WEBVTT

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

Добрый день, уважаемые студенты. Мы с вами начинаем разбор вопроса Practice Test-а.

00:00:07.000 --> 00:00:11.000

Согласно вопросу у нас three-tier web application, то есть три уровня.

00:00:11.000 --> 00:00:15.000

В первом уровне front-end у нас статические веб-страницы.

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

Во втором уровне, уровне логики приложения работают на одной EC2 инстанс машине.

00:00:22.000 --> 00:00:26.000

И здесь выполняются достаточно долгие задания.

00:00:26.000 --> 00:00:33.000

На последнем уровне, уровне данных у нас работает MySQL база данных.

00:00:33.000 --> 00:00:58.000

Вам как Solutions Architect пришло задание, чтобы все приложение сделать highly available, то есть высоко доступное, за счет decoupling ее частей, то есть сделать их менее связанными между собой.

00:00:58.000 --> 00:01:00.000

Что для этого можно сделать?

00:01:00.000 --> 00:01:05.000

Нам необходимо выбрать один правильный вариант ответа из предложенных опций.

00:01:05.000 --> 00:01:12.000

На этом слайде вы видите варианты ответа, так же видите, что второй вариант ответа является верным.

00:01:12.000 --> 00:01:17.000

Тем не менее, давайте разберем каждый вариант.

00:01:17.000 --> 00:01:28.000

Если посмотреть на структуру ответов, они все состоят из трех частей, и каждая часть относится к каждому из трех уровней нашего веб-приложения.

00:01:28.000 --> 00:01:33.000

Первое предложение говорит про front-end уровень.

00:01:33.000 --> 00:01:50.000

Мы видим, что для первого варианта предлагается все наши веб-страницы, ассеты, то есть картинки, видео, перенести на уровень back-end, положить возле приложения внутри EC2 инстанса и увеличить размер инстанса.

00:01:50.000 --> 00:02:03.000

Это противоречить тому, что требовалось в вопросе, нам нужно было компоненты разделить и сделать слабосвязанными, а в этом случае мы наоборот их объединяем, поэтому этот вариант можем сразу исключать.

00:02:03.000 --> 00:02:10.000

Если посмотреть на второй вариант, предлагается все статические ассеты и веб-страницы перенести на S3.

00:02:10.000 --> 00:02:17.000

Да, у нас на стороне S3 есть возможность хостить статические веб-сайты, можем ее включить и использовать.

00:02:17.000 --> 00:02:22.000

Это решение намного дешевле, нежели хостить веб-страницы на EC2.

00:02:22.000 --> 00:02:26.000

Поэтому этот вариант пока оставляем, смотрим третий.

00:02:26.000 --> 00:02:33.000

На третьем варианте предлагается все статические ассеты, но не веб-страницы перенести на S3.

00:02:33.000 --> 00:02:44.000

Возможно, это какая-то ошибка, если это не ошибка, а намеренно, то этот вариант скорее всего не верный, так как S3 также может хостить веб-страницы, не только ассеты.

00:02:44.000 --> 00:02:51.000

Если мы посмотрим на четвертый вариант, здесь предлагается все статические ассеты и веб-страницы перенести на CloudFront.

00:02:54.000 --> 00:03:11.000

Этот вариант логически не совсем верен, так как CloudFront сам по себе является Content Delivery Network. Идея в том, что это сеть из серверов с кэшем, и для того, чтобы в кэш положить некоторые данные, нам нужен источник.

00:03:11.000 --> 00:03:19.000

Источником обычно используется Amazon S3, они очень хорошо работают в связке, то есть S3 плюс CloudFront.

00:03:19.000 --> 00:03:23.000

Поэтому четвертый вариант логически не совсем верен, мы его исключаем.

00:03:23.000 --> 00:03:26.000

У нас состоится второй и третий варианты ответов.

00:03:26.000 --> 00:03:28.000

Давайте посмотрим на вторую часть.

00:03:28.000 --> 00:03:38.000

Во второй части для второго варианта предлагается использовать Elastic Container Service и настроить там Auto Scaling.

00:03:38.000 --> 00:03:45.000

Ничего критичного и противоречивого мы не видим, мы можем наше приложение перенести на контейнеры.

00:03:45.000 --> 00:03:50.000

Давайте посмотрим, что у нас есть на третьем варианте во втором предложении.

00:03:50.000 --> 00:03:55.000

Здесь предлагается настроить concurrency limit и наше приложение перенести на Lambda.

00:03:55.000 --> 00:03:59.000

Таким образом перейти на serverless архитектуру.

00:03:59.000 --> 00:04:08.000

Этот вариант не совсем подходит, потому что Lambda максимально может работать 15 минут, то есть 900 секунд.

00:04:08.000 --> 00:04:15.000

В вопросе же сказано, что у нас отрабатывают long-running processes, то есть скорее всего это больше 15 минут.

00:04:15.000 --> 00:04:19.000

И здесь была как раз таки отсылка, чтобы не использовать Lambda.

00:04:19.000 --> 00:04:24.000

Поэтому мы исключаем третий вариант, остается второй вариант.

00:04:24.000 --> 00:04:31.000

Тем не менее, давайте посмотрим на третье предложение в других вариантах.

00:04:31.000 --> 00:04:40.000

Во втором варианте предлагается мигрировать нашу базу данных на Amazon RDS и включить Multi-AZ deployment.

00:04:40.000 --> 00:04:46.000

Да, это хорошее решение, использование managed service помогает разгрузиться от административных задач.

00:04:46.000 --> 00:04:56.000

Более того есть готовые решения Multi-AZ, которая помогает вам сделать вашу базу данных highly available, то есть высоко доступной.

00:04:56.000 --> 00:05:05.000

Если что-то происходит на стороне одной availability зоны, происходит автоматический failover, и secondary database становится основным.

00:05:05.000 --> 00:05:08.000

Таким образом база продолжает работать.

00:05:08.000 --> 00:05:21.000

Если же мы посмотрим на третье предложение в третьем варианте, здесь предлагается мигрировать нашу базу данных в DynamoDB.

00:05:21.000 --> 00:05:39.000

DynamoDB - это NoSQL база данных, поэтому ту базу данных, которая уже хостится на MySQL, это реляционные базы данных, проще перенести в сервис RDS.

00:05:39.000 --> 00:05:42.000

DynamoDB это NoSQL, поэтому необходимо провести работы по моделированию данных и поменяется вся структура.

00:05:42.000 --> 00:05:48.000

Поэтому это сложно с точки зрения усилий.

00:05:48.000 --> 00:05:53.000

Исходя из этого мы однозначно можем утверждать, что второй вариант является верным.

00:05:53.000 --> 00:06:08.000

На следующих страницах будет также текстовый разбор этого вопроса, можете не спеша остановить видео и ознакомиться с ним.

00:06:08.000 --> 00:06:13.000

На этом мы завершаем разбор вопроса Practice Test-а. Спасибо за внимание.