 --> 00:18:58.400

Представим случай, у нас есть VPC A и у нее CIDR блок 10.1.0.0 и CIDR блок размера 16.

00:18:58.400 --> 00:19:09.400

Другой случай, это когда у нас есть VPC B и у нее CIDR блок 10.2.0.0 и размера 16. Это 65000 адресов.

00:19:09.400 --> 00:19:34.400

Для того чтобы вы смогли видеть ресурсы в VPC B из VPC A, вам необходимо в Route Table привязанном к VPC A прописать дополнительный Routing Rule, где Destination вы указываете CIDR блок VPC B и как Target указываете Peering Connection Resource.

00:19:34.400 --> 00:19:42.400

То есть вы через этот Peering Connection Resource идете в ресурсы в CIDR блоке VPC B.

00:19:42.400 --> 00:19:47.400

То же самое, но противоположно вы проделываете со стороны VPC B.

00:19:47.400 --> 00:20:06.400

Там для Route Table прописываете дополнительный Routing Rule, где как Destination указываете CIDR блок VPC A и как Target указываете тот же Peering Connection Resource, то есть PCX-1.

00:20:06.400 --> 00:20:12.400

Когда мы подключаем несколько VPC между собой, следует помнить о некоторых ограничениях.

00:20:12.400 --> 00:20:18.400

Мы работаем в пространстве приватных IP адресов.

00:20:18.400 --> 00:20:28.400

Второе это то, что мы можем подключать VPC, находящиеся в различных регионах и различных AWS аккаунтах.

00:20:28.400 --> 00:20:35.400

Другой очень важный момент это то, что CIDR блоки VPC не должны пересекаться.

00:20:35.400 --> 00:20:42.400

Если они пересекаются, то между ними настроить VPC Peering не получится.

00:20:42.400 --> 00:20:47.400

Между несколькими VPC вы можете создать только один Peering Resource.

00:20:47.400 --> 00:20:55.400

То есть не получится сделать резервный Peering Resource, так чтобы трафик в случае проблем с первым проходил через резервный.

00:20:55.400 --> 00:21:13.400

На самом деле даже необходимости в этом нет, так как Peering Connection Resource создается на стороне глобальной инфраструктуры AWS и уже в нем внутри сидит Highly Availability и дополнительно его чем-то компенсировать нет необходимости.

00:21:13.400 --> 00:21:27.400

И последний момент, который следует помнить, что при соединении нескольких VPC через VPC Peering эффект транзитивности не создается.

00:21:27.400 --> 00:21:32.400

То есть представьте, у вас есть VPC A, VPC B и VPC C.

00:21:32.400 --> 00:21:37.400

Далее представим, что вы сделали VPC Peering между VPC A и VPC B.

00:21:37.400 --> 00:21:43.400

Далее сделали VPC Peering Connection между VPC B и VPC C.

00:21:43.400 --> 00:21:52.400

И это не означает, что у вас появится возможность из VPC A обращаться к ресурсам, находящимся в VPC C.

00:21:52.400 --> 00:21:56.400

То есть эффекта транзитивности здесь не будет.

00:21:56.400 --> 00:22:14.400

Если вам необходимо, чтобы из VPC A можно было обращаться к ресурсам из VPC C и обратно, вам необходимо с учетом всех тех ограничений, которые мы перечислили ранее, создать новый Peering Connection между этими двумя VPC.

00:22:14.400 --> 00:22:29.400

Когда вы работаете с несколькими VPC, следует помнить, что необходимо создавать connection между двумя VPC только в тех местах, где есть необходимость.

00:22:29.400 --> 00:22:48.400

Так как в погоне за подключением всех VPC между собой мы можем прийти к тому, что у нас достаточно большое количество Peering Connection и это становится сложно контролируемым.

00:22:48.400 --> 00:22:57.400

Другой момент это то, что мы должны строить архитектуру таким образом, чтобы оно могло легко масштабироваться.

00:22:57.400 --> 00:23:07.400

Для тех случаев, когда у вас растет количество VPC и вам, тем не менее, необходимо их между собой соединять, есть другие специально созданные сервисы.

00:23:07.400 --> 00:23:11.400

Про них мы будем говорить в следующих секциях.

00:23:11.400 --> 00:23:15.400

Теперь давайте разберем один пример.

00:23:15.400 --> 00:23:23.400

Когда мы работаем в нескольких регионах, у нас есть основная инфраструктура, находящаяся в регионе A.

00:23:23.400 --> 00:23:28.400

Далее, есть часть инфраструктуры, представим DR, находится в регионе B.

00:23:28.400 --> 00:23:32.400

И в основном регионе у нас несколько различных VPC.

00:23:32.400 --> 00:23:38.400

И здесь представлен пример, когда в одном из VPC у нас так называемые shared resources.

00:23:38.400 --> 00:23:50.400

Это может быть что угодно, либо модуль приложения, либо может быть, например, Active Directory, который хранит информацию всех учеток наших пользователей внутри корпоративной сети.

00:23:50.400 --> 00:23:58.400

И в этом случае это тот VPC, к которому многие VPC должны будут подключаться.

00:23:58.400 --> 00:24:01.400

И здесь вы видите в диаграмме как раз таки пример.

00:24:01.400 --> 00:24:13.400

Мы видим, что наши VPC, обслуживающие Internal Application, подключаются напрямую через VPC Peering к нашему shared VPC.

00:24:13.400 --> 00:24:20.400

А для тех случаев, где нет необходимости подключаться к этому VPC, оно не настроено.

00:24:20.400 --> 00:24:32.400

Также вы видите, что настроено VPN подключение через сайт VPN из локального дата-центра в наш shared VPC.

00:24:32.400 --> 00:24:37.400

На этом мы подошли к концу четвертой части нашей сегодняшней лекции.

00:24:37.400 --> 00:24:40.400

Давайте остановимся и вспомним самые основные моменты.

00:24:40.400 --> 00:24:48.400

VPC Peering - это тот ресурс, тот функционал, который позволяет соединить напрямую два отдельно стоящих VPC между собой.

00:24:48.400 --> 00:24:55.400

Эти VPC могут находиться в различных регионах, также они могут находиться в различных AWS аккаунтах.

00:24:55.400 --> 00:25:13.400

При создании VPC Peering тот VPC, который отправляет запрос, создает Peering Connection, а вторая часть VPC, которая принимает это приглашение, должна это приглашение принять и после этого у нас активируется Peering Connection ресурс.

00:25:13.400 --> 00:25:22.400

Далее нам необходимо настроить route tables и возможно security группы, чтобы ресурсы между двумя VPC могли обмениваться информацией.

00:25:22.400 --> 00:25:27.400

При создании Peering Connection есть определенные ограничения.

00:25:27.400 --> 00:25:41.400

Мы работаем в пространстве приватных IP адресов, то есть публичные IP адреса не задействованы, так как мы остаемся в границах глобальной инфраструктуры AWS и в интернет не выходим.

00:25:41.400 --> 00:25:49.400

Другой момент это то, что при создании подключения между несколькими VPC, CIDR блоки не должны пересекаться.

00:25:49.400 --> 00:25:52.400

Если они пересекаются, то Peering Connection настроить не получится.

00:25:52.400 --> 00:26:06.400

Поэтому нужно заблаговременно это продумать и при создании любых VPC в рамках организации создавать их таким образом, чтобы CIDR блоки изначально не пересекались.

00:26:06.400 --> 00:26:11.400

Даже если в будущем они не будут соединены, наперед мы этого знать не можем.

00:26:11.400 --> 00:26:20.400

Намного дешевле это в начале учесть, нежели потом VPC со всеми ее ресурсами переносить в другой VPC.

00:26:20.400 --> 00:26:23.400

Это потребует немалых усилий и затрат компании.

00:26:26.400 --> 00:26:37.400

Между двумя VPC невозможно настроить несколько Peering Connection, но в этом даже нет необходимости, так как Highly Availability уже сидит внутри Peering Connection.

00:26:37.400 --> 00:26:44.400

И последний момент, который часто приходит на экзамене в виде вопроса, это эффект транзитивности.

00:26:44.400 --> 00:26:53.400

Если мы подключаем VPC A к VPC B, дальше VPC B к VPC C, то между VPC A и VPC C связь не появляется.

00:26:53.400 --> 00:27:00.400

Если вам нужно их между собой соединить, то нужно явно создать Peering Connection между двумя этими VPC.

00:27:03.400 --> 00:27:11.400

Мы с вами добрались до пятой части нашей сегодняшней лекции, и здесь подробнее остановимся на сервисе AWS Transit Gateway.

00:27:11.400 --> 00:27:27.400

Представим случай, у нас есть несколько VPC, и необходимо предоставить возможность каждой VPC подключаться до любого другого VPC в рамках нашей инфраструктуры.

00:27:27.400 --> 00:27:44.400

Проблема с VPC Peering в том, что при увеличении количества VPC и необходимости обеспечить полное подключение между всеми VPC, количество VPC Peering очень быстро растет.

00:27:44.400 --> 00:27:46.400

Давайте попробуем вместе посчитать.

00:27:46.400 --> 00:27:50.400

Представим, что у нас есть две VPC, и нужно соединить между собой.

00:27:50.400 --> 00:27:53.400

В этом случае у нас создается один Peering Connection.

00:27:53.400 --> 00:28:12.400

Далее, если у нас увеличивается количество VPC на одну и равняется трем, в этом случае уже необходимо создать три VPC Peering Connection, то есть между первым и вторым, вторым и третьим, а также между третьим и первым, так как транзитивность не поддерживается.

00:28:12.400 --> 00:28:25.400

Если мы говорим про четыре VPC, то в этом случае вы можете представить, что чтобы все VPC соединить между собой, нам уже нужно 6 Peering Connection.

00:28:25.400 --> 00:28:39.400

Если же попробуем посчитать случай с пятью VPC, то в этом случае нам необходимо будет 15 Peering Connection.

00:28:39.400 --> 00:28:44.400

И вы видите, как быстро растет необходимое количество VPC Peering.

00:28:44.400 --> 00:28:57.400

Для того, чтобы вот этот конкретный случай легче управлять, было разработано специальное решение.

00:28:57.400 --> 00:29:01.400

Это сервис AWS Transit Gateway.

00:29:01.400 --> 00:29:28.400

Идея ее в том, что вы подключаете некоторый сетевой Connection, будь то VPC, либо подключение от локального дата-центра в виде VPN Connection, либо в виде Direct Connect Gateway, то при подключении к AWS Transit Gateway все существующие подключения, становятся доступными для нового подключения.

00:29:28.400 --> 00:29:48.400

То есть в этом случае вы видите на примере шесть различных подключений, и нам не пришлось создавать более 15 Peering Connection, а достаточно сделать по одному Connection для каждого из источника до AWS Transit Gateway.

00:29:48.400 --> 00:30:00.400

И таким образом каждый источник получил возможность подключаться к любым другим подключениям, которые находятся в этом Transit Gateway.

00:30:00.400 --> 00:30:03.400

Давайте теперь подробнее поговорим, что это за сервис.

00:30:03.400 --> 00:30:17.400

AWS Transit Gateway - это так называемый networking сервис, который позволяет легче подключать между собой большое количество сетевых подключений.

00:30:17.400 --> 00:30:29.400

Оно работает в виде центрального хаба, и все подключения, которые к нему подключаются, получают доступ ко всем другим подключенным ресурсам.

00:30:29.400 --> 00:30:40.400

На один AWS Transit Gateway можно подключить до 5000 различных Connection.

00:30:40.400 --> 00:30:54.400

Здесь я бы хотел обратить внимание, что для Transit Gateway, также и для Site-to-Site VPN сервиса имеются определенные ограничения, недостатки и преимущества.

00:30:54.400 --> 00:30:56.400

Также они отличаются по прайсингу.

00:30:56.400 --> 00:31:11.400

Поэтому в зависимости от вашего бизнес-кейса, вам обязательно нужно все эти моменты учесть, а именно моменты, связанные с архитектурой, а также, что также немаловажно, это вопросы оплаты.

00:31:11.400 --> 00:31:20.400

Может получиться так, что Transit Gateway для вас невыгодное решение, и следует оставаться с VPC Peering Connection.

00:31:20.400 --> 00:31:42.400

Либо обратный случай, VPC Peering Connection вам не подходит, так как количество подключений растет и это невыгодно и может оказаться, что Transit Gateway не только упрощает управление всеми этими Connection, но и выходит более выгодным по расходам.

00:31:42.400 --> 00:31:50.400

Для того, чтобы лучше понять, как происходит настройка AWS Transit Gateway, давайте рассмотрим следующий пример.

00:31:50.400 --> 00:31:53.400

У нас есть 3 VPC со следующими CIDR блоками.

00:31:53.400 --> 00:32:01.400

Нам необходимо соединить их все между собой, без использования Peering Connection, а с использованием Transit Gateway.

00:32:01.400 --> 00:32:05.400

Первым делом нам необходимо создать ресурс AWS Transit Gateway.

00:32:05.400 --> 00:32:12.400

Далее, на стороне каждого VPC необходимо настроить ENI, то есть Elastic Network Interface.

00:32:12.400 --> 00:32:17.400

Это тот ресурс, через который происходит подключение к VPC.

00:32:17.400 --> 00:32:32.400

Подключаем ее к Route Table, то есть у каждого VPC есть настроенный дефолтовый Route Table, который разрешает трафик между всеми ресурсами в рамках конкретного VPC.

00:32:32.400 --> 00:32:37.400

Теперь нам необходимо произвести изменения в каждой из Route Table.

00:32:37.400 --> 00:32:50.400

На примере третьего VPC мы видим, что мы указали как Destination CIDR блок, охватывающий CIDR блоки всех трех VPC, и как Target указали Transit Gateway.

00:32:50.400 --> 00:32:58.400

Далее, следующим шагом нам необходимо на стороне Transit Gateway настроить Route Table.

00:32:58.400 --> 00:33:13.400

И здесь мы указываем, что как Destination к каждому из VPC мы указываем Route с Target Elastic Network Interface соответствующего VPC.

00:33:13.400 --> 00:33:27.400

После чего при достижении трафика до Transit Gateway, Transit Gateway смотрит куда направляется трафик, исходя из правила в Route Table, смотрит какой Target использовать.

00:33:27.400 --> 00:33:55.400

Представим, что мы с VPC 3 обращаемся к ресурсам из CIDR блока первого VPC, то он посмотрит на IP адрес, увидит на стороне Route Table у третьего VPC Route, который говорит, что Target мы используем Transit Gateway и обращаемся на вот этот CIDR блок.

00:33:55.400 --> 00:33:59.400

После чего трафик уходит на сторону Transit Gateway.

00:33:59.400 --> 00:34:10.400

Transit Gateway также смотрит на свой Route Table и видит, что тот IP адрес, к которому мы обращаемся, относится к первому VPC.

00:34:10.400 --> 00:34:35.400

А чтобы направить трафик к первому VPC, нам необходимо использовать соответствующий Elastic Network Interface и направлять, соответственно, на этот ресурс этот трафик. После чего можно будет этот трафик увидеть на стороне VPC 1, то есть фактически connection будет предоставлен.

00:34:35.400 --> 00:34:38.400

Теперь рассмотрим следующий случай.

00:34:38.400 --> 00:34:42.400

У нас есть VPN connection с локального дата-центра.

00:34:42.400 --> 00:34:59.400

И первый вариант – это предоставить изолированное подключение от каждого VPC до ресурсов в локальном дата-центре, так чтобы VPC между собой не пересекались.

00:34:59.400 --> 00:35:17.400

В этом случае нам достаточно изменить Route Table в Transit Gateway и указать как Destination обозначение всего интернета, либо определенный CIDR блок и направить этот трафик VPN.

00:35:17.400 --> 00:35:32.400

Таким образом, даже если вы как трафик указываете IP адрес из первого VPC, то он все равно, согласно Route Rule, направит ее в сторону VPN.

00:35:32.400 --> 00:35:39.400

И ресурсы, находящиеся в разных VPC, между собой обмениваться трафиком не смогут.

00:35:39.400 --> 00:35:45.400

Но при этом они смогут направлять трафик через VPN в локальные дата-центры.

00:35:45.400 --> 00:36:03.400

Следующий случай – это когда нам необходимо весь трафик, который остается внутри AWS, но идет в другой VPC, сохранить, но так, чтобы другой трафик направился в сторону VPN.

00:36:03.400 --> 00:36:14.400

В этом случае, мы можем либо изменить Route Table у Transit Gateway, либо добавить еще один Route Table, в котором мы укажем предыдущие Routes.

00:36:14.400 --> 00:36:29.400

Таким образом, когда мы захотим из третьего VPC, например, обратиться к ресурсам из второго VPC, согласно Route Table второму, мы успешно направим наш трафик.

00:36:29.400 --> 00:36:49.400

Далее, если мы захотим направить наш трафик не к VPC, а к другому IP адресу, он не совпадет с правилами, связанными с Route Table вторым, и уже посмотрит Route Table первый, увидит, что этот трафик необходимо направить на сторону VPN, и соответственно туда этот трафик уйдет.

00:36:53.400 --> 00:37:01.400

Для того, чтобы закрепить, как необходимо производить настройку в случае AWS Transit Gateway, давайте сделаем небольшой челлендж.

00:37:01.400 --> 00:37:09.400

В рамках этого челленджа нам дается 5 различных VPC с уникальным CIDR блоком, не пересекающимся.

00:37:09.400 --> 00:37:18.400

Также есть у нас Transit Gateway, и нам необходимо все VPC соединить полностью между собой через AWS Transit Gateway.

00:37:18.400 --> 00:37:28.400

Какие действия необходимо сделать на уровне Route Table каждого VPC, и какие действия необходимо произвести на уровне Route Table Transit Gateway.

00:37:28.400 --> 00:37:32.400

Вы можете остановить это видео и попытаться ответить самостоятельно.

00:37:37.400 --> 00:37:43.400

Отлично, я думаю, вы смогли правильно ответить, давайте теперь проверим, что же необходимо было сделать.

00:37:43.400 --> 00:37:49.400

Первым делом, на уровне каждого VPC нам необходимо добавить один Route.

00:37:49.400 --> 00:38:00.400

Этот Route как Destination содержит CIDR блок, этот CIDR блок должен охватывать CIDR блоки всех существующих VPC, и как Target необходимо указать наш Transit Gateway.

00:38:00.400 --> 00:38:14.400

Таким образом, мы помним с вами, что на стороне Route Table дефолтовый первый Route содержит CIDR блок этого VPC, и как Target указывает local.

00:38:14.400 --> 00:38:33.400

Это означает, что если IP адрес, которому направляется трафик, относится к CIDR блоку конкретного VPC, то он распознается как локальный трафик и, соответственно, направляется к ресурсу, находящемся в этом VPC.

00:38:33.400 --> 00:38:39.400

То есть, это тот Route, который позволяет всем ресурсам внутри этого VPC общаться между собой.

00:38:39.400 --> 00:39:07.400

Теперь следующий Route, который мы добавили, он охватывает все другие CIDR блоки, и если у нас есть обращение к IP адресу, который не относится к CIDR блоку конкретного VPC, а уходит наружу, то в этом случае мы проверяем, относится ли он к общему CIDR блоку.

00:39:07.400 --> 00:39:10.400

Если да, то этот трафик уходит к Transit Gateway.

00:39:10.400 --> 00:39:13.400

Подобные изменения необходимо произвести на уровне каждого VPC.

00:39:13.400 --> 00:39:23.400

Теперь если мы говорим про Transit Gateway, здесь нам необходимо в Route Table добавить 5 Routes по 1 Route для каждого VPC.

00:39:23.400 --> 00:39:25.400

Этот Route содержит следующее.

00:39:25.400 --> 00:39:34.400

Как Destination мы указываем CIDR block VPC, и как Target указываем Elastic Network Interface этого же VPC.

00:39:34.400 --> 00:39:55.400

То есть если мы получаем в Transit Gateway трафик, который направляется на такой-то IP адрес, он проверяет все свои Routes, и если этот IP адрес относится к одному из CIDR блоков в Destination, то он смотрит на Target и направляет этот трафик к этому Target.

00:39:55.400 --> 00:40:07.400

Таким образом, мы смогли настроить связь между всеми VPC, используя Transit Gateway.

00:40:07.400 --> 00:40:16.400

На этом мы подошли к концу 5-й секции, и здесь можем отметить основные моменты, связанные с сервисом AWS Transit Gateway.

00:40:16.400 --> 00:40:25.400

AWS Transit Gateway выполняет ту же роль, что и VPC Peering Connection, но отличается тем, что является центральным хабом.

00:40:25.400 --> 00:40:44.400

В случае с Peering Connection мы подключаем две VPC между собой напрямую, а в случае Transit Gateway нам достаточно подключить VPC к Transit Gateway, и он получает доступ ко всем существующим connection, подключенным к этому же Transit Gateway.

00:40:44.400 --> 00:40:54.400

В зависимости от вашего бизнес-кейса или задачи, вам необходимо будет определиться, что является для вас оптимальным.

00:40:54.400 --> 00:41:16.400

Либо использовать VPC Peering Connection в случае, если у вас небольшое количество соединений между VPC, и в обратном случае, если у вас большое количество VPC, и необходимо, чтобы каждый VPC был соединен с каждым другим VPC, то в этом случае более подходящим решением для вас может быть AWS Transit Gateway.

00:41:16.400 --> 00:41:33.400

Мы с вами добрались до шестой и заключительной части нашей сегодняшней лекции, и здесь мы поговорим про возможности, позволяющие ресурсам из VPC обращаться к другим AWS сервисам.

00:41:33.400 --> 00:41:38.400

Для того, чтобы это реализовать, рекомендуется использовать VPC Endpoints.

00:41:38.400 --> 00:41:55.400

Это некоторый ресурс, который позволяет, не выходя в интернет, оставаясь в рамках глобальной инфраструктуры AWS, подключаться ресурсам из VPC напрямую к AWS сервисам.

00:41:55.400 --> 00:42:03.400

Таким образом, нам нет необходимости настраивать Internet Gateway, VPN соединение, NAT Gateway и так далее.

00:42:03.400 --> 00:42:10.400

Существует два варианта подключения VPC Endpoints.

00:42:10.400 --> 00:42:25.400

Первый - это Interface Endpoint, когда нам выделяется приватный IP адрес, и через этот приватный IP адрес, используя функционал AWS Private Link, мы подключаемся к другим сервисам.

00:42:25.400 --> 00:42:32.400

Не все сервисы доступны. Примеры этих сервисов - это Amazon CloudWatch, Amazon EC2 и Elastic Cloud Balancing.

00:42:32.400 --> 00:42:35.400

Второй вариант - это Gateway Endpoint.

00:42:35.400 --> 00:42:44.400

В этом случае производятся изменения на стороне route table, чтобы определенный трафик направлять на поддерживаемые AWS сервисы.

00:42:44.400 --> 00:42:49.400

Примерами в этом случае являются Amazon S3 и DynamoDB.

00:42:49.400 --> 00:42:56.400

Давайте теперь посмотрим на примерах, как мы можем настроить Interface Endpoint.

00:42:56.400 --> 00:43:06.400

Самым первым делом нам необходимо решить, с каким сервисом мы будем работать и поддерживается ли этот сервис через Interface Endpoint.

00:43:06.400 --> 00:43:15.400

Если поддерживается, далее нам необходимо определиться с VPC, определиться с subnet, откуда ресурсы могут подключаться к этому сервису.

00:43:15.400 --> 00:43:20.400

В четвертом шаге вы видите, по умолчанию можно активировать приватный DNS.

00:43:20.400 --> 00:43:22.400

Я чуть позже расскажу, для чего это нужно.

00:43:22.400 --> 00:43:32.400

И последним шагом мы для нашего Network Interface настраиваем необходимые Security Groups.

00:43:32.400 --> 00:43:41.400

Теперь давайте рассмотрим примеры, как это все взаимодействует.

00:43:41.400 --> 00:43:56.400

Когда мы настраиваем подключение из нашего VPC к определенному сервису, пусть это будет Amazon Kinesis Data Streams, мы можем к нему обратиться через default DNS hostname.

00:43:56.400 --> 00:44:11.400

Это URL сервиса, через который мы можем обратиться к этому сервису, но проблема в том, что это обращение идет через Internet gateway, через интернет и вы дополнительным параметром указываете, что вы хотите.

00:44:11.400 --> 00:44:15.400

Для некоторых нагрузок это неприемлемо.

00:44:15.400 --> 00:44:19.400

Есть другой вариант. Можно воспользоваться Endpoint Specific DNS Hostname.

00:44:19.400 --> 00:44:27.400

Это когда для нашего Interface Endpoint назначается некоторый DNS Hostname.

00:44:27.400 --> 00:44:33.400

Грубо говоря, это URL адрес, по обращению который трафик направляется на нужный вам сервис.

00:44:33.400 --> 00:44:37.400

Так вот, в этом случае Private DNS он не включен.

00:44:37.400 --> 00:44:42.400

Таким образом, если мы хотим обратиться к этому сервису, у нас есть два варианта.

00:44:42.400 --> 00:44:50.400

В случае обращения через default DNS hostname трафик идет через Internet gateway, через интернет добирается до нашего сервиса.

00:44:50.400 --> 00:45:06.400

Либо нам необходимо в стороне наших приложений явно указывать наш специфичный уникальный URL, который был сгенерирован в рамках Interface Endpoint с подключением к Amazon Kinesis Data Stream.

00:45:06.400 --> 00:45:20.400

При обращении по этому URL оно происходит в рамках глобальной инфраструктуры AWS, не выходит в интернет и ресурсы внутри VPC напрямую обращаются к определенному сервису.

00:45:20.400 --> 00:45:54.400

Теперь другой случай, когда мы активировали Private DNS, то в этом случае нет необходимости через интернет, через Internet gateway направлять трафик по дефолтовому DNS Hostname, а этот дефолтовый DNS Hostname также остается активным, но при обращении по этому URL мы направляем наш трафик через Interface gateway на нужный нам сервис.

00:45:54.400 --> 00:46:04.400

Грубо говоря, мы знаем дефолтовый DNS Hostname для каждого из сервисов, она не меняется, она стандартная.

00:46:04.400 --> 00:46:39.400

Так вот, если мы хотим ее использовать, что очень удобно, мы можем включить опцию Private DNS и таким образом для наших приложений внутри VPC нет необходимости дополнительно там в переменных средах указывать Endpoint Specific DNS Hostname, а сразу, зная публичный URL определенного сервиса, напрямую к нему обращаться, зная, что он идет через Interface Endpoint в рамках глобальной инфраструктуры AWS и без выхода в интернет.

00:46:39.400 --> 00:46:53.400

Отлично, на этом мы добрались до конца шестой части нашей сегодняшней лекции, разобрали, что у нас есть возможность для ресурсов из VPC подключаться к сервисам AWS.

00:46:53.400 --> 00:47:01.400

Есть два варианта. Первый вариант - это через Interface gateway, а второй через Gateway Endpoint.

00:47:01.400 --> 00:47:18.400

В том и ином случае поддерживаются те или иные сервисы AWS, в зависимости от того, какой сервис вам нужен, и вы сможете посмотреть список поддерживаемых сервисов и понять с каким из вариантов вам необходимо двигаться.

00:47:21.400 --> 00:47:25.400

На этом мы подошли к концу нашей сегодняшней лекции.

00:47:25.400 --> 00:47:31.400

Мы разобрали достаточно важные вопросы, связанные с networking.

00:47:31.400 --> 00:47:43.400

Мы рассмотрели вопросы, как мы можем подключить нашу локальную инфраструктуру с облаками AWS, какие для этого есть возможности, как минимум два варианта.

00:47:43.400 --> 00:47:50.400

Также мы рассмотрели вариант подключения VPC, нескольких VPC между собой внутри облака AWS.

00:47:50.400 --> 00:47:53.400

Также есть несколько вариантов, в зависимости от вашего случая.

00:47:53.400 --> 00:48:10.400

И последний вариант, мы также рассмотрели случай, когда сервис не может быть расположен внутри VPC, но из VPC мы можем работать безопасно с этими сервисами, используя VPC Endpoint.

00:48:13.400 --> 00:48:20.400

На этом слайде вы можете видеть ключевые слова, которые помогут вам выйти на дополнительные ресурсы в интернете.

00:48:20.400 --> 00:48:32.400

Преимущественно они ведут документацию AWS, либо на записи из re:Invent, находящейся в видео хостинге YouTube.

00:48:34.400 --> 00:48:37.400

На этом мы завершаем нашу сегодняшнюю лекцию.

00:48:37.400 --> 00:48:44.400

Спасибо за внимание, увидимся с вами на следующих наших активностях.