WEBVTT

00:00:00.000 --> 00:00:07.000

Добрый день, уважаемые студенты! Я рад вас всех видеть на очередной сессии разбора лабораторной работы.

00:00:07.000 --> 00:00:16.000

Тема сегодняшней лабораторной работы - это создание VPC и всех ее компонентов в рамках нашего бизнес-кейса с кафе.

00:00:16.000 --> 00:00:18.000

Итак, давайте начнем.

00:00:21.000 --> 00:00:26.000

Начинаем разбор лабораторной работы с основной страницы AWS Management Console.

00:00:26.000 --> 00:00:35.000

Здесь нам необходимо перейти в сервис VPC. Для этого необходимо воспользоваться строкой поиска сервисов и начать вводить VPC.

00:00:35.000 --> 00:00:38.000

Самая первая ссылка - это та ссылка, которая нам нужна.

00:00:38.000 --> 00:00:48.000

На основной странице сервиса VPC нам необходимо воспользоваться навигационным меню слева и перейти в раздел существующих VPC.

00:00:48.000 --> 00:00:59.000

Здесь мы видим, что для нас уже в рамках лабораторной работы была создана Lab VPC, на базе которой мы будем создавать оставшиеся компоненты.

00:00:59.000 --> 00:01:08.000

Первым делом необходимо создать subnets. Для этого в левом навигационном меню необходимо выбрать пункт Subnets.

00:01:08.000 --> 00:01:15.000

Здесь необходимо в открывшейся странице вверху справа нажать на кнопку Create Subnet.

00:01:15.000 --> 00:01:22.000

В самом начале необходимо будет выбрать VPC, в нашем случае Lab VPC.

00:01:22.000 --> 00:01:27.000

Далее необходимо будет ввести Subnet name, в нашем случае это Public Subnet.

00:01:27.000 --> 00:01:40.000

Чуть ниже необходимо будет выбрать Availability Zone. В рамках лабораторной работы должна быть первая Availability Zone и это us-east-1a.

00:01:40.000 --> 00:01:51.000

И как CIDR блок для нашего subnet нам необходимо скопировать значение с задания лабораторной работы и ввести ее в соответствующее поле.

00:01:51.000 --> 00:01:57.000

После чего необходимо прокрутить до конца страницы и нажать на кнопку Create Subnet.

00:01:57.000 --> 00:02:09.000

Как только мы нажмем на кнопку Create Subnet, нас направят на страницу со списком всех subnets и отфильтрованные по нашему созданному subnet.

00:02:09.000 --> 00:02:15.000

Теперь нам необходимо этот Public Subnet сделать фактически Public.

00:02:15.000 --> 00:02:21.000

Для этого необходимо в левом навигационном меню перейти в пункт Internet Gateways.

00:02:21.000 --> 00:02:29.000

Здесь мы также создаем новый ресурс, для этого в верхней правой части нажимаем на кнопку Create Internet Gateway.

00:02:29.000 --> 00:02:38.000

Во время создания Internet gateway много входных данных вводить не нужно, достаточно ввести имя этого ресурса, в нашем случае это Lab IGW.

00:02:38.000 --> 00:02:41.000

То есть сокращенно от Internet Gateway.

00:02:41.000 --> 00:02:47.000

И после чего можем в нижней части страницы нажать на кнопку Create Internet Gateway.

00:02:47.000 --> 00:02:57.000

Как только мы это сделаем, нас направят на основную страницу ресурса, в нашем случае Internet Gateway, и вы увидите, что состояние Detached.

00:02:57.000 --> 00:03:03.000

Также в уведомлении зеленым цветом вы увидите кнопку Attach to VPC.

00:03:03.000 --> 00:03:12.000

Это значит, что нам необходимо этот Internet Gateway привязать к одному из VPC, в нашем случае это Lab VPC.

00:03:12.000 --> 00:03:21.000

Давайте нажмем на эту кнопку, нас направят на страницу настройки, здесь необходимо выбрать VPC, в нашем случае Lab VPC.

00:03:21.000 --> 00:03:27.000

Выбираем ее и в нижней части страницы нажимаем на кнопку Attach Internet Gateway.

00:03:27.000 --> 00:03:35.000

Как только мы это сделаем, вы увидите соответствующее сообщение, что Internet Gateway был успешно привязан к этому VPC.

00:03:35.000 --> 00:03:44.000

Также вы увидите, что состояние поменялось на Attached, и в поле VPC ID стоит ссылка на наш Lab VPC.

00:03:44.000 --> 00:03:55.000

Теперь нам необходимо настроить Route Table, для этого воспользуемся навигационным меню слева и выберем пункт Route Tables.

00:03:55.000 --> 00:04:12.000

Здесь, как мы знаем, при создании VPC создается дефолтовый Route Table, мы можем воспользоваться ею и создать необходимые Routes для нашего Public Subnet.

00:04:12.000 --> 00:04:20.000

В вашем случае может отобразиться несколько Route Tables, вы можете найти правильные, посмотрев на столбец VPC.

00:04:20.000 --> 00:04:41.000

Там будет VPC ID и через разделитель название VPC. Давайте ее выберем и в открывшемся блоке страницы с данными этого ресурса необходимо выбрать вкладку Routes и далее нажать на кнопку Edit Routes.

00:04:41.000 --> 00:04:51.000

На следующей странице нам необходимо будет как Destination указать весь интернет и через интернет мы будем выходить через наш Internet Gateway.

00:04:51.000 --> 00:04:57.000

Как только мы заполним необходимые поля, мы можем сохранить изменения.

00:04:57.000 --> 00:05:12.000

Как только мы это сделаем, мы увидим соответствующее сообщение, теперь если мы перейдем на страницу subnets, то увидим наш Public Subnet и все необходимые компоненты для него были созданы.

00:05:12.000 --> 00:05:30.000

И он не только в названии является Public, а также фактически является Public Subnet, ресурсы созданные в рамках этого Subnet могут выходить в интернет, также из интернета можно обращаться к ресурсам внутри этого Subnet.

00:05:30.000 --> 00:05:37.000

Следующим шагом нам необходимо создать Bastion Host. Для этого воспользуемся сервисом EC2.

00:05:37.000 --> 00:05:42.000

Для этого необходимо будет в строке поиска сервисов начать вводить EC2 и перейти по первой ссылке.

00:05:42.000 --> 00:05:51.000

Как только мы окажемся на основной странице сервиса EC2, необходимо будет нажать на кнопку Launch Instance и нас направит на страницу ввода входных данных.

00:05:51.000 --> 00:06:04.000

Здесь как Name введем Bastion Host, далее как операционную систему выберем Amazon Linux и как AMI выберем Amazon Linux 2 AMI (HVM).

00:06:04.000 --> 00:06:12.000

Прокрутим чуть ниже, как Instance Type необходимо будет выбрать t2.micro, если этого варианта нет, выберем t3.micro.

00:06:12.000 --> 00:06:20.000

Как Key Pair выберем vockey и в секции Network Settings необходимо будет нажать на кнопку Edit.

00:06:20.000 --> 00:06:22.000

Мы будем вносить некоторые изменения.

00:06:22.000 --> 00:06:29.000

Как VPC необходимо будет выбрать Lab VPC, как Subnet выбираем Public Subnet, нами созданный.

00:06:29.000 --> 00:06:40.000

Для поля Auto-assign Public IP необходимо будет его отключить, так как мы будем создавать Elastic IP Address и привязывать к этому инстансу.

00:06:40.000 --> 00:06:47.000

Далее необходимо будет выбрать опцию Create Security Group, как имя ввести Bastion Host SG.

00:06:47.000 --> 00:06:59.000

После чего необходимо будет нажать на кнопку Add Security Group Rule в секции Inbound Security Group Rules и добавить следующее правило.

00:06:59.000 --> 00:07:13.000

Мы разрешаем SSH трафик, как Source указываем MyIP, вводится мой публичный IP адрес с размером CIDR блока 32, в котором находится один единственный IP адрес.

00:07:13.000 --> 00:07:21.000

Как Description необходимо будет указать некоторые значения, чтобы вы в будущем могли идентифицировать для чего оно было создано.

00:07:21.000 --> 00:07:25.000

В моем случае это SSH и Yelzhan, имя пользователя.

00:07:25.000 --> 00:07:34.000

Как только мы введем все эти настройки, мы можем в правой части страницы нажать на кнопку Launch Instance.

00:07:34.000 --> 00:07:55.000

Нас направит на страницу с сообщением, что создание инстанса было инициировано, мы можем нажать на ссылку с ID инстанса и перейти на страницу со списком инстансов и отфильтрованным по нашему инстансу.

00:07:55.000 --> 00:08:11.000

Через некоторое время состояние инстанса перейдет в Running и пока это происходит, мы вне зависимости от этого можем продолжить создание следующих ресурсов, это Elastic IP address.

00:08:11.000 --> 00:08:17.000

Для этого в левом навигационном меню необходимо выбрать опцию Elastic IPs.

00:08:17.000 --> 00:08:31.000

Здесь необходимо будет нажать на кнопку Allocate Elastic IP Address и на странице для ввода входных данных выбрать опцию Amazon's Pool of IPv4 Addresses.

00:08:31.000 --> 00:08:40.000

Все остальные настройки оставляем по умолчанию, необходимо прокрутить до конца страницы и нажать на кнопку Allocate.

00:08:40.000 --> 00:08:59.000

Как только мы это сделаем, нас направят на страницу со списком Elastic IP address, вы также увидите в сообщении, что необходимо этот Elastic IP address ассоциировать с ресурсом, что мы и сделаем.

00:08:59.000 --> 00:09:09.000

Один из вариантов это нажать на кнопку Associate this Elastic IP address вверху страницы, либо выбрать этот Elastic IP address.

00:09:09.000 --> 00:09:18.000

Далее нажать на кнопку Actions и в выпадающем меню выбрать опцию Associate Elastic IP address. Он направит нас на ту же страницу.

00:09:18.000 --> 00:09:25.000

На странице настроек необходимо будет выбрать, к какому ресурсу мы привязываем это либо Network Interface, либо Instance.

00:09:25.000 --> 00:09:38.000

В нашем случае это Instance. Как только мы ее выберем, необходимо будет в поле ниже, где Instance, необходимо будет выбрать наш EC2 Instance с именем Bastion Host.

00:09:38.000 --> 00:09:45.000

Как только мы это сделаем, можем прокрутить до конца страницы и нажать на кнопку Associate.

00:09:45.000 --> 00:09:57.000

Как только мы это проделаем, мы получим сообщение о том, что Elastic IP address был успешно привязан к ресурсу, к инстансу с таким-то ID.

00:09:57.000 --> 00:10:10.000

Это значит, что наш EC2 инстанс помимо того, что может выходить в интернет, также из интернета он доступен и к нему можно обратиться по статическому IP Адресу.

00:10:10.000 --> 00:10:19.000

Этот IP Адрес вы видите на нашем экране.

00:10:19.000 --> 00:10:27.000

Давайте теперь проверим, действительно ли мы можем подключаться к нашему EC2 инстансу в Public Subnet.

00:10:27.000 --> 00:10:39.000

Для этого необходимо скачать private key. Мы возвращаемся на страницу с лабораторной работы, нажимаем на кнопку Details и в выпадающем списке нажимаем на кнопку Show.

00:10:39.000 --> 00:10:55.000

Во всплывающем окне необходимо будет скачать PPK ключ в случае, если это Windows операционная система, в случае, если вы работаете с Linux операционной системой или MacOS, необходимо будет скачать PEM ключ.

00:10:55.000 --> 00:11:03.000

Как только вы это сделаете, в случае с Windows операционной системой вам необходимо будет скачать программу PuTTY.

00:11:03.000 --> 00:11:14.000

Мы неоднократно работали с этой программой на предыдущих наших лабораторных работах, поэтому она скорее всего уже установлена, необходимо будет ее запустить.

00:11:14.000 --> 00:11:24.000

Также мы с вами ранее уже подключались с использованием PPK ключа к другим инстансам.

00:11:24.000 --> 00:11:38.000

И если у вас есть сохраненная сессия, вы можете ее подгрузить, после чего ввести в поле для ввода IP адреса IP адрес нашего Elastic IP ресурса.

00:11:38.000 --> 00:12:00.000

Если же у вас созданной сессии нет, вы можете в левом навигационном меню программы PuTTY пройти по пути Connection, далее SSH, после чего Auth и Credentials, и там для соответствующего поля указать путь приватного ключа.

00:12:00.000 --> 00:12:10.000

После чего вы можете вернуться обратно на вкладку Session, самая первая вкладка, и здесь ввести публичный IP адрес нашего EC2 инстанса.

00:12:10.000 --> 00:12:22.000

Как только мы введем все необходимые настройки, мы можем нажать на кнопку Open. Откроется командная строка и вы увидите подобное сообщение, если вы впервые подключаетесь к этому инстансу.

00:12:22.000 --> 00:12:42.000

Так как это нами созданный инстанс и мы доверяем ему, мы можем нажать на кнопку Accept. При всех других случаях необходимо удостовериться, что вы подключаетесь к доверенному инстансу, и только после этого вы подтверждаете ваше подключение.

00:12:42.000 --> 00:13:02.000

Как только нажмем на кнопку Accept, необходимо будет указать пользователя, дефолтовые пользователи для Amazon Linux, это ec2-user. Как только мы это введем и нажмем на кнопку Enter, мы увидим сообщение о том, что мы успешно авторизованы и увидим промпт для ввода команд.

00:13:02.000 --> 00:13:18.000

Это значит, что мы успешно смогли удаленно подключиться к нашему EC2 инстансу в public subnet, используя его публичный IP адрес. Отлично, я вас поздравляю, мы завершили это задание и переходим к следующему.

00:13:18.000 --> 00:13:32.000

В рамках следующего задания нам необходимо создать private subnet и все необходимые для него компоненты. Для этого мы в строке поиска сервисов начнем вводить VPC и перейдем к соответствующему сервису.

00:13:32.000 --> 00:13:48.000

Далее воспользуемся левым навигационным меню и выберем пункт Subnets. Здесь необходимо будет создать новый subnet, для этого нажмем на кнопку Create subnet и в открывшейся странице для ввода входных данных начнем эти данные вводить.

00:13:48.000 --> 00:14:04.000

Как VPC ID мы выбираем Lab VPC и как Subnet Name необходимо будет написать Private subnet. Прокрутим чуть ниже, как Availability zone выбираем первую availability zone в этом регионе, в нашем случае это us-east-1a.

00:14:04.000 --> 00:14:24.000

И как значение CIDR block необходимо будет скопировать со страницы задания лабораторной работы, в нашем случае это 10.0.1.0/24. Как только мы введем все эти данные, мы можем прокрутить до конца страницы и нажать на кнопку Create subnet.

00:14:24.000 --> 00:14:37.000

Мы видим, что наш subnet был успешно создан, теперь мы создаем следующие компоненты, для этого воспользуемся левым навигационным меню и перейдем к пункту NAT gateways.

00:14:37.000 --> 00:14:50.000

У нас созданных NAT gateways не имеется, поэтому нажмем на кнопку Create NAT gateway. Здесь в открывшейся странице для ввода входных данных как Name введем Lab NAT Gateway.

00:14:50.000 --> 00:15:01.000

Далее, как Subnet выберем Public subnet, то есть наш NAT Gateway будет находиться в public subnet и будет давать возможность ресурсам из private subnet подключаться к интернету.

00:15:01.000 --> 00:15:12.000

Прокрутим чуть ниже, и мы с вами помним, что при создании NAT Gateway необходимо его ассоциировать с публичным статическим IP адресом.

00:15:12.000 --> 00:15:29.000

И здесь есть возможность путем нажатия на кнопки Allocate IP address создать ресурс Elastic IP и так как мы ее создаем в рамках создания NAT Gateway, то ID Elastic IP впишется в соответствующее поле.

00:15:29.000 --> 00:15:36.000

После чего нам необходимо будет прокрутить до конца страницы и нажать на кнопку Create NAT Gateway.

00:15:36.000 --> 00:15:46.000

Как только мы создадим NAT Gateway, останется буквально последний шаг, это работа с route table.

00:15:46.000 --> 00:15:52.000

Необходимо будет воспользоваться левым навигационным меню и перейти к пункту Routes tables.

00:15:52.000 --> 00:16:01.000

Ранее мы с вами использовали дефолтовый route table для привязки ее к public subnet.

00:16:01.000 --> 00:16:08.000

Сейчас нам необходимо создать отдельный route table для того чтобы привязать его к private subnet.

00:16:08.000 --> 00:16:20.000

Итак, давайте нажмем на кнопку Create route table, в открывшейся странице для ввода входных данных нам необходимо ввести имя Private route table и как VPC выбрать Lab VPC.

00:16:20.000 --> 00:16:23.000

После чего мы можем нажать на кнопку Create route table.

00:16:23.000 --> 00:16:31.000

Наш route table создастся и мы перейдем на основную страницу ресурса.

00:16:31.000 --> 00:16:40.000

Здесь во вкладке Routes увидим, что у нас только один единственный дефолтовый rule, который помогает общаться с ресурсом внутри VPC.

00:16:40.000 --> 00:16:48.000

Нам необходимо нажать на кнопку Edit routes и здесь необходимо будет добавить новый route.

00:16:48.000 --> 00:16:54.000

Как Destination необходимо указать обозначение интернета, это все нули слеш ноль.

00:16:54.000 --> 00:17:04.000

Далее, как Target указать NAT gateway, то есть ресурсы находящиеся в private subnet будут подключаться к интернету через NAT gateway.

00:17:04.000 --> 00:17:06.000

А NAT gateway у нас находится в public subnet.

00:17:06.000 --> 00:17:13.000

Как только мы заполним эти два поля, мы можем нажать на кнопку Save changes.

00:17:13.000 --> 00:17:18.000

Отлично, на этом мы почти подготовили наш route table.

00:17:18.000 --> 00:17:29.000

Не забывайте, что route table это независимый отдельный ресурс и если вы работаете с кастомным route table необходимо явно привязать его к subnet.

00:17:29.000 --> 00:17:47.000

Ранее нам не пришлось привязывать route table к subnet, так как это был дефолтовый route table и дефолтовый route table он привязывается ко всем subnet, которые создаются в рамках VPC.

00:17:47.000 --> 00:17:57.000

В нашем случае мы создали кастомный route table и здесь необходимо привязать его явно к нашему private subnet.

00:17:57.000 --> 00:18:06.000

Для этого необходимо перейти во вкладку Subnet associations и здесь нажать на кнопку Edit subnet associations.

00:18:06.000 --> 00:18:15.000

Нас направят на страницу с subnet, здесь отображаются все subnets доступные для нас в рамках VPC.

00:18:15.000 --> 00:18:18.000

Вы увидите, что есть public subnet и private subnet.

00:18:18.000 --> 00:18:28.000

Оба они привязаны к дефолтовому route table, вы это видите в соответствующем столбце, это последний столбец в рамках этой небольшой таблицы.

00:18:28.000 --> 00:18:37.000

На этой странице нам необходимо будет выбрать Private subnet и нажать на кнопку Save associations.

00:18:37.000 --> 00:18:38.000

Что произойдет?

00:18:38.000 --> 00:18:50.000

Мы с вами помним, что subnet может быть ассоциирован только с одним route table в один момент времени, тогда как один route table может использоваться в нескольких subnet.

00:18:50.000 --> 00:19:07.000

Поэтому привязка private subnet к дефолтовому route table удалится и создается новая связка private subnet с кастомным route table.

00:19:07.000 --> 00:19:14.000

Отлично, как только мы это сделали, мы можем в списке Subnet associations увидеть наш private subnet.

00:19:14.000 --> 00:19:28.000

Также увидим соответствующее сообщение о том, что route table с таким то ID был успешно привязан к нашему private subnet.

00:19:28.000 --> 00:19:42.000

Мы произвели все необходимые настройки, теперь необходимо проверить все ли корректно работает и есть ли возможность подключаться из интернета к ресурсам, находящимся в private subnet.

00:19:42.000 --> 00:19:56.000

Для этого необходимо создать инстанс, переходим к сервису EC2, далее на основной странице сервиса EC2 нажимаем на кнопку Launch instance и начинаем вводить входные данные.

00:19:56.000 --> 00:20:07.000

Как Name указываем Private instance, далее выбираем как операционную систему Amazon Linux, как AMI выбираем Amazon Linux 2 AMI (HVM).

00:20:07.000 --> 00:20:17.000

Как Instance Type выбираем t2.micro и в секции приватного ключа нам необходимо будет создать новый.

00:20:17.000 --> 00:20:32.000

Для этого нажимаем на кнопку Create new key pair указываем отличное от предыдущего приватного ключа имя, пусть это будет vockey2 и необходимо будет выбрать правильный формат.

00:20:32.000 --> 00:20:41.000

Напоминаю, что если вы работаете с Windows операционной системой вам необходимо будет выбрать формат PPK, он подходит для программы PuTTY.

00:20:41.000 --> 00:20:49.000

Если же вы работаете с MacOS либо с Linux операционной системой вам необходимо выбрать формат PEM.

00:20:49.000 --> 00:20:55.000

Отлично, выбрали формат, ввели имя, теперь необходимо нажать на кнопку Create key pair.

00:20:55.000 --> 00:21:03.000

Как только вы ее нажмете, вам будет предложено сохранить приватный ключ на локальном компьютере.

00:21:03.000 --> 00:21:09.000

Это единственный кейс, когда вы этот приватный ключ можете скачать, поэтому сохраняем ее.

00:21:09.000 --> 00:21:24.000

Далее, в следующей секции настройках сети необходимо будет выбрать Lab VPC, как Subnet выбираем Private subnet и чуть ниже нам необходимо будет выбрать опцию Create security group.

00:21:24.000 --> 00:21:33.000

Мы создаем новую security группу, нам необходимо будет в секции Inbound security group rules добавить новый rule.

00:21:33.000 --> 00:21:43.000

Здесь мы указываем SSH traffic, как Source type указываем Custom и в Source указываем security группу нашего bastion host.

00:21:43.000 --> 00:21:57.000

Это значит, что только subnets, к которым привязана эта security группа, может быть использована как инициатор подключения по SSH.

00:21:57.000 --> 00:22:03.000

Только ресурсы из public subnet смогут подключаться к этому инстансу.

00:22:03.000 --> 00:22:11.000

Как Description давайте укажем Bastion host SG, чтобы легче было понимать для чего этот rule был создан.

00:22:11.000 --> 00:22:19.000

Отлично, мы ввели все необходимые настройки, теперь мы можем в правой части экрана нажать на кнопку Launch instance.

00:22:19.000 --> 00:22:27.000

Пока наш инстанс будет создаваться, давайте попробуем ответить на следующий вопрос.

00:22:27.000 --> 00:22:35.000

У нас есть созданный bastion host в public subnet, к нему мы уже можем подключаться, мы это проверили.

00:22:35.000 --> 00:22:42.000

Теперь мы в private subnet создаем другой инстанс и идея в том, чтобы подключаться к нему.

00:22:42.000 --> 00:22:48.000

Мы создали bastion host специально для того чтобы подключаться к ресурсам в private subnet, это хорошо.

00:22:48.000 --> 00:22:58.000

Но представьте, как это будет выглядеть, для того чтобы подключиться к bastion host, мы используем приватный ключ сохраненный локально на нашем компьютере.

00:22:58.000 --> 00:23:02.000

И с локального компьютера инициируем SSH соединение.

00:23:02.000 --> 00:23:28.000

Теперь, для того чтобы с bastion host инициировать SSH соединение к инстансу из private subnet, нам также на bastion host-е, то есть локально с того места, где мы инициируем SSH соединение, необходимо, чтобы был приватный ключ, а именно тот ключ, который мы создали и после чего у нас получится подключиться к приватному инстансу.

00:23:28.000 --> 00:23:40.000

Это не самый лучший вариант с точки зрения безопасности, так как bastion host это то место, где все сотрудники подключаются к нему, а дальше от него подключаются к необходимым ресурсам.

00:23:40.000 --> 00:23:50.000

Так как в компании могут работать различные команды, то и targets, куда они в конечном счете будут подключаться, могут отличаться.

00:23:50.000 --> 00:24:02.000

И эти места используют различные приватные ключи. Если вы в bastion host сохраните ваш приватный ключ, то другие пользователи могут воспользоваться этим ключом и не авторизованно его использовать.

00:24:02.000 --> 00:24:18.000

Поэтому как решение командой разработки программы PuTTY, была создана еще одна дополнительная программа, это больше как плагин к программе PuTTY называется Pageant.

00:24:18.000 --> 00:24:33.000

Идея ее в том, что вы можете, сохранив все ключи локально в месте, где инициируется SSH подключение, подключаться к инстансам последовательно.

00:24:33.000 --> 00:24:44.000

То есть мы можем локально с нашего компьютера, сохранив приватные ключи только на этом компьютере, через bastion host подключиться к инстансу в private subnet.

00:24:44.000 --> 00:24:54.000

Таким образом, используя эту программу, нам нет необходимости приватный ключ от private instance хранить в bastion host.

00:24:54.000 --> 00:25:03.000

Согласитесь, это очень удобная и хорошая программа. Давайте теперь посмотрим как она работает и как ее необходимо настроить.

00:25:03.000 --> 00:25:14.000

Первым делом необходимо ее скачать, устанавливать ее не нужно, как только вы запускаете exe, запускается сама программа и здесь необходимо будет нажать на кнопку Add key.

00:25:14.000 --> 00:25:23.000

Далее, необходимо будет выбрать оба наших ключа и добиться того, чтобы они отобразились в списке приватных ключей.

00:25:23.000 --> 00:25:29.000

После этого мы произвели все необходимые настройки в программе Pageant, можем нажать на кнопку Close.

00:25:29.000 --> 00:25:34.000

Далее, уже на стороне PuTTY необходимо активировать соответствующий функционал.

00:25:34.000 --> 00:25:49.000

А именно в левом навигационном меню необходимо выбрать страницу Auth и на самой странице необходимо будет поставить галочку для опции Allow agent forwarding.

00:25:49.000 --> 00:26:00.000

Как только вы это сделаете, вам далее останется выбрать приватный ключ от bastion host, это на странице Credentials.

00:26:00.000 --> 00:26:07.000

Самое первое поле необходимо будет нажать на кнопку Browse и указать путь для нашего PPK ключа.

00:26:07.000 --> 00:26:16.000

После этого необходимо будет нам скопировать публичный IP адрес от bastion host, это мы можем сделать двумя путями.

00:26:16.000 --> 00:26:25.000

Первый - это в списке EC2 инстансов, если выбрать наш bastion host, то в метаданных есть соответствующее поле.

00:26:25.000 --> 00:26:32.000

Также мы можем подсмотреть публичный IP адрес в списке Elastic IP адресов.

00:26:32.000 --> 00:26:41.000

Как только мы скопировали IP адрес, необходимо его вставить в PuTTY и нажать на кнопку Open.

00:26:41.000 --> 00:26:56.000

Здесь необходимо будет ввести название пользователя в нашем случае это ec2-user нажимаем на Enter и вы увидите промпт, где можно будет вводить команды, это означает то что мы успешно подключились к нашему bastion host.

00:26:56.000 --> 00:27:00.000

Здесь отображается приватный IP адрес.

00:27:00.000 --> 00:27:06.000

Теперь нам необходимо попробовать подключиться к инстансу в private subnet.

00:27:06.000 --> 00:27:14.000

Для этого мы вернемся обратно на страницу со списками EC2 инстансов и выберем наш private instance.

00:27:14.000 --> 00:27:25.000

Обратите внимание, что у него отсутствует публичный IP адрес, это корректно, так как этот инстанс был создан в private subnet и из интернета к нему подключиться не получится.

00:27:25.000 --> 00:27:42.000

Но у него есть private IP адрес и мы будем его использовать, так как bastion host и этот private instance находятся в одном VPC, таким образом они могут подключаться друг к другу через приватный IP адреса.

00:27:42.000 --> 00:28:04.000

Давайте скопируем это значение и инициируем подключение следующей команды, SSH пробел далее имя пользователя это ec2-user и через символ собачкой мы пишем наш приватный IP адрес инстанса в private subnet.

00:28:04.000 --> 00:28:15.000

Как только мы написали команду, нажимаем на кнопку Enter, далее подтверждаем, что мы действительно хотим подключиться к этому инстансу, вводим команду Yes и нажимаем Enter.

00:28:15.000 --> 00:28:23.000

После чего вы увидите, что появится промпт и IP адрес поменялся, это IP адрес нашего private instance.

00:28:23.000 --> 00:28:33.000

Отлично, я вас поздравляю, мы смогли успешно подключиться к private instance в private subnet.

00:28:33.000 --> 00:28:46.000

Теперь самая последняя проверка, нам необходимо убедиться, что действительно этот инстанс в private subnet может выходить в интернет.

00:28:46.000 --> 00:29:01.000

Для этого можем ввести следующую команду ping и IP адрес со всеми восьмерками это DNS google, либо можем ввести URL адрес любого из знакомых вам веб-сайтов.

00:29:01.000 --> 00:29:19.000

Давайте введем ping и google.com, вы видите, что в первом, что во втором случае, обмениваются успешно пакеты, что говорит о том, что у инстансов в private subnet есть выход в интернет.

00:29:19.000 --> 00:29:30.000

Если у вас действительно также, это говорит о том, что все настройки необходимые до этого момента, мы произвели верно, все корректно настроили.

00:29:30.000 --> 00:29:41.000

Мы с вами переходим к следующему заданию и у нас остался еще один компонент, это Network ACL, еще один компонент VPC, который является дополнительным слоем защиты.

00:29:41.000 --> 00:29:45.000

Давайте посмотрим, как ее создавать и как можно с ней работать.

00:29:45.000 --> 00:30:04.000

Для этого в AWS Management Console перейдем в сервис VPC. Далее здесь воспользуемся навигационным меню слева и выберем опцию Network ACL. Обратите внимание, что в группе security у нас находится два пункта, это Network ACL и Security groups.

00:30:04.000 --> 00:30:29.000

Как только мы перешли к списку Network ACLs, обратите внимание, что у нас есть дефолтовый Network ACL, который привязан к Lab VPC, если вы ее выберете и перейдете в секции Metadata на вкладку Subnet associations, вы увидите два наших subnet, это private и public subnet.

00:30:29.000 --> 00:30:35.000

Они по умолчанию ассоциированы с дефолтовым Network ACL.

00:30:35.000 --> 00:30:41.000

Если мы посмотрим на Inbound rules, мы увидим, что весь трафик разрешен с приоритетом 100.

00:30:41.000 --> 00:30:48.000

Так как это наивысший приоритет в рамках Inbound rules, то весь трафик будет разрешен.

00:30:48.000 --> 00:30:51.000

Также обратите внимание, что это дефолтовый Network ACL.

00:30:51.000 --> 00:30:57.000

А для дефолтового Network ACL весь входящий и исходящий трафик автоматически разрешен.

00:30:57.000 --> 00:31:07.000

Это же мы можем перепроверить во вкладке Outbound rules, вы видите, что с приоритетом 100 весь outbound трафик разрешен.

00:31:07.000 --> 00:31:10.000

Давайте теперь создадим кастомный Network ACL.

00:31:10.000 --> 00:31:15.000

Для этого вверху справа необходимо будет нажать на кнопку Create Network ACL.

00:31:15.000 --> 00:31:24.000

Здесь необходимо будет ввести имя этого ресурса, в нашем случае Lab Network ACL и как VPC необходимо будет выбрать Lab VPC.

00:31:24.000 --> 00:31:29.000

После чего в нижней части страницы мы можем нажать на кнопку Create Network ACL.

00:31:29.000 --> 00:31:45.000

Как только создаться наш кастомный Network ACL, нас направит на список этих ресурсов, и вы увидите в соответствующем столбце, что Lab Network ACL является не дефолтовым Network ACL.

00:31:45.000 --> 00:31:46.000

Отлично.

00:31:46.000 --> 00:31:52.000

Давайте теперь выберем его и в секции Inbound rules посмотрим, какие у нас есть rules.

00:31:52.000 --> 00:32:01.000

Вы увидите, что трафик по умолчанию запрещен, и для inbound rules и для outbound rules.

00:32:01.000 --> 00:32:06.000

Это дефолтовое поведение кастомного Network ACL.

00:32:06.000 --> 00:32:12.000

Давайте теперь добавим некоторые rules для того чтобы разрешить трафик.

00:32:12.000 --> 00:32:17.000

Для этого необходимо будет нажать на кнопку Edit outbound rules.

00:32:17.000 --> 00:32:30.000

Здесь необходимо будет нажать на кнопку Add new rule и приоритетом 100 разрешить весь трафик до CIDR блока публичного subnet.

00:32:30.000 --> 00:32:33.000

И после чего сохраняем изменения.

00:32:33.000 --> 00:32:38.000

Все те же самые изменения необходимо внести для Inbound rules.

00:32:38.000 --> 00:32:52.000

А именно, мы добавляем новый rule с приоритетом 100, разрешающий весь трафик до источника, это CIDR блок публичного subnet и сохраняем эти изменения.

00:32:52.000 --> 00:32:57.000

Далее, напоминаю здесь, что Network ACL является самостоятельным ресурсом.

00:32:57.000 --> 00:33:05.000

Поэтому при создании кастомного Network ACL необходимо явно его привязать к необходимым subnets.

00:33:05.000 --> 00:33:13.000

В нашем случае мы привяжем его к private subnet.

00:33:13.000 --> 00:33:17.000

Как только мы это сделаем, нам необходимо провести эксперимент.

00:33:17.000 --> 00:33:23.000

Чтобы это сделать нам необходимо в public subnet создать тестовый инстанс.

00:33:23.000 --> 00:33:26.000

Поэтому давайте перейдем в сервис EC2.

00:33:26.000 --> 00:33:31.000

Далее, нажмем на кнопку Launch instance и ведем входные данные.

00:33:31.000 --> 00:33:33.000

Как Name укажем Test Instance.

00:33:33.000 --> 00:33:41.000

Далее, как операционную систему выберем Amazon Linux и как AMI выберем Amazon Linux 2 AMI (HVM).

00:33:41.000 --> 00:33:45.000

Как Instance type выберем t2.micro.

00:33:45.000 --> 00:33:49.000

Приватный ключ можем не указывать, так как к нему подключаться не будем.

00:33:49.000 --> 00:33:53.000

В настройках сети необходимо будет внести некоторые изменения.

00:33:53.000 --> 00:33:58.000

В поле VPC необходимо будет выбрать Lab VPC.

00:33:58.000 --> 00:34:04.000

Как Subnet необходимо будет выбрать Public subnet и создать новую Security группу.

00:34:04.000 --> 00:34:12.000

Здесь необходимо будет нажать на кнопку Add security group rule и ввести соответствующие значения.

00:34:12.000 --> 00:34:22.000

А именно мы разрешаем весь трафик ICMP по IPv4 протоколу.

00:34:22.000 --> 00:34:30.000

Далее, как Source type указываем Anywhere, то есть весь интернет и как Description напишем Allow ICMP.

00:34:30.000 --> 00:34:35.000

Это в том числе разрешает трафик с использованием команды ping.

00:34:35.000 --> 00:34:49.000

Как только мы заполним эти все поля, необходимо будет в правой части экрана прокрутить до конца страницы и нажать на кнопку Launch instance.

00:34:49.000 --> 00:35:00.000

Пока наш инстанс создается, нам необходимо будет выбрать Test Instance и скопировать его приватный IP адрес.

00:35:00.000 --> 00:35:14.000

Как только мы скопируем, возвращаемся в SSH сессию, где мы подключены к инстансу в private subnet и запустим команду ping и IP адрес нашего тестового инстанса.

00:35:14.000 --> 00:35:17.000

Этот тестовый инстанс находится в public subnet.

00:35:17.000 --> 00:35:26.000

Вы увидите, что у нас обмениваются пакеты, они успешно уходят и возвращаются, указывается время.

00:35:26.000 --> 00:35:34.000

Давайте оставим обмен пакетами выполняться дальше и вернемся в AWS Management Console.

00:35:34.000 --> 00:35:45.000

Здесь нам необходимо будет выбрать кастомный Network ACL, далее во вкладке Outbound rules нажать на кнопку Edit outbound rules.

00:35:45.000 --> 00:36:15.000

Здесь мы сейчас введем новое правило приоритетом выше, пусть будет приоритет 50, и для Type трафика ICMP по IPv4, для Destination это приватный IP адрес тестового инстанса, CIDR блок размера 32, так как мы указываем конкретный IP адрес и в поле Allow/Deny укажем Deny.

00:36:15.000 --> 00:36:23.000

То есть мы запрещаем ICMP трафик до нашего тестового инстанса и сохраняем эти изменения.

00:36:23.000 --> 00:36:33.000

Как только мы сохраним изменения, обратите внимание, что в сессии PuTTY обмен пакетами прекратился.

00:36:33.000 --> 00:36:45.000

Мы можем выйти из программы и запустить команду ping и приватный IP адрес тестового инстанса еще раз и вы увидите, что пакеты не обмениваются.

00:36:45.000 --> 00:36:57.000

Это говорит о том, что мы воспользовались дополнительным слоем защиты Network ACL и запретили интересующий нас трафик до интересующего CIDR блока.

00:36:57.000 --> 00:37:06.000

В нашем случае это IP адрес тестового инстанса в public subnet.

00:37:06.000 --> 00:37:12.000

Отлично, мы с вами выполнили все задания в рамках лабораторной работы с чем я вас поздравляю.

00:37:12.000 --> 00:37:15.000

Осталось только ответить на некоторые тестовые вопросы.

00:37:15.000 --> 00:37:22.000

Для того чтобы до них добраться, необходимо на странице лабораторной работы нажать на кнопку Details.

00:37:22.000 --> 00:37:27.000

Далее в выпадающем списке нажать на кнопку Show.

00:37:27.000 --> 00:37:36.000

Во всплывающем окне необходимо будет прокрутить в самый низ и нажать на ссылку Access the multiple choice questions.

00:37:36.000 --> 00:37:40.000

Вас направят на отдельную страницу в веб-браузере со списком вопросов.

00:37:40.000 --> 00:37:43.000

Всего дано 6 вопросов.

00:37:43.000 --> 00:37:46.000

Давайте попробуем ответить на каждый из них.

00:37:46.000 --> 00:37:54.000

Самый первый вопрос: для чего используется Internet gateway с public subnet?

00:37:54.000 --> 00:38:07.000

Вы можете остановить это видео и попробовать ответить на него самостоятельно.

00:38:07.000 --> 00:38:20.000

Internet gateway используется для того, чтобы предоставить выход в интернет для ресурсов в public subnet и для тех ресурсов, у которых есть публичный IP адрес.

00:38:20.000 --> 00:38:23.000

Второй вопрос.

00:38:23.000 --> 00:38:31.000

Что дает возможность выходить в интернет и устанавливать обновление ПО ресурсам из private subnet?

00:38:31.000 --> 00:38:39.000

Вы также можете остановить это видео и попробовать ответить на него самостоятельно.

00:38:39.000 --> 00:38:41.000

Правильный ответ - NAT gateway.

00:38:41.000 --> 00:38:43.000

Переходим к следующему вопросу.

00:38:43.000 --> 00:38:47.000

Третий вопрос.

00:38:47.000 --> 00:38:55.000

Можно ли из интернета напрямую обратиться к инстансу в private subnet?

00:38:55.000 --> 00:39:03.000

Вы можете остановить видео и попытаться ответить самостоятельно.

00:39:03.000 --> 00:39:05.000

Правильный ответ - NO.

00:39:05.000 --> 00:39:11.000

То есть к инстансу, находящемуся в private subnet, напрямую из интернета подключиться невозможно.

00:39:11.000 --> 00:39:26.000

Таким образом, в private subnet мы можем хранить чувствительные ресурсы, именно базу данных, возможно инстансы, где отрабатывает логика вашего приложения.

00:39:26.000 --> 00:39:28.000

Четвертый вопрос.

00:39:28.000 --> 00:39:37.000

Почему мы использовали два различных приватных ключа для private instance и bastion host?

00:39:37.000 --> 00:39:44.000

Можете остановить видео и попытаться ответить самостоятельно.

00:39:44.000 --> 00:39:55.000

Правильный ответ является то, что мы создали отдельные ключи, чтобы в случае утечки приватного ключа минимизировать его возможный ущерб.

00:39:55.000 --> 00:40:09.000

То есть в случае если публично станет доступным приватный ключ от приватного инстанса, пользователи все равно добраться до него не смогут, так как не смогут подключиться с этим ключом к bastion host.

00:40:09.000 --> 00:40:34.000

В обратном случае, если же у нас уйдет в публичный доступ приватный ключ bastion host, то будет возможность подключиться только к этому bastion host, а далее к другим инстансам, в том числе к нашему private инстансу подключиться также не получится, так как к нему подключается через другой приватный ключ.

00:40:34.000 --> 00:40:39.000

Следующий вопрос.

00:40:39.000 --> 00:40:49.000

Можно ли запустить команду ping на стороне bastion host в сторону приватного инстанса и сможем ли мы получить ответ?

00:40:49.000 --> 00:40:53.000

Подумайте хорошенько и попытайтесь ответить самостоятельно.

00:40:53.000 --> 00:41:05.000

Помимо того, что вы ответите на вопрос, также подумайте почему тот или иной ответ является верным.

00:41:05.000 --> 00:41:09.000

Окей, правильным ответом является No.

00:41:09.000 --> 00:41:11.000

Отвечаю почему.

00:41:11.000 --> 00:41:25.000

Когда мы настраивали security group для приватного инстанса, мы создавали inbound rule только для SSH трафика.

00:41:25.000 --> 00:41:40.000

Так как ping не относится к SSH трафику, он относится к другому виду трафика, то из bastion host в инстанс из private subnet мы не можем успешно выполнить команду ping.

00:41:40.000 --> 00:41:48.000

Шестой вопрос, также очень хороший вопрос, мне он очень понравился.

00:41:48.000 --> 00:42:04.000

Какие security group rules разрешают приватному EC2 инстансу получить обратный трафик, обратный результат выполнения команды ping до тестового инстанса?

00:42:04.000 --> 00:42:11.000

Я напоминаю, что приватный инстанс находится в private subnet, а тестовый инстанс находится в public subnet.

00:42:11.000 --> 00:42:16.000

Хорошенько подумайте, вы можете остановить видео и попытаться ответить самостоятельно.

00:42:20.000 --> 00:42:33.000

Правильный ответ на шестой вопрос, это outbound rules для private instance и inbound rules для тестового инстанса.

00:42:33.000 --> 00:42:46.000

Почему? Потому что security group - stateful, что говорит о том, что трафик, который вошел, вне зависимости от того, разрешен ли outbound трафик или нет, он вернется обратно.

00:42:46.000 --> 00:43:00.000

И обратный случай, тот трафик, который разрешен как outbound, вне зависимости от того, для конкретной security группы разрешен inbound или нет, он все равно вернется.

00:43:00.000 --> 00:43:16.000

Поэтому, чтобы инициировать команду ping, нам необходимо, чтобы security группа приватного инстанса содержала соответствующие outbound rules, зная, что inbound трафик обязательно вернется.

00:43:16.000 --> 00:43:25.000

А для тестового инстанса эта команда является inbound трафиком, поэтому необходимо, чтобы inbound трафик был разрешен.

00:43:25.000 --> 00:43:36.000

А так как security группа является stateful, то тот трафик, который вошел, он обратно вернется, вне зависимости от того, какие прописаны outbound рулы.

00:43:36.000 --> 00:43:47.000

На этом мы ответили на все вопросы. Теперь давайте проверим результат оценивания выполнения нашей лабораторной работы.

00:43:47.000 --> 00:43:55.000

Для этого на странице лабораторной работы необходимо нажать на кнопку Submit, подтвердить его и подождать некоторое время.

00:43:55.000 --> 00:44:01.000

Так как эта лабораторная работа является достаточно объемной, то проверка займет некоторое время.

00:44:01.000 --> 00:44:15.000

Как только скрипт оценивания отработает, вы увидите свои баллы. В моем случае, я набрал 56 баллов из 56, значит мы с вами все задания выполнили корректно, а также ответили правильно на все вопросы.

00:44:15.000 --> 00:44:29.000

Если же у вас не набран максимальный балл, то вы можете прокрутить чуть ниже, посмотреть, в каком месте у вас не достают баллы, исправить ее и запустить оценивание еще раз.

00:44:29.000 --> 00:44:39.000

Если же вы набрали максимальные баллы, я вас поздравляю. Отлично. Теперь нам необходимо правильно выйти со всех систем.

00:44:39.000 --> 00:44:51.000

В первую очередь это относится к AWS Management Console. Нам необходимо в верхней правой части нажать на имя пользователя и в выпадающем списке нажать на кнопку Sign out.

00:44:51.000 --> 00:44:57.000

Далее, необходимо вернуться на страницу с лабораторной работой и здесь нажать на кнопку End lab.

00:44:57.000 --> 00:45:06.000

Подтвердить, что мы завершаем лабораторную работу и дождаться сообщения, что мы можем закрывать страницу с лабораторной работой.

00:45:06.000 --> 00:45:16.000

На этом мы завершаем сессию разбора лабораторной работы. Это была достаточно объемная лабораторная работа.

00:45:16.000 --> 00:45:26.000

Мы познакомились со всеми компонентами VPC и я очень надеюсь, что вы получили полное представление о VPC и ее компонентах.

00:45:26.000 --> 00:45:32.000

Спасибо за внимание. Увидимся с вами на следующих наших активностях.