WEBVTT

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

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

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

Согласно вопросу, компания хостит свое приложение на EC2 инстансах.

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

Здесь используется Auto Scaling group для автомасштабирования.

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

Также используется Application Load Balancer для распределения нагрузки и это все работает на нескольких availability зонах.

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

Во время работы приложения получилось так, что Application Load Balancer автоматически терминейтнул несколько инстансов, так как они ответили Не успешно для health check.

00:00:38.000 --> 00:00:44.000

Поэтому эти инстансы незамедлительно были удалены.

00:00:44.000 --> 00:00:53.000

Также остаток логов, который не успел загрузиться в CloudWatch Logs, они также были удалены вместе с инстансом.

00:00:53.000 --> 00:01:07.000

Необходимо из предложенных вариантов выбрать самый легкий путь, который позволит достоверно получить все логи из EC2 инстанса перед его удалением.

00:01:11.000 --> 00:01:16.000

На этом слайде вы видите варианты ответов, они достаточно объемные.

00:01:16.000 --> 00:01:21.000

Вы также видите, что правильным ответом на этот вопрос является четвертый вариант.

00:01:21.000 --> 00:01:26.000

Тем не менее, давайте попробуем разобрать этот вопрос.

00:01:26.000 --> 00:01:38.000

Когда приходят вопросы с большими вариантами ответов, то я вам крайне рекомендую не читать отдельно каждый вариант, а начать уже сразу сравнивать варианты ответов.

00:01:38.000 --> 00:01:48.000

Скорее всего, по мере сравнивания у вас постепенно будут отпадать некоторые варианты, которые логически не подходят, либо не являются достоверными.

00:01:48.000 --> 00:01:55.000

Давайте попробуем ответить на этот вопрос, так как если бы мы были на реальном экзамене AWS.

00:01:55.000 --> 00:02:18.000

Прочитав первое предложение всех вариантов, мы видим, что они все похожи, кроме второго варианта, который предлагает создать lifecycle hook, который для Auto Scaling group будет двигать инстансы из состояния Terminating в состояние Pending:Wait.

00:02:18.000 --> 00:02:26.000

Во всех других вариантах оно будет менять состояние из Terminating в Terminating:Wait.

00:02:26.000 --> 00:02:41.000

Здесь логически можно предположить, что Pending:Wait скорее всего является неверным вариантом ответа, так как задача авторов вопроса спрятать правильный ответ максимально глубже.

00:02:41.000 --> 00:02:45.000

Поэтому мы можем исключить второй вариант ответа.

00:02:45.000 --> 00:03:01.000

Более того, если мыслить логически, если приложение перестало работать и мы переводим ее в состояние Terminating, то переводя ее в состояние Pending:Wait, следующим статусом будет Running.

00:03:01.000 --> 00:03:08.000

И нет никакого смысла возвращать в работу приложение, если оно не работает.

00:03:08.000 --> 00:03:21.000

Поэтому правильнее все-таки из Terminating в состояние перемещать в Terminating:Wait, чтобы дать какое-то время перед ее окончательным отключением.

00:03:21.000 --> 00:03:25.000

У нас остались первый, третий и четвертый варианты ответов.

00:03:25.000 --> 00:03:34.000

Второе предложение говорит нам о том, что мы должны настроить CloudWatch Events Rule.

00:03:34.000 --> 00:03:46.000

Оно должно привязаться к событию EC2 Instance-terminate Lifecycle Action, либо EC2 Instance Terminate Successful.

00:03:46.000 --> 00:03:58.000

А в третьем варианте ответа предлагается сразу настроить Step Functions, не настраивая CloudWatch Events Rule.

00:03:58.000 --> 00:04:11.000

Третий вариант мы можем отсечь, потому что Step Functions сам по себе не поймет, что EC2 инстанс переходит в состояние Terminating:Wait и надо что-то с ним делать.

00:04:11.000 --> 00:04:12.000

Поэтому этот вариант мы убираем.

00:04:12.000 --> 00:04:23.000

Более того, Step Functions сам по себе не проделывает никаких действий, чтобы были какие-то действия, вы в рамках Step Functions будете создавать Lambda.

00:04:23.000 --> 00:04:34.000

Если вы обратите внимание, что Lambda упоминается и в первом и в четвертом варианте, но в этом случае она используется без Step Functions, то есть это намного проще.

00:04:34.000 --> 00:04:38.000

Step Functions лишь добавляет сложность к имплементации.

00:04:38.000 --> 00:04:44.000

Нам в рамках вопроса нужно было выбрать самое простое решение, поэтому третий вариант ответа мы также исключаем.

00:04:44.000 --> 00:04:47.000

Остается первый и четвертый варианты ответа.

00:04:47.000 --> 00:04:56.000

Теперь если обратить внимание на сам event, четвертый привязывается к событию EC2 Instance-terminate Lifecycle Action.

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

Событие говорит о том, что инстанс переходит в состояние Terminate и нужно в этот момент что-то с ним делать.

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

А в первом варианте ответа event уже другой.

00:05:08.000 --> 00:05:14.000

Он говорит о том, что EC2 инстанс успешно был отключен и после этого приходит event.

00:05:14.000 --> 00:05:21.000

Чисто логически первый вариант можно отсечь, так как после того как инстанс отключился мы уже с ним ничего сделать не сможем.

00:05:21.000 --> 00:05:24.000

Нам нужно, чтобы он оставался включенным.

00:05:24.000 --> 00:05:34.000

Поэтому не до конца прочитав варианты ответа, мы уже попытались выявить, что четвертый вариант ответа является верным.

00:05:34.000 --> 00:05:39.000

Если прочитать ее до конца, нам предлагается настроить Lambda.

00:05:39.000 --> 00:05:50.000

Она привязывается к событию EC2 Instance-terminate Lifecycle Action и target для этого rule является Lambda функция.

00:05:50.000 --> 00:05:56.000

Lambda функция в свою очередь подключается к CloudWatch Agent в EC2 инстансе.

00:05:56.000 --> 00:06:12.000

И не дождавшись очередной отправки группы логов, запускает команду для принудительной отправки тех логов, которые остались, еще не успели отправиться.

00:06:12.000 --> 00:06:24.000

После того как успешно все логи отправляются в CloudWatch Logs, можно завершать работу Lambda и можно терминейтить EC2 инстанс.

00:06:24.000 --> 00:06:34.000

Так как все логи уже собраны в CloudWatch Logs и этого достаточно, чтобы по логам определить в чем была проблема и попытаться ее решить.

00:06:34.000 --> 00:06:58.000

На следующих слайдах вы увидите текстовый разбор данного вопроса, с которым вы также можете ознакомиться.

00:06:58.000 --> 00:07:04.000

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