WEBVTT

00:00:00.000 --> 00:00:02.520

Добрый день, уважаемые студенты!

00:00:02.520 --> 00:00:07.400

Я рад вас всех видеть на очередной сессии обзора лабораторной работы.

00:00:07.400 --> 00:00:14.840

Тема сегодняшней лабораторной работы - это настройка networking для нашего бизнес кейса, то есть для кафе.

00:00:14.840 --> 00:00:29.080

Эта лабораторная работа является Challenge Lab, то есть будет даваться минимальное количество инструкции и нам необходимо самостоятельно найти решение.

00:00:29.080 --> 00:00:38.040

Также я бы хотел отметить, что сама лабораторная работа достаточно объемная и я надеюсь, вам она понравится и будет интересно ее выполнять.

00:00:38.040 --> 00:00:42.160

Итак, давайте начнем.

00:00:42.160 --> 00:00:48.320

Начинаем мы с вами лабораторную работу с основной страницы курса Cloud Architecting в AWS Academy.

00:00:48.320 --> 00:00:53.120

Нам необходимо воспользоваться навигационным меню слева и перейти к модулю.

00:00:53.120 --> 00:01:03.760

Здесь нас интересует шестой модуль, а именно ссылка на лабораторную работу Challenge Lab Creating VPC Networking Environment for the Cafe.

00:01:03.760 --> 00:01:15.720

Как только вы окажетесь на главной странице лабораторной работы, необходимо будет скрыть окно Terminal, чтобы оно нам не мешало и перепроверить заголовок лабораторной работы.

00:01:15.720 --> 00:01:21.040

Если все сделано, мы переходим к нашему бизнес кейсу.

00:01:21.040 --> 00:01:33.480

Согласно нему, София и Нихил недавно только улучшили архитектуру в облаке AWS и перенесли базу данных в EC2 Instance в Managed Service Amazon RDS.

00:01:33.480 --> 00:01:54.440

Это улучшило работу всей инфраструктуры, разгрузило ребят от административных работ, связанных с поддержкой базы данных, и также они перенесли ресурсы базы данных из public subnet в private subnet, его еще больше обезопасив.

00:01:54.440 --> 00:01:56.640

Они на днях встретились с Матео.

00:01:56.640 --> 00:02:10.000

Матео является системным администратором в AWS и он выслушал обновление наших главных героев и посоветовал еще дальше улучшать архитектуру.

00:02:10.000 --> 00:02:24.280

Он предложил настроить корректно VPC, создать все необходимые ее компоненты и также создать все необходимые ресурсы для удаленного доступа с административными правами.

00:02:24.280 --> 00:02:33.720

Также необходимо будет добавить дополнительные уровни безопасности, чтобы это решение работало корректно и злоумышленники не могли его взломать.

00:02:33.720 --> 00:02:42.840

В рамках лабораторной работы мы с вами научимся создавать VPC и ее основные компоненты.

00:02:42.840 --> 00:02:52.000

Далее научимся создавать ресурсы, которые позволяют безопасно, удаленно подключаться к ресурсам из private subnet.

00:02:52.000 --> 00:03:05.280

А также мы настроим дополнительные уровни безопасности, чтобы еще больше обезопасить периметр нашего VPC.

00:03:05.280 --> 00:03:15.520

Вы видите на слайде, какой станет архитектура в облаке AWS после выполнения лабораторной работы.

00:03:15.520 --> 00:03:24.040

На выполнение этой лабораторной работы рекомендуется выделить 90 минут.

00:03:24.040 --> 00:03:32.480

Если этого времени будет недостаточно, вы всегда можете не нажимая кнопку End lab нажать на кнопку Start lab и обнулить счетчик.

00:03:32.480 --> 00:03:48.080

Также здесь нам напоминают о том, что создается временный AWS аккаунт, специально предназначенный для лабораторной работы и все сервисы и компоненты, не относящиеся к лабораторной работе могут быть недоступны.

00:03:48.080 --> 00:03:54.760

А все те сервисы и компоненты, которые используются в лабораторной работе, они гарантированно будут доступны.

00:03:54.760 --> 00:04:06.800

Для того, чтобы начать лабораторную работу необходимо нажать на кнопку Start lab, подтвердить то, что мы начинаем лабораторную работу и дождаться завершения создания временного AWS аккаунта.

00:04:06.800 --> 00:04:09.360

Вы увидите статус, что он готов к работе.

00:04:09.360 --> 00:04:20.640

После чего вы можете нажатием кнопки AWS перейти во временный AWS аккаунт и начать выполнять лабораторную работу.

00:04:20.640 --> 00:04:38.200

Мы с вами добрались до первого блока задания и в рамках бизнес кейса нам необходимо создать все необходимые компоненты внутри VPC, начиная от bastion host до subnet, которые еще не созданы.

00:04:38.200 --> 00:04:43.560

Далее нам необходимо сделать таким образом, чтобы bastion host был доступен из интернета.

00:04:43.560 --> 00:05:01.280

После этого создать все необходимое, чтобы из private subnet можно было подключаться безопасно к интернету, но чтобы доступ к ресурсам в private subnet был только из EC2 инстанса из public subnet.

00:05:01.280 --> 00:05:03.720

В нашем случае это bastion host.

00:05:03.720 --> 00:05:04.960

Итак, давайте начнем.

00:05:04.960 --> 00:05:05.960

Первое задание.

00:05:05.960 --> 00:05:09.840

В рамках первого задания нам необходимо создать public subnet.

00:05:09.840 --> 00:05:21.880

Обратите внимание, что при просмотре списка существующих VPC во временном AWS аккаунте VPC с названием Lab VPC уже было создано за нас.

00:05:21.880 --> 00:05:24.000

Мы его будем использовать в дальнейшем.

00:05:24.000 --> 00:05:25.000

Следующий шаг.

00:05:25.000 --> 00:05:27.520

Нам необходимо создать public subnet.

00:05:27.520 --> 00:05:36.160

Для этого мы переходим на списки subnet-ов и нажмем на соответствующую кнопку для создания нового subnet.

00:05:36.160 --> 00:05:45.040

Как Name укажем Public subnet, как VPC выберем Lab VPC, как Availability Zone необходимо выбрать ту, которая заканчивается на a.

00:05:45.040 --> 00:05:55.160

В случае, если временный AWS аккаунт был создан в регионе North Virginia, то как availability зону необходимо будет выбрать us-east-1a.

00:05:55.160 --> 00:06:01.040

Как CIDR block укажем следующее значение и создадим наш subnet.

00:06:01.040 --> 00:06:11.680

Далее нам необходимо создать Internet gateway и привязать ее к нашему VPC.

00:06:11.680 --> 00:06:25.160

Следующим шагом нам необходимо настроить route table нашего public subnet так, чтобы интернет трафик как target использовал наш Internet gateway.

00:06:25.160 --> 00:06:37.520

После этого необходимо будет route table привязать к нашему public subnet в случае, если был создан кастомный route table.

00:06:37.520 --> 00:06:57.240

А в случае, если мы использовали и изменили default route table, то привязывать нет необходимости, оно уже автоматически привязано ко всем subnet внутри VPC, где мы явно не указывали кастомный route table.

00:06:57.240 --> 00:07:03.640

Во второй задаче нам необходимо создать bastion host.

00:07:03.640 --> 00:07:13.920

Bastion host мы будем создавать в public subnet, это тот EC2 инстанс, к которому мы будем через интернет подключаться по SSH.

00:07:13.920 --> 00:07:19.520

Давайте рассмотрим какие параметры необходимо указать при создании этого EC2 инстанса.

00:07:19.520 --> 00:07:26.720

Как операционную систему мы используем Amazon Linux, версию Amazon Linux 2 AMI (HVM).

00:07:26.720 --> 00:07:29.360

Instance type должен быть t2.micro.

00:07:29.360 --> 00:07:37.540

Далее в признаке Auto-assign public IP необходимо будет галочку убрать, чтобы публичный IP адрес не присвоился.

00:07:37.540 --> 00:07:45.880

Почему это нужно мы сейчас посмотрим и дойдем чуть дальше в следующем шаге.

00:07:45.880 --> 00:08:11.600

Как Name необходимо указать Bastion host и также необходимо создать Security group с названием Bastion host SG и по нему разрешить только inbound traffic по 22 порту, то есть это SSH трафик, который разрешает inbound трафик только для нашего IP адреса.

00:08:11.600 --> 00:08:17.560

Как private key будем использовать vockey.

00:08:17.560 --> 00:08:23.520

Теперь на третьем задании нам необходимо будет создать Elastic IP адрес.

00:08:23.520 --> 00:08:35.200

Как вы помните Bastion host это тот инстанс, к которому подключаются удаленно все сотрудники компании и оно является единой точкой входа.

00:08:35.200 --> 00:08:40.600

Поэтому очень важно, чтобы ее IP адрес был известен всем и не менялся.

00:08:40.600 --> 00:08:59.960

Если бы мы создали для инстанса публичный IP адрес, то при пересоздании по тем или иным причинам bastion host у него публичный IP адрес мог бы смениться и вернуть на предыдущий мы бы никак не смогли.

00:08:59.960 --> 00:09:11.720

Таким образом сотрудники, у которых было записано старый IP адрес bastion host не смогли бы подключиться к нашей инфраструктуре в облаке.

00:09:11.720 --> 00:09:19.840

Поэтому правильным решением будет зафиксировать публичный IP адрес, а для этого мы создаем Elastic IP адрес.

00:09:19.840 --> 00:09:27.180

Как только мы ее создадим, нам необходимо будет ее привязать к нашему bastion host.

00:09:27.180 --> 00:09:32.240

На четвертом задании нам необходимо проверить соединение к bastion host.

00:09:32.240 --> 00:09:46.960

Для этого необходимо на странице задания лабораторной работы нажать на кнопку Details и в выпадающем списке выбрать возле надписи AWS кнопку Show.

00:09:46.960 --> 00:09:52.000

Здесь у вас есть возможность скачать приватный ключ в двух вариантах.

00:09:52.000 --> 00:09:59.040

В случае если вы работаете на Windows операционной системе вам необходимо скачать ключ с окончанием PPK.

00:09:59.040 --> 00:10:10.400

Если же вы работаете на Linux операционной системе либо macOS, в этом случае вам необходимо скачать ключ с расширением PEM.

00:10:10.400 --> 00:10:17.360

Как только вы скачали, вы можете закрыть это окно и попробовать подключиться к bastion host.

00:10:17.360 --> 00:10:32.120

Как подключаться к удаленному EC2 инстансу мы не раз проделывали в лабораторных работах и Демо активностях, поэтому здесь мы не будем останавливаться на этом подробно.

00:10:32.120 --> 00:10:39.440

На пятом задании нам необходимо создать private subnet.

00:10:39.440 --> 00:10:58.120

Параметры private subnet как Name укажем Private subnet, как Availability zone укажем ту, которую указывали для Public subnet, как CIDR блок укажем следующее значение и на этом мы создадим private subnet.

00:10:58.120 --> 00:11:06.400

Но для того чтобы предоставить возможность выхода в интернет безопасно из private subnet нам необходимо создать NAT gateway.

00:11:06.400 --> 00:11:27.680

NAT gateway создается в public subnet и напоминаю, что для NAT gateway в связке необходимо создать Elastic IP адрес. После чего в route table нашего private subnet необходимо добавить routing rule, который будет направлять интернет трафик на наш NAT gateway.

00:11:27.680 --> 00:11:37.920

В рамках задания лабораторной работы нам необходимо создать route table с названием private route table.

00:11:37.920 --> 00:11:55.200

Это очень важно, так как при автооценивании нашей выполненной работы это может также учитываться и сказаться на ваших баллах, поэтому необходимо в точности соблюдать значение, которое мы должны вводить для наших ресурсов.

00:12:07.200 --> 00:12:24.560

На седьмом задании мы должны будем создать EC2 instance в private subnet для последующей проверки подключения из bastion host в EC2 instance из private subnet.

00:12:24.560 --> 00:12:55.080

Давайте рассмотрим какие параметры необходимо ввести, как AMI используем Amazon Linux2 AMI (HVM), как Instance type используем t2.micro, как имя укажем Private instance и в security группе откроем inbound трафик только по 22 порту для security группы bastion host.

00:12:55.080 --> 00:13:07.480

Также обратите внимание, что для инстанса в private subnet мы будем создавать новый private key.

00:13:07.480 --> 00:13:12.080

Давайте его назовем vockey2.

00:13:12.080 --> 00:13:18.640

Мы добрались до восьмого задания и это последний шаг перед тем, как мы начнем все нами созданное тестировать.

00:13:18.640 --> 00:13:35.240

Здесь нам необходимо настроить возможность подключения к bastion host, а также из bastion host к EC2 instance в private subnet, не сохраняя приватный ключ в bastion host.

00:13:35.240 --> 00:13:37.280

Давайте перефразирую.

00:13:37.280 --> 00:13:49.960

Когда мы подключаемся через интернет по протоколу SSH к любому удаленному серверу, мы локально у себя на компьютере храним приватный ключ.

00:13:49.960 --> 00:14:05.400

Когда мы подключаемся через PuTTY, мы указываем в настройках программы, где находится этот ключ и во время подключения мы указываем публичный IP-адрес и вместе с ключом мы успешно подключаемся к этому инстансу.

00:14:05.400 --> 00:14:21.720

Теперь следующий шаг, когда мы подключились к этому инстансу, в нашем случае это bastion host, нам необходимо еще раз дальше подключиться также по SSH к другому EC2 инстансу, но этот EC2 инстанс находится в private subnet.

00:14:21.720 --> 00:14:32.240

Поэтому чтобы это сделать, нам необходимо чтобы в bastion host у нас был приватный ключ для подключения ко второму EC2 инстансу.

00:14:32.240 --> 00:14:46.400

Это не очень хорошо и небезопасно, так как все пользователи, которые будут подключаться к bastion host под тем же пользователем, они будут возможно видеть ключи, которые они не должны были видеть.

00:14:46.400 --> 00:14:55.960

Поэтому это не самый хороший подход с точки зрения безопасности и есть готовое решение, так называемый SSH passthrough.

00:14:55.960 --> 00:15:21.400

Идея в том, что вы храните локально все необходимые ключи и при подключении последовательно к нескольким инстансам по SSH, эти ключи ищатся не из инстанса, откуда вы подключаетесь, а они подтягиваются из локального компьютера, где было инициировано самое первое SSH соединение.

00:15:21.400 --> 00:15:25.720

Что для этого необходимо сделать?

00:15:25.720 --> 00:15:33.320

Для пользователей операционной системы Windows необходимо перейти по ссылке и скачать программу Pageant.

00:15:33.320 --> 00:15:38.640

Эта программа также разрабатывается командой, которая создала программу PuTTY.

00:15:38.640 --> 00:15:43.760

Как только мы ее загрузили, необходимо будет ее установить и запустить.

00:15:43.760 --> 00:15:59.960

Как только вы ее запустите, там достаточно простой интерфейс и необходимо добавить два наших ключа vockey от Bastion Host и vockey2 от инстанса из private subnet в эту программу.

00:15:59.960 --> 00:16:07.400

Обратите внимание, что ваши private ключи должны быть с расширением .PPK.

00:16:07.400 --> 00:16:25.320

Далее при подключении через PuTTY необходимо будет сделать дополнительную настройку, а именно перейти по пути настройки Connection &gt; SSH &gt; Auth и для признака Allow agent forwarding поставить галочку.

00:16:25.320 --> 00:16:37.040

Также не забудьте в той же секции заполнить секцию Credentials, а именно поле Private key file for authentication.

00:16:37.040 --> 00:16:41.480

Здесь нам необходимо выбрать private key для подключения к bastion host.

00:16:41.480 --> 00:16:45.680

В нашем случае это vockey.

00:16:45.680 --> 00:16:58.320

Как только все мы это сделаем, нам необходимо использовать публичный IP-адрес нашего bastion host и уже можем пробовать подключаться.

00:16:58.320 --> 00:17:08.120

Для пользователей с операционной системой macOS или Linux необходимо будет ввести последовательно следующие команды, которые вы видите в задании.

00:17:08.120 --> 00:17:28.880

Идея такая же, вы сохраняете ключи в определенном месте в настройках и далее при подключении с дополнительным признаком создаете SSH соединение к bastion host.

00:17:28.880 --> 00:17:32.960

Теперь нам необходимо подключиться к bastion host.

00:17:32.960 --> 00:17:43.360

Как только мы подключились, от bastion host нам необходимо подключиться к EC2инстансу в private subnet.

00:17:43.360 --> 00:17:48.120

Как это сделать вы уже знаете, поэтому подробнее останавливаться здесь не буду.

00:17:48.120 --> 00:17:55.760

Как только мы успешно подключились к EC2инстансу в private subnet, давайте проверим есть ли у этого инстанса выход в интернет.

00:17:55.760 --> 00:18:18.080

Для этого можете вести простейшую команду ping и IP-адрес со всеми восьмерками, в случае если вы увидите обмен пакетами и этот обмен пакетами успешен, то это говорит о том, что выход в интернет у этого инстанса есть и мы все настройки ввели корректно.

00:18:18.080 --> 00:18:30.360

Мы с вами добрались до второго блока нашей лабораторной работы и здесь мы будем создавать дополнительные уровни безопасности.

00:18:30.360 --> 00:18:56.440

Согласно бизнес-кейсу, София и Нихил рассказали свои обновления в облачной архитектуре другу Матео, он был впечатлен и после того как выслушал все обновления, дополнительно порекомендовал использовать Network ACL, чтобы еще больше обезопасить IT-архитектуру в облаке AWS, что мы с вами сейчас и сделаем.

00:18:56.440 --> 00:19:00.120

Самым первым нам необходимо создать Network ACL.

00:19:00.120 --> 00:19:12.800

Этот Network ACL является кастомным, мы с вами помним, что дефолтный Network ACL разрешает весь входящий и исходящий трафик и по умолчанию привязывается к subnet-ам.

00:19:12.800 --> 00:19:28.440

В случае, когда мы создаем Custom Network ACL, то у него по умолчанию доступ входящий и исходящий закрыт и нам необходимо создавать дополнительные правила, чтобы доступ разрешать.

00:19:28.440 --> 00:19:42.680

Нам необходимо назвать наш Custom Network ACL Lab Network ACL и привязать ее к нашему private subnet.

00:19:42.680 --> 00:19:57.240

Далее нам необходимо для этого Custom Subnet разрешить весь входящий и исходящий трафик и далее уже на 11 задании мы будем его тестировать.

00:19:57.240 --> 00:20:08.960

Чтобы его протестировать, нам необходимо в public subnet создать тестовыйEC2 Instance со следующими параметрами.

00:20:08.960 --> 00:20:29.040

Как AMI использовать Amazon Linux 2 AMI (HVM), как Instance Type у нас будет t2.micro, название будет Test Instance и в Security группе этого инстанса необходимо разрешить весь входящий ICMP IPv4 трафик.

00:20:29.040 --> 00:20:37.440

Это как раз таки тот трафик, который приходит после запуска команды ping в командной строке.

00:20:37.440 --> 00:21:13.280

Нам необходимо сохранить приватный IP адрес этого тестового Instance, как только он создастся и мы возвращаемся к командной строке, где у нас текущее подключение до приватного Instance в private subnet и мы здесь будем пробовать через ping обратиться к приватному IP адресу тестового Instance.

00:21:13.280 --> 00:21:33.160

Вы увидите, что ping проходит, пакеты обмениваются и это говорит о том, что приватный инстанс может обратиться к нашему тестовому инстансу в public subnet.

00:21:33.160 --> 00:21:43.600

Теперь нам необходимо произвести изменения в нашем кастомном Network ACL, напоминаю, что этот Network ACL привязан к private subnet.

00:21:43.600 --> 00:21:51.320

Здесь нам необходимо запретить весь ICMP IPv4 трафик для конкретного IP адреса.

00:21:51.320 --> 00:21:59.640

Когда нам нужно указать один IP адрес, то используется CIDR блок размера 32, который в себе содержит один IP адрес.

00:21:59.640 --> 00:22:05.400

Обратите внимание, что очередность этого правила должна быть выше, чем другие правила, чтобы оно сработало.

00:22:05.400 --> 00:22:19.960

В Terminal, где у нас отрабатывает команда ping, давайте его не будем закрывать и пусть оно отправляет пакеты.

00:22:19.960 --> 00:22:27.800

Как только мы сохраним изменения в Network ACL, вы увидите, что пакеты перестали обмениваться.

00:22:27.800 --> 00:22:36.800

Это говорит о том, что мы запретили доступ через Network ACL и добавили дополнительный уровень безопасности.

00:22:36.800 --> 00:23:00.360

Вы увидите, что у нас появился кастомный Network ACL, который запрещает доступ специально для тестового инстанса в public subnet.

00:23:00.360 --> 00:23:09.800

После того, как это все сработало корректно, это говорит о том, что вы все настройки ввели правильно, мы добрались до самого последнего блока заданий.

00:23:09.800 --> 00:23:26.040

И здесь необходимо на основной странице лабораторной работы нажать на кнопку Details, далее в выпадающем меню нажать на ссылку Access the multiple choice questions.

00:23:26.040 --> 00:23:27.800

Там содержится 6 вопросов.

00:23:27.800 --> 00:23:31.480

Давайте пройдемся по каждому из вопросов.

00:23:31.480 --> 00:23:33.840

Первый вопрос звучит следующим образом.

00:23:33.840 --> 00:23:47.760

Какая цель у Internet gateway в public subnet?

00:23:47.760 --> 00:23:48.760

Далее второй вопрос.

00:23:48.760 --> 00:24:01.480

Что дает возможность инстансу из private subnet подключаться в интернет и скачивать обновления?

00:24:01.480 --> 00:24:02.480

Третий вопрос.

00:24:02.480 --> 00:24:08.960

Может ли инстанс из private subnet напрямую подключаться в интернет?

00:24:08.960 --> 00:24:10.400

Четвертый вопрос.

00:24:10.400 --> 00:24:21.640

Почему мы создали два отдельных ключа для инстанса из private subnet и для инстанса из public subnet, а именно для bastion host?

00:24:21.640 --> 00:24:22.640

Пятый вопрос.

00:24:22.640 --> 00:24:34.520

Может ли bastion host успешно запустить команду ping и получить ответ от инстанса из private subnet?

00:24:34.520 --> 00:24:36.480

Это очень хороший вопрос.

00:24:36.480 --> 00:24:37.480

Шестой вопрос.

00:24:37.480 --> 00:25:06.400

Какое правило в security group инстанса в private subnet дает разрешение входящему трафику команды ping, который мы отправили из этого инстанса тестовому инстансу в public subnet?

00:25:06.400 --> 00:25:21.120

На странице с вопросами у вас будут варианты, вы можете посмотреть на них и будут заведомо неверные варианты ответов и методами исключения попытаться ответить правильно.

00:25:21.120 --> 00:25:37.880

Обратите внимание, что сделанные действия в рамках AWS Management Console, что вопросы тестовые, они могут быть пересданы любое количество раз, поэтому экспериментируйте.

00:25:37.880 --> 00:25:49.680

Чтобы сдать задание, необходимо на главной странице лабораторной работы нажать на кнопку Submit и подтвердить то, что вы запускаете скрипт автооценивания.

00:25:49.680 --> 00:26:00.040

Так как лабораторная работа объемная, необходимо будет некоторое время подождать, когда оно успешно отработает и в правой части страницы вы увидите набранные вами баллы.

00:26:00.040 --> 00:26:15.760

Если вы не добрали максимальный балл, то вы всегда можете посмотреть на каком шаге, в каком задании у вас не хватает баллов, вернуться к этому заданию и попробовать его сделать еще раз.

00:26:15.760 --> 00:26:28.360

Как только вы внесете необходимые изменения, вы можете еще раз запустить скрипт автооценивания и обновить свои баллы в рамках лабораторной работы.

00:26:28.360 --> 00:26:34.400

Как только вы закончили работу, вам необходимо правильно выйти со всех систем.

00:26:34.400 --> 00:26:40.620

В первую очередь это касается AWS Management Console, необходимо будет разлогиниться.

00:26:40.620 --> 00:26:53.600

Далее на главной странице лабораторной работы необходимо будет нажать на кнопку End lab, подтвердить, что вы завершаете лабораторную работу и уже после этого закрывать страницу с лабораторной работы.

00:26:53.600 --> 00:26:59.080

На этом мы завершаем сессию обзора лабораторной работы.

00:26:59.080 --> 00:27:03.200

Это очень интересная лабораторная работа, очень надеюсь, что она вам понравится.

00:27:03.200 --> 00:27:05.760

Если будут вопросы, задавайте.

00:27:05.760 --> 00:27:11.960

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