МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»**

**Московский техникум космического приборостроения**

### СПЕЦИАЛЬНОСТЬ: 09.02.07 Информационные системы и программирование (квалификация «Администратор баз данных»)

**П О Я С Н И Т Е Л Ь Н А Я З А П И С К А**

**к курсовому проекту по теме:**

## Использование фреймворков для безопасного программирования биометрической системы идентификации личности

|  |  |  |
| --- | --- | --- |
| Руководитель разработки от техникума | (подпись, дата) | Гончаренко С.В. |
| Разработчик | (подпись, дата) | Симонян П. Р. |

Москва 2021

**«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»**

**Московский техникум космического приборостроения**

УТВЕРЖДАЮ

Председатель ПЦК спец. 09.02.07

Н.А. Жилкина

« » 2021г.

**З А Д А Н И Е**

### на выполнение курсового проекта

по профессиональному модулю «ПМ.04 Сопровождение и обслуживание программного обеспечения компьютерных систем»

Студент Симонян П.Р. ТИП-51

(фамилия, инициалы, индекс группы)

Руководитель Гончаренко С.В.

(фамилия, инициалы)

График выполнения работы: 25% к 4 нед., 50% к 8 нед., 75% к 12 нед., 100% к 15 нед.

1. ***Тема курсового проекта***

Использование фреймворков для безопасного программирования биометрической идентификации личности

1. ***Техническое задание***

Сравнить возможности фреймворков для безопасного программирования биометрической идентификации личности

1. ***Оформление курсового проекта***
   1. Расчетно-пояснительная записка на листах формата А4.
   2. Перечень графического материала КП (плакаты, схемы, чертежи и т.п.) – схемы алгоритма

Дата выдачи задания «\_ » 2021г.

Руководитель курсового проекта Гончаренко С.В.

# СОДЕРЖАНИЕ

[Введение 4](#_TOC_250005)

1. [Постановка задачи 5](#_TOC_250004)
2. [Теоретические основы 5](#_TOC_250003)
   1. Облачные вычисления 5
   2. Интеграция приложений в Amazon Web Services 5
      1. Сервисы интеграции приложений Amazon Web Services 5
      2. Примеры использования интеграции приложений 5
      3. Перемещение эксабайтов данных с AWS Snowmobile 5
   3. Миграция данных с помощью Amazon Web Services 5
      1. Решения Amazon Web Services для миграции 5
   4. Передача данных в Amazon Web Services 5
      1. Перемещение эксабайтов данных с AWS Snowmobile 5
      2. Работа с потоковыми данными в Amazon Kinesis Streams 5
3. [Практическая реализация 5](#_TOC_250002)
   1. Создание таблицы NoSQL и выполнение запросов к ней 5
   2. Настройка документной базы данных 5
   3. Хранение и извлечение файла с помощью Amazon S3 5
4. Тестирование 5
   1. План тестирования 5
   2. Тест-кейсы 5
   3. Баг-репорты 5
   4. Отчёт о тестировании 5
5. [Результаты 5](#_TOC_250001)
   1. Создание бесплатного аккаунта Amazon Web Services 5
   2. Результаты создания таблицы NoSQL и выполнения запросов к ней 5
   3. Результаты настройки документной базы данных 5
   4. Результаты хранения и извлечения файла с помощью Amazon S3 5
   5. Руководство пользователя 5

[Заключение 5](#_TOC_250000)

Приложение А. Руководство пользователя по созданию таблицы NoSQL и выполнения запросов к ней 5

Приложение Б. Руководство пользователя по настройке документной базы данных 5

Приложение В. Руководство пользователя по хранению. и извлечению файла с помощью Amazon S3 5

Приложение Г. Отчёт о тестировании 5

# ВВЕДЕНИЕ

Благодаря развитию аппаратного обеспечения, в XXI веке большую популярность обрели биометрические датчики, которые позволяют сделать взаимодействие с различными устройствами проще и безопаснее. В отличие от классических данных, используемых при аутентификации, таких как пароль или электронные ключи (токенов), биометрические данные не могут быть переданы, так как индивидуальны для каждого человека. Их подделка и подбор практически невозможны, поэтому биометрия считается одним из самых безопасных способов валидации личности.

Существует множество фреймворков для реализации функций биометрических сканеров и анализа собранной информации. В данной работе будут рассмотрены самые популярные из них, а также будет проведено их сравнение.

# ПОСТАНОВКА ЗАДАЧИ

Целью курсового проекта является сравнение возможностей фреймворков для безопасного программирования биометрической идентификации личности.

Основания для проведения работы: учебный план специальности, рабочая программа дисциплины, распоряжение по учебному заведению.

Требования к курсовому проекту: требования к структуре документов определены соответствующими стандартами ЕСКД и КСПД; требования к оформлению определены соответствующими методическими указаниями.

Порядок контроля и приёмки курсового проекта: контроль выполнения курсового проекта проводится руководителем поэтапно в соответствии с утверждённым графиком выполнения курсового проекта; на завершающем этапе руководитель осуществляет нормоконтроль представленной исполнителем документации и принимает решение о допуске (недопуске) курсового проекта к защите.

При работе над курсовым проектом учитывать: полноту раскрытия рассматриваемой темы, актуальность выбираемых фреймворков, корректность их сравнения.

Плановые сроки выполнения: 1 семестр 2021 учебного года.

Исходные данные: в курсовом проекте необходимо провести сравнение на основе официальной документации фреймворков и подтверждённой информации, находящейся в открытом доступе.

Состав курсового проекта:

* Текстовые и графические документы;
* Программная и технологическая документация (при необходимости);
* Пояснительная (расчётно-пояснительная) записка;
* Обязательные структурные элементы: титульный лист (одна страница), содержание (от одной страницы), ведение (от одной страницы), список литературы (от одной страницы), заключение (от одной страницы),
* Объём основной части работы: минимум два раздела с минимум двумя параграфами (от пяти страниц) в каждом разделе.

Содержание пояснительной записки:

* Титульный лист;
* Лист задания;
* Содержание;
* Введение;
* Исходные данные;
* Основной раздел (теоретическая, аналитическая и проектно-аналитическая часть);
* Заключение;
* Список использованной литературы;
* Приложения.

Детализированное техническое задание на курсовой проект:

* во введении обосновать актуальность исследуемой темы, степень её изученности, дать критический обзор работ, характеризующих сущность рассматриваемого вопроса, определить цель и задачи, описать применяемые методы исследования;
* в теоретической части изложить теоретические концептуальные основы исследуемой проблемы и собственное понимание существа вопросов исследуемой темы, сформировать своё аргументированное мнение по неоднозначным вопросам;
* в аналитической части изложить особенности функционирования объекта исследования, проанализировать и раскрыть особенности сбора, обработки и подготовки информации о процессах, целесообразно рассмотреть порядок формирования и представления информации об отдельных объектах исследования;
* в проектно-аналитической части разработать и внести предложения по рациональной методологии исследования объекта с использованием современных информационных технологий, доказать целесообразность и необходимость предлагаемых решений, спрогнозировать результаты от внедрения предложений;
* в заключении изложить основные результаты исследования, сформулировать выводы, предложения и перспективы дальнейшей разработки темы.

# ТЕОРЕТИЧЕСКИЕ ОСНОВЫ

**Биометрия** - это идентификация человека по уникальным биологическим признакам.

В современные устройства и специализированные средства защиты биометрия встраивается все чаще по следующим причинам:

1. Оказывается минимальная когнитивная нагрузка. Для проведения идентификации требуется совершить интуитивные, не требующие умственных усилий или запоминания, действия: коснуться сканера, посмотреть в камеру, произнести фразу.
2. С приходом в массово - потребительские устройства биометрия стала гораздо доступнее и востребованнее.
3. Биометрические сканеры не всегда работают безотказно, но все же остаются достаточно безопасным методом идентификации личности.
4. Тенденция усиления защиты приватности удачно сочетается с тем фактом, что биометрические данные невозможно передать и крайне тяжело подделать.

Остановимся подробнее на последнем пункте.

**Биометрические данные** - уникальные биологические признаки людей. Самыми востребованными являются:

1. Отпечатки пальцев
2. Сетчатка глаза
3. Голос

В отличие от паролей, которые сейчас используются повсеместно, биометрические данные обладают одинаковой сложностью. Это вносит дополнительную защиту, так как пользователь не может сделать этот способ идентификации личности менее надёжным, как если бы он придумал недостаточно сложный и легко угадываемый пароль.

Помимо этого, биометрические данные гораздо тяжелее украсть (или, в случае биометрии, подделать). Скопировать их представляется возможным только для специально обученных людей.

Однако возникают сложности, которые все же не дают гарантию безопасности данного метода идентификации:

1. Все же существуют люди с очень похожими биометрическими признаками. Например, близнецы или люди с похожим тембром голоса.
2. Ключевую роль в обеспечении безопасности и точности биометрии берет на себя разработчик биометрических датчиков и программного обеспечения для них.

Следует учитывать, что надёжность биометрии, в отличие от обычных паролей, сильнее зависит от качества аппаратного и программного продукта, которые обеспечивают ее.

**Фреймворк** - это программный продукт, упрощающий создание и поддержку технически сложных и нагруженных проектов. Обычно задачу программирования биометрических датчиков, методов сбора и обработки информации, полученных с их помощью, берут на себя крупные и проверенные компании или разработчики данных сканеров. Примером вторых могут послужить решения компании BioLink, которая сама выпускает биометрические датчики и комплекты для разработки программного обеспечения (SDK) под них. Но чаще всего встречаются SDK компаний, агрегирующих аппаратное обеспечение разных производителей, чтобы предоставить разработчикам общий интерфейс для их использования. Такими являются Google и Microsoft.

Таким образом, фреймворки способствуют:

А) Усиленной безопасности. Перед тем, как перейти в использование конечным пользователям (в данном случае, это пользователи фреймворков - разработчики), решения для программирования биометрии тщательно тестируются организациями, которые специализируются на этом. В других случаях, такие компании, как, например, Microsoft, имеют все средства для того, чтобы провести качественное тестирование собственных программных решений и предоставить разработчикам безопасный программный продукт.

Б) Облегчённой интеграции. Решения для программирования биометрии, как правило, хорошо задокументированы и не требуют углублённых знаний программиста о работе биометрических сканеров. Ещё проще дела обстоят у разработчиков для массовых платформ: Apple, Google, Microsoft и некоторые другие компании предлагают единые решения для разработчиков под их экосистемы. Пользователю их фреймворков гораздо меньше приходится принимать во внимание разнообразие биометрических датчиков, используемых в конкретных устройствах разных производителей или моделей.

В) Популяризации усиленной безопасности с помощью биометрии. Использование особенностей современных биометрических датчиков может позволить себе любой разработчик, который должен лишь использовать готовое решение в виде фреймворка и изучить документацию к нему. Благодаря этому биометрия пришла в банковскую сферу, сферу онлайн - платежей, а также в другие массовые продукты, предъявляющие специальные требования к безопасности.

Теперь, разобравшись в преимуществах и важности биометрии, стоит рассмотреть некоторые самые популярные решения, которые на данный момент доступны разработчикам.

OpenBR - Open Source фреймворк, позволяющий распознавать лица, возраст и пол по фотографии.

NEUROtechnology Megamatcher SDK - Дорогой профессиональный продукт, ориентированный на крупные компании, которые предъявляют к безопасности особые требования, а также, например, гос. структуры. Введён в список сравнения для того, чтобы дать представление о том, чем профессиональные решения отличаются от массово - потребительских.

Huawei FIDO (Fast Identity Online) BioAuthn - Фреймворк для использования биометрии на смартфонах и планшетах компании Huawei.

Samsung Knox SDK - Масштабный фреймворк компании Samsung, включающий в себя возможность идентификации личности по отпечатку пальца или радужной оболочки глаза.

Apple Local Authentication Framework - Фреймворк позволяющий использовать биометрию на всех устройствах компании Apple, аппаратно поддерживающих её: от iPhone до iMac с использованием соответствующей клавиатуры.

Google’s Android Biometric Library - Фреймворк для использования биометрии на устройствах под операционной системой Android.

Microsoft Windows Biometric Framework - Фреймворк для использования биометрии в операционной системе Windows.

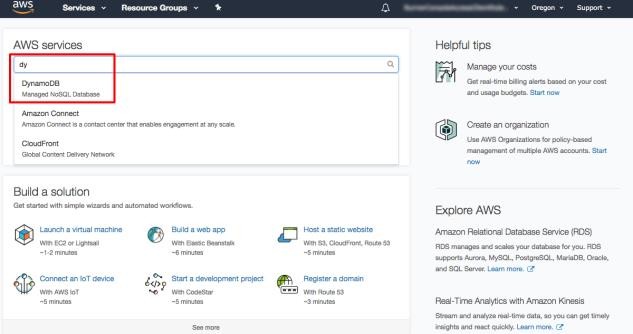
# ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ

3.2 Создание таблицы NoSQL и выполнение запросов к ней с помощью Amazon DynamoDB

В результате можно создать простую таблицу, добавлять данные, сканировать и запрашивать данные, удалять их, а также удалять всю таблицу с помощью консоли DynamoDB. DynamoDB – это полностью управляемая база данных NoSQL, которая поддерживает как документно-ориентированную модель, так и хранилища типа «ключ- значение». Благодаря гибкой модели данных, стабильной производительности и автоматическому масштабированию пропускной способности этот сервис является отличной платформой для мобильных и интернет-приложений, игр, рекламы, IoT и многих других приложений. Все действия доступны на уровне бесплатного пользования.

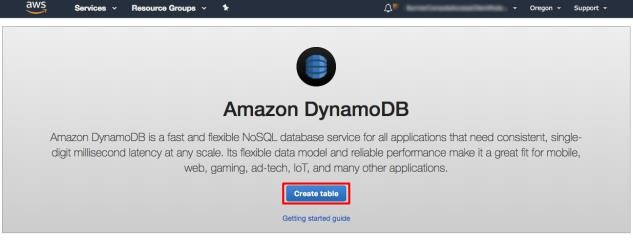
Для создания таблицы NoSQL и выполнения запросов к ней требуется аккаунт. Уровень бесплатного пользования AWS – это 25 ГБ хранилища и до 200 миллионов запросов в месяц для Amazon DynamoDB.

Откройте Консоль управления AWS, оставив открытым данное пошаговое руководство. Когда экран загрузится, начните вводить *DynamoDB* в строке поиска и откройте консоль DynamoDB.

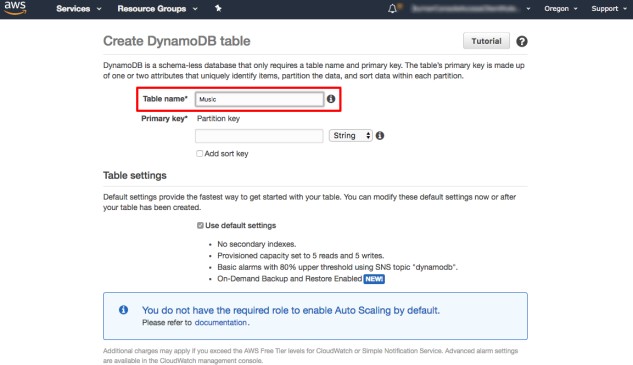


Шаг 1. Создание таблицы NoSQL

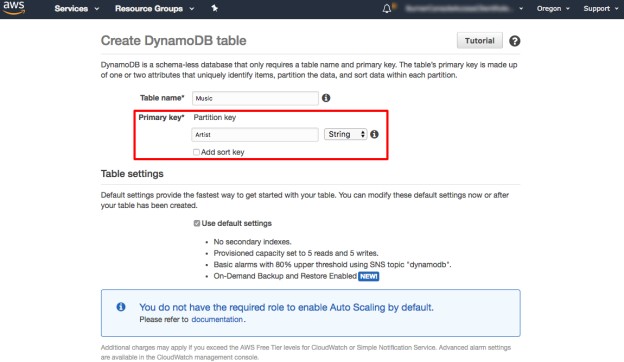
На данном шаге вы создадите таблицу с помощью консоли DynamoDB. а) На консоли DynamoDB выберите Create table.



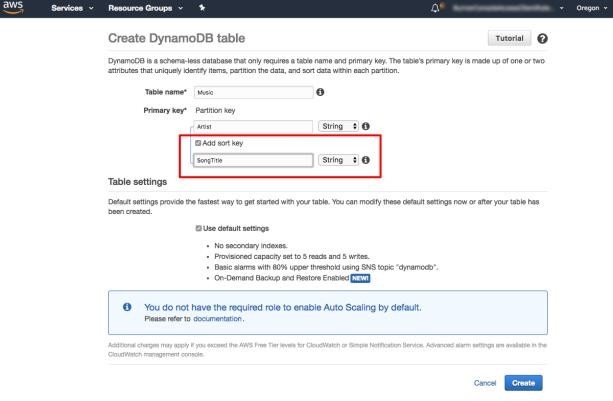
б) В качестве примера использования для этого руководства мы используем музыкальную библиотеку. В поле Table name введите *Music.*



в) Ключ секции служит для распределения данных между разделами с целью обеспечения масштабируемости. Важно выбрать для ключа атрибут с большим диапазоном значений, который с большей вероятностью обеспечит равномерно распределенную схему доступа. Введите *Artist* в поле Partition key.



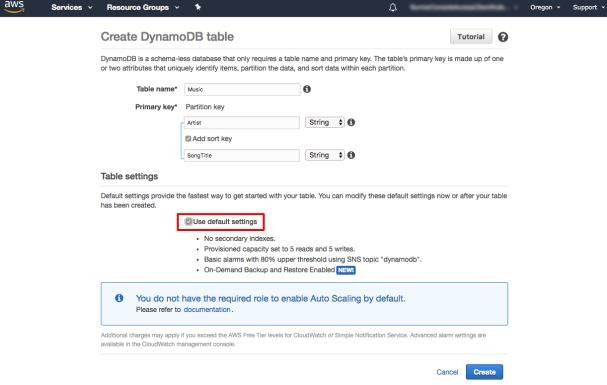
г) Поскольку у каждого исполнителя может быть много композиций, с помощью ключа сортировки можно легко сортировать данные. Установите флажок Add sort key. Введите *songTitle* в поле Add sort key.



д) Затем для таблицы потребуется включить функцию Auto Scaling в DynamoDB.

Auto Scaling в DynamoDB меняет количество выделенных для таблицы ресурсов чтения и записи в зависимости от количества запросов. С помощью роли AWS Identity and Access Management (AWS IAM) под названием *DynamoDBAutoscaleRole* сервис DynamoDB будет управлять автоматическим масштабированием от вашего имени. DynamoDB создает для вас эту роль при первом включении Auto Scaling в аккаунте.

Укажите DynamoDB на необходимость создания этой роли, сняв флажок Use default settings.



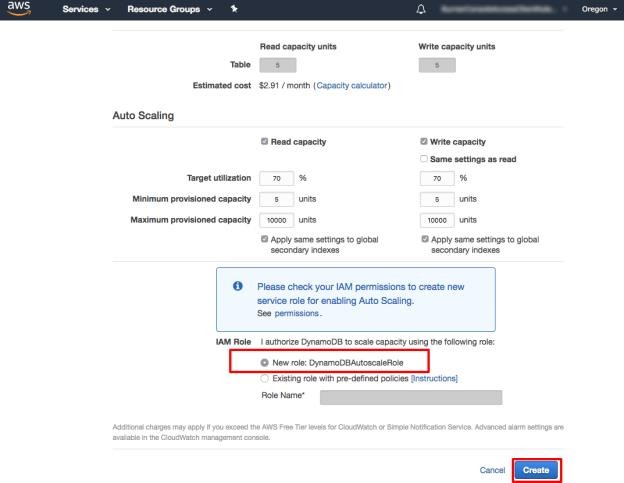
е) Прокрутите экран вниз через пункты Secondary indexes, Provisioned capacity и Auto Scaling до кнопки Create. В рамках данного руководства эти настройки изменяться не будут.

Обратите внимание, что в разделе Auto Scaling DynamoDB создаст для вас роль

*DynamoDBAutoscaleRole*. Теперь выберите Create.

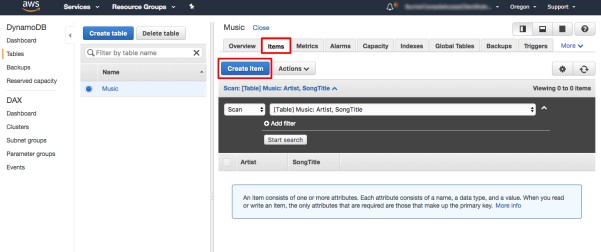
Когда таблица Music будет готова к использованию, она отобразится в списке таблиц с флажком .

Итак, создали таблицу NoSQL с помощью консоли DynamoDB.



Шаг 2. Добавление данных в таблицу NoSQL

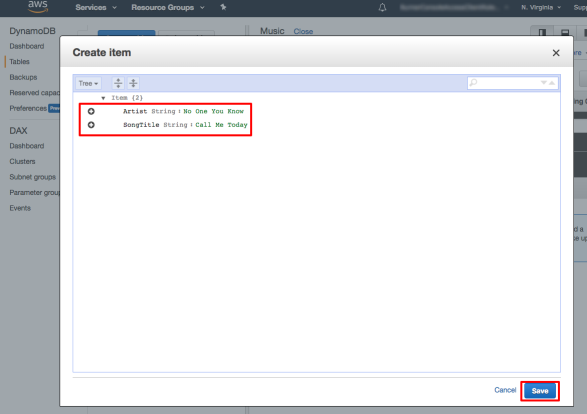
На этом шаге мы добавим данные в новую таблицу DynamoDB. а) Перейдите на вкладку Items. На вкладке Items выберите Create item .



б) В окне ввода данных введите следующее:

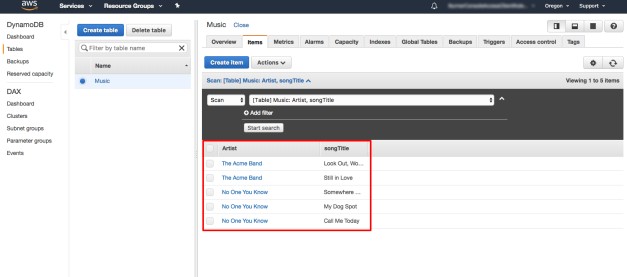
* + Для атрибута Artist введите значение *No One You Know*.
  + Для атрибута songTitle введите *Call Me Today.*

Чтобы сохранить элемент, нажмите кнопку Save.



в) Повторите процедуру, чтобы добавить еще несколько элементов в таблицу *Music*:

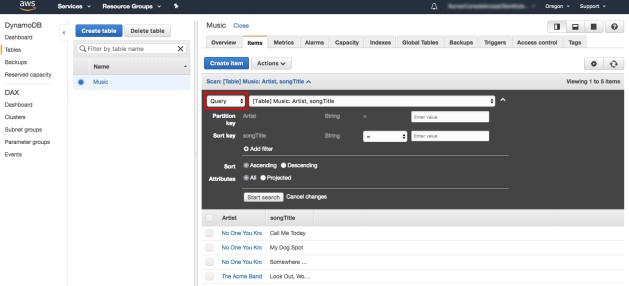
* + Artist: *No One You Know*; songTitle: *My Dog Spot*
  + Artist: *No One You Know*; songTitle: *Somewhere Down The Road*
  + Artist: *The Acme Band*; songTitle: *Still in Love*
  + Artist: *The Acme Band*; songTitle: *Look Out, World*



Шаг 3. Запрос к таблице NoSQL

На этом шаге выполняется поиск данных в таблице с использованием операций запросов. В DynamoDB операции запросов являются эффективным инструментом и используют ключи для поиска данных. Операции сканирования охватывают всю таблицу.

а) В раскрывающемся списке в темно-сером баннере над элементами измените значение Scan на Query.

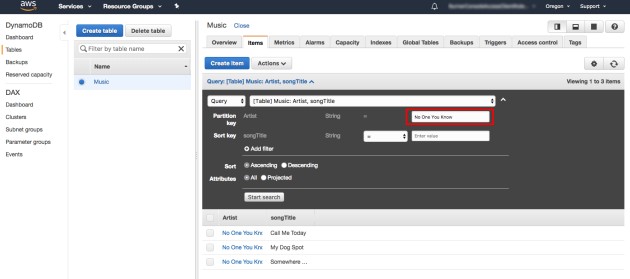


б) Можно по-разному использовать консоль для отправки запросов в таблицу *Music*. Для первого запроса выполните следующее действие:

* + В поле Artist введите *No One You Know* и нажмите кнопку Start search. На экране появятся все песни исполнителя *No One You Know*.

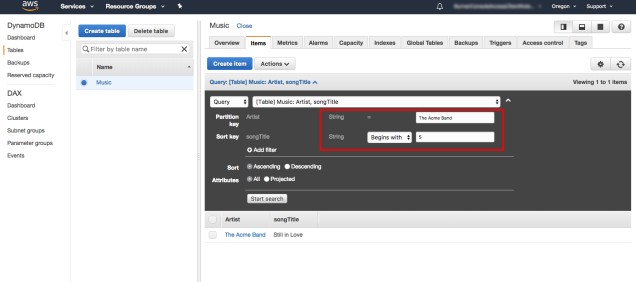
Попробуйте другой запрос:

* + В поле Artist введите *The Acme Band* и нажмите кнопку Start search. На экране появятся все песни исполнителя *The Acme Band*.



в) Попробуйте другой запрос, но на этот раз сузьте результаты поиска:

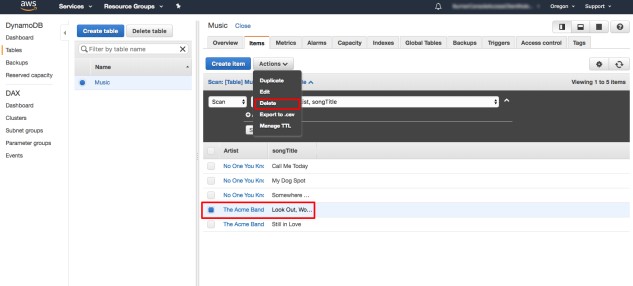
* + В поле Artist введите *The Acme Band*.
  + В поле songTitle выберите Begins with из раскрывающегося списка и введите *S*.
  + Нажмите Start search. Отобразится только композиция Still in Love в исполнении *The Acme Band*.



Шаг 4. Удаление существующего элемента

На этом шаге мы удалим элемент из таблицы DynamoDB. а) Вновь измените раскрывающийся список Query на Scan.

Установите флажок рядом с *The Acme Band*. В раскрывающемся списке Actions выберите Delete. Консоль предложит подтвердить операцию удаления. Выберите Delete, чтобы соответствующий элемент был удален.

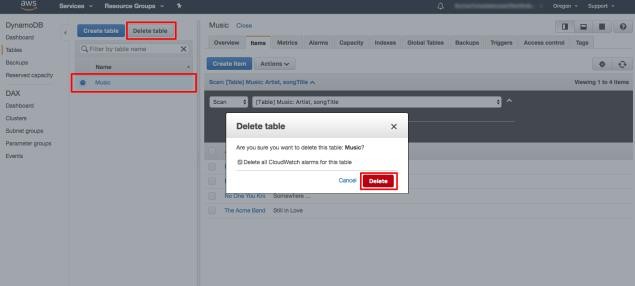


Шаг 5. Удаление таблицы NoSQL

На этом шаге мы удалим таблицу DynamoDB.

а) Можно легко удалить таблицу из консоли DynamoDB. Рекомендуется удалять таблицы, которыми вы больше не пользуетесь, чтобы не платить за соответствующие ресурсы.

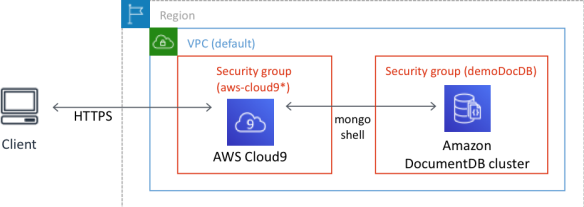
* + На консоли DynamoDB выберите параметр рядом с таблицей Music и выберите Delete table.
  + В диалоговом окне подтверждения выберите Delete.



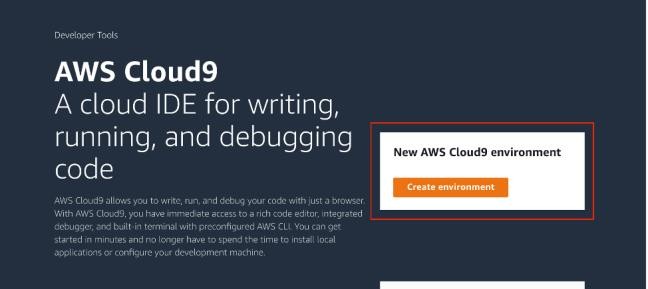
3.2 Настройка документной базы данных с Amazon DocumentDB (с поддержкой MongoDB) и AWS Cloud9

Сервис Amazon DocumentDB (с поддержкой совместимости с MongoDB) – это быстрая, масштабируемая, высокодоступная и полностью управляемая документная база данных, которая поддерживает рабочие нагрузки MongoDB, позволяет хранить и индексировать данные JSON, а также запрашивать их. Описано подключение к кластеру Amazon DocumentDB из среды AWS Cloud9 через оболочку Mongo, а также создание несколько запросов.

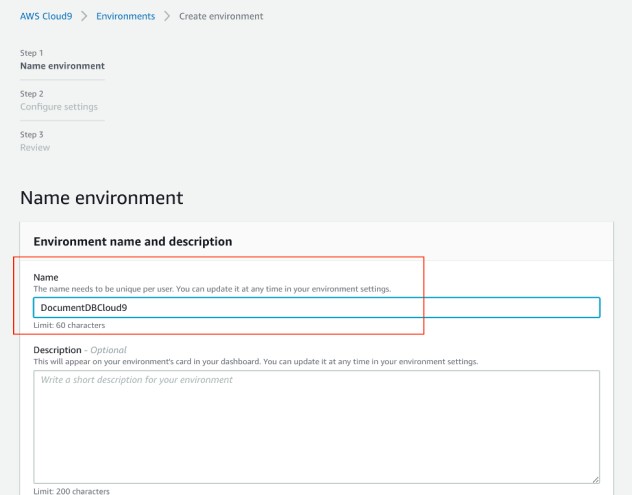
На схеме ниже показана финальная архитектура:



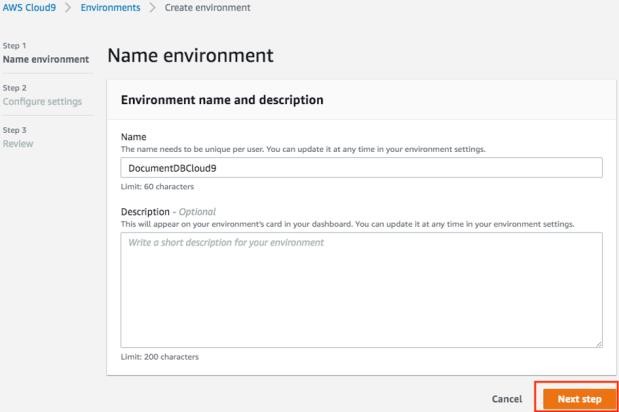
1. Создание среды AWS Cloud9
   1. В Консоли управления AWS перейдите в консоль AWS Cloud9 и нажмите кнопку Создать среду.



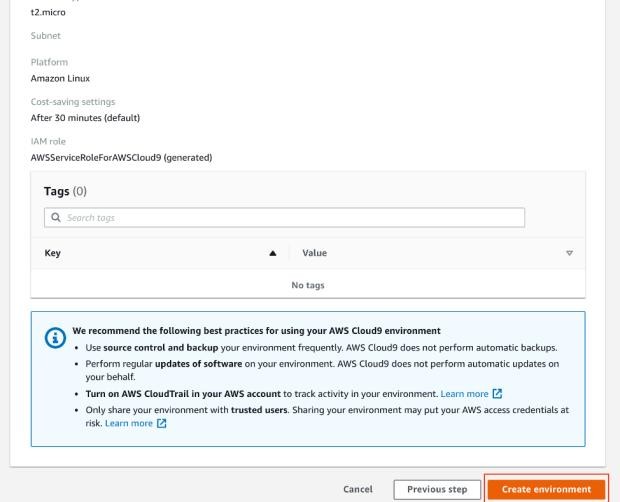
* 1. Введите название DocumentDBCloud9.



* 1. Нажмите кнопку Следующий шаг.

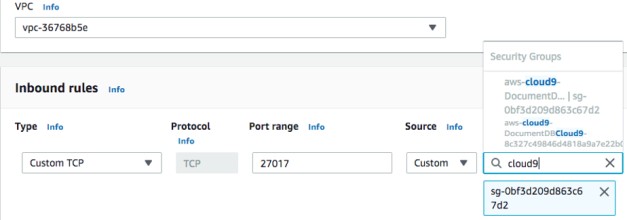


* 1. В разделе Настройки конфигурации примите все параметры по умолчанию.
  2. Нажмите кнопку Следующий шаг.
  3. В разделе Обзор, нажмите кнопку Создать среду.



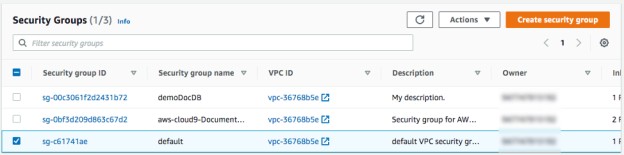
* 1. Создание среды AWS Cloud9 может занять до трех минут.

1. Создание группы безопасности
   1. В консоли управления Amazon EC2 в разделе Сеть и безопасность выберите Группы безопасности.
   2. Нажмите Создать группу безопасности.
   3. В поле Название группы безопасности введите *demoDocDB*.
   4. В поле Описание опишите группу.
   5. В поле VPC примите использование вашего частного виртуального облака по умолчанию.
   6. В разделе Правила для входящих подключений нажмите Добавить правило.
   7. В поле Тип выберите Настраиваемое правило TCP.
   8. В поле Диапазон портов введите 27017.
   9. Исходная группа безопасности – это группа для среды AWS Cloud9, которую вы только что создали. Под информацией о параметре Источник выберите стандартное значение Настраиваемый. Введите cloud9 в поле, расположенное рядом со значением Настраиваемый, чтобы увидеть список всех доступных групп безопасности.



* 1. Выберите группу безопасности aws-cloud9-*<environment name>*.
  2. Примите все параметры по умолчанию и нажмите кнопку Создать группу безопасности. Правила для исходящих подключений менять не требуется.

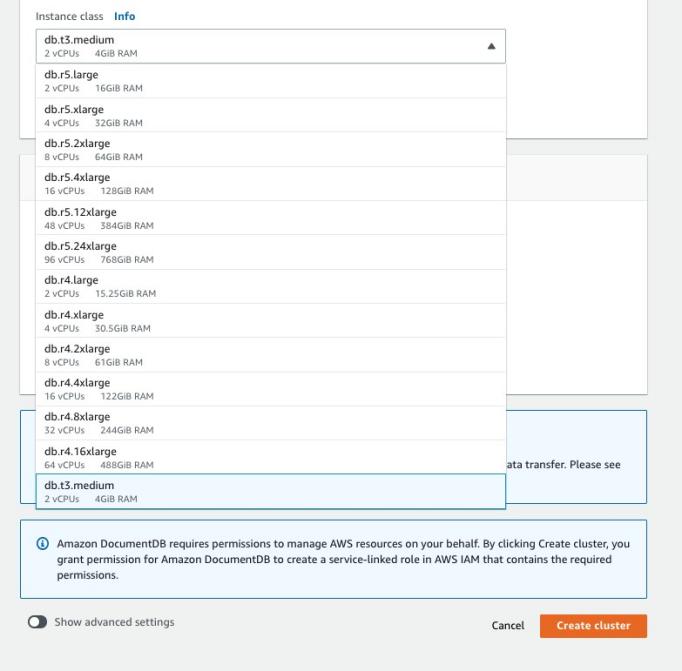
*На скриншоте показаны группы безопасности, которые вы создали на этом этапе. Также вы можете видеть группу AWS Cloud9, которая появилась после создания среды AWS Cloud9.*



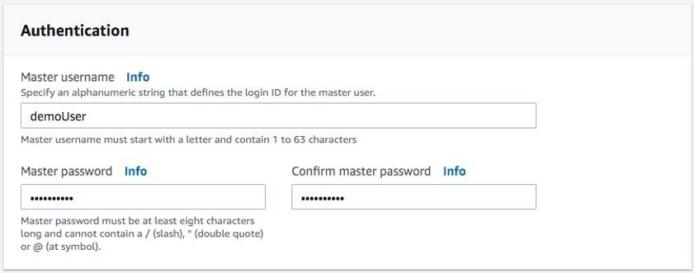
1. Создание кластера Amazon DocumentDB
   1. В консоли управления Amazon DocumentDB в разделе Кластеры нажмите кнопку Создать.

getting-started-amazon-documentdb-with-aws-cloud9-8

* 1. На странице Создание кластера Amazon DocumentDB выберитеdb.t3.medium в поле Класс инстанса, а затем укажите 1 в поле Количество инстансов. Это позволит сократить затраты.



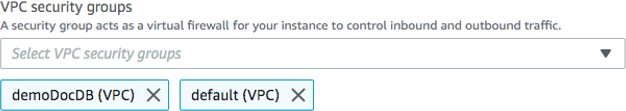
* 1. Другие настройки оставьте по умолчанию.
  2. В разделе Аутентификация введите имя пользователя и пароль.



* 1. Активируйте параметр Показывать расширенные настройки.

getting-started-amazon-documentdb-with-aws-cloud9-11

* 1. В разделе Настройки сети выберите demoDocDB в поле Группы безопасности VPC.



* 1. Нажмите кнопку Создать кластер.

*Сервис Amazon DocumentDB запустил процесс создания кластера, который может занять несколько минут. Вы можете подключиться к кластеру, когда у кластера и инстанса будет*

*отображаться состояние «Доступно». В то время как Amazon DocumentDB создает кластер, вы можете завершить оставшиеся шаги, чтобы подключиться к кластеру.*

1. Установка оболочки Mongo
   1. Если ваша среда AWS Cloud9 открыта, перейдите к шагу 3.
   2. В консоли управления AWS Cloud9 выберите DocumentDBCloud9 в разделе Ваши среды.
   3. Выберите Открыть IDE.
   4. Вставьте следующий код в открытую командную строку, чтобы создать файл репозитория: *echo -e "[mongodb-org-3.6] \nname=MongoDB Repository\nbaseurl=https://repo.mongodb.org/yum/amazon/2013.03/mongodb- org/3.6/x86\_64/\ngpgcheck=1 \nenabled=1 \ngpgkey=htt[ps://w](http://www.mongodb.org/static/pgp/server-)ww[.mongodb.org/static/pgp/s](http://www.mongodb.org/static/pgp/server-)erv[er-](http://www.mongodb.org/static/pgp/server-) 3.6.asc" | sudo tee /etc/yum.repos.d/mongodb-org-3.6.repo*
   5. По завершении установите оболочку Mongo с помощью этого кода:

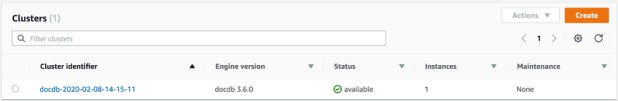
*sudo yum install -y mongodb-org-shell*

* 1. Чтобы шифровать данные при передаче, скачайте сертификат ЦС для Amazon DocumentDB. Используйте для этого следующий код:

*wget https://s3.amazonaws.com/rds-downloads/rds-combined-ca-bundle.pem*

* 1. Теперь вы можете подключиться к кластеру Amazon DocumentDB.

1. Подключение к кластеру Amazon DocumentDB
   1. Найдите свой кластер в консоли управления Amazon DocumentDB в разделе Кластеры. В этом пособии для примера используется кластер docdb-2020-02-08-14-15-11.



* 1. Выберите кластер, который вы создали, нажав на идентификатор кластера (docdb-2020-02- 08-14-15-11 в этом примере).
  2. Скопируйте строку подключения, указанную под Подключение к кластеру с помощью оболочки Mongo.

getting-started-amazon-documentdb-with-aws-cloud9-14Не заполняйте поле *<insertYourPassword>* , чтобы ввести пароль при подключении с помощью оболочки Mongo. Так вам не придется предоставлять пароль открытым текстом в незашифрованном виде.

* 1. Строка подключения выглядит как код, представленный на скриншоте.

getting-started-amazon-documentdb-with-aws-cloud9-15

* 1. При вводе пароля вы должны увидеть подсказку rs0:PRIMARY>. Это означает, что вы успешно подключились к своему кластеру Amazon DocumentDB.

Если у вас возникнут вопросы или проблемы, посетите страницу Поиск и устранение неполадок, связанных с Amazon DocumentDB.

1. Вставка данных и запросы к ним
   1. После подключения к кластеру вы можете создать несколько запросов, чтобы ознакомиться с работой документной базы данных.

Чтобы вставить документ, введите следующий код:

*db.collection.insert({"hello":"DocumentDB"})*

Вы получите такой результат:

*WriteResult({ "nInserted" : 1 })*

* 1. Вы можете прочитать созданный документ с помощью команды findOne(), поскольку она возвращает только один документ. Используйте для этого следующий код: *db.collection.findOne()*

Вы получите такой результат:

*{ "\_id" : ObjectId("5e401fe56056fda7321fbd67"), "hello" : "DocumentDB" }*

* 1. Чтобы создать несколько запросов, воспользуйтесь примером с профилями для разработчиков игр. Сначала введите несколько записей в набор соответствующих профилей. Используйте для этого следующий код:

*db.profiles.insertMany([*

*{ "\_id" : 1, "name" : "Tim", "status": "active", "level": 12, "score":202},*

*{ "\_id" : 2, "name" : "Justin", "status": "inactive", "level": 2, "score":9},*

*{ "\_id" : 3, "name" : "Beth", "status": "active", "level": 7, "score":87},*

*{ "\_id" : 4, "name" : "Jesse", "status": "active", "level": 3, "score":27}*

*])*

Вы получите такой результат:

*{ "acknowledged" : true, "insertedIds" : [ 1, 2, 3, 4 ] }*

* 1. Используйте команду find(), чтобы получить все документы из набора профилей. Используйте для этого следующий код:

*db.profiles.find()*

Вы получите такой результат:

*{ "\_id" : 1, "name" : "Tim", "status" : "active", "level" : 12, "score" : 202 }*

*{ "\_id" : 2, "name" : "Justin", "status" : "inactive", "level" : 2, "score" : 9 }*

*{ "\_id" : 3, "name" : "Beth", "status" : "active", "level" : 7, "score" : 87 }*

*{ "\_id" : 4, "name" : "Jesse", "status" : "active", "level" : 3,*

* 1. Создайте запрос для получения одного документа с помощью фильтра. Используйте следующий код:

*db.profiles.find({name: "Jesse"})*

Вы получите такой результат:

*{ "\_id" : 4, "name" : "Jesse", "status" : "active", "level" : 3, "score" : 27 }*

* 1. Обычный способ, которым пользуются разработчики игр, заключается в поиске профиля определенного пользователя и повышении значений в его профиле. Например, вы хотите провести акцию для самых лучших активных геймеров. Если геймер заполняет опрос, вы добавляете ему 10 баллов.

Для этого используйте команду findAndModify. Допустим, Тим получил и заполнил опрос. Чтобы добавить баллы для профиля Тима, введите следующий код: *db.profiles.findAndModify({*

*query: { name: "Tim", status: "active"}, update: { $inc: { score: 10 } }*

*})*

Вы получите такой результат:

*{*

*"\_id" : 1, "name" : "Tim",*

*"status" : "active", "level" : 12,*

*"score" : 202*

*}*

* 1. Вы можете проверить правильность результата с помощью следующего запроса:

*db.profiles.find({name: "Tim"})*

Вы получите такой результат:

*{ "\_id" : 1, "name" : "Tim", "status" : "active", "level" : 12, "score" : 212 }*

1. Очистка данных

По завершении обзора выключите кластер Amazon DocumentDB или удалите кластер, чтобы не платить за его использование в будущем.

По умолчанию после 30 минут неактивного состояния среда AWS Cloud9 останавливает работу соответствующих инстансов EC2, что также позволяет экономить средства.

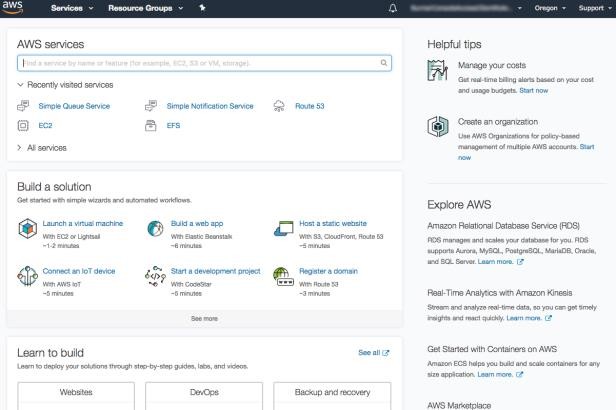
3.4 Хранение и извлечение файла с помощью Amazon S3

Amazon Simple Storage Solution (S3) – это сервис, позволяющий хранить большие объемы данных (называемые *объектами*). С помощью этого пособия вы научитесь создавать корзины Amazon S3, а также загружать в них файлы, извлекать и удалять их.

Действия, описываемые в данном пособии, можно выполнить в рамках уровня бесплатного пользования AWS. Для хранения файлов в AWS требуется аккаунт. Уровень бесплатного пользования AWS включает в себя хранилище объемом 5 ГБ, 20 000 запросов Get и 2000 запросов Put для сервиса Amazon S3.

Шаг 1. Вход в консоль Amazon S3

Щелкните здесь, и в новом окне браузера откроется Консоль управления AWS. При этом руководство останется открытым. Когда стартовый экран загрузится, введите имя пользователя и пароль, чтобы начать работу. Введите *S3* в строке поиска и выберите S3 , чтобы открыть консоль сервиса.

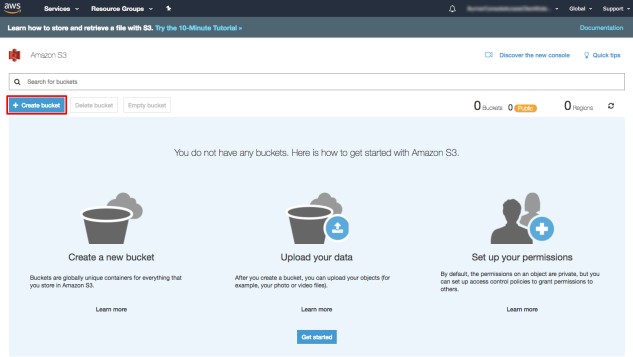


Шаг 2. Создание корзины S3

На этом шаге мы создадим *корзину* Amazon S3. Корзина – это контейнер для хранения файлов.

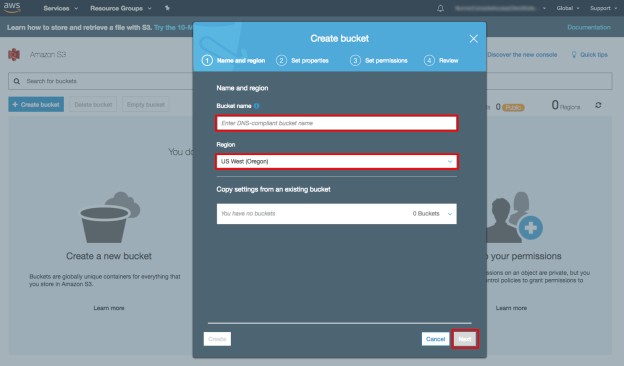
1. На панели S3 нажмите Создать корзину.

Если вы создаете корзину впервые, откроется экран, показанный на следующем изображении. Если ранее вы уже создавали корзины S3, в панели управления S3 будут показаны все имеющиеся корзины.

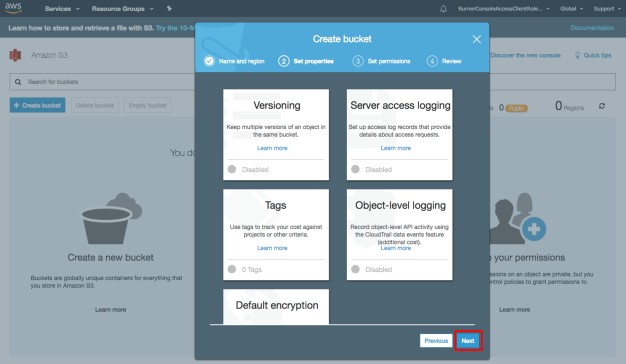


1. Введите имя корзины. Все корзины в Amazon S3 должны иметь уникальные имена. Существует и ряд других ограничений для имен корзин S3. Выберите регион, в котором требуется создать корзину.

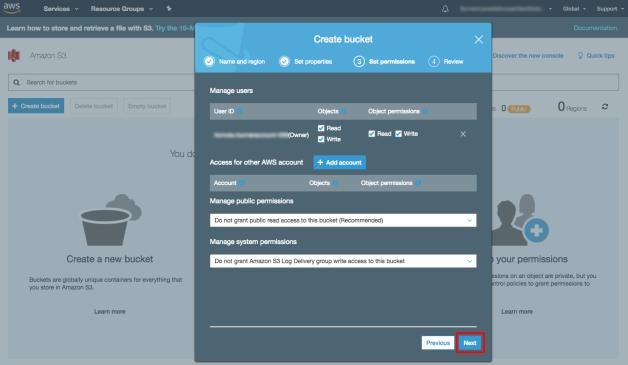
Нажмите Далее.



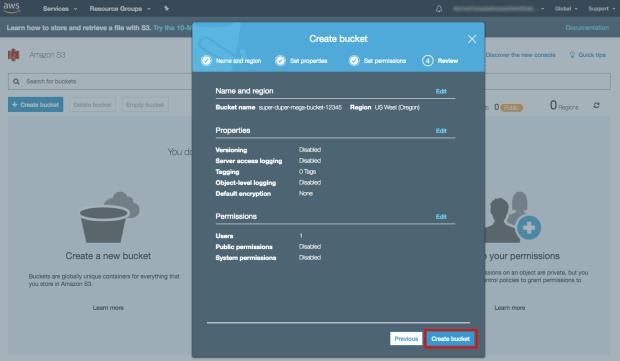
1. Для корзин S3 доступна масса полезных дополнительных возможностей, включая управление версиями, ведение журнала доступа к серверу, теги, ведение журнала на уровне объектов и шифрование по умолчанию. В этом пособии они не будут использоваться. Нажмите Далее.



1. Для корзины S3 можно установить набор разрешений. Оставьте значения по умолчанию и нажмите Далее.



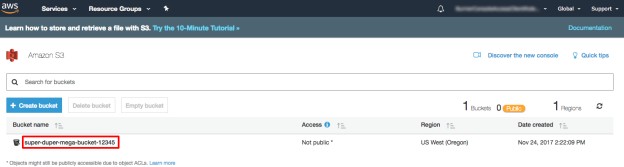
1. Проверьте параметры и нажмите Создать корзину.



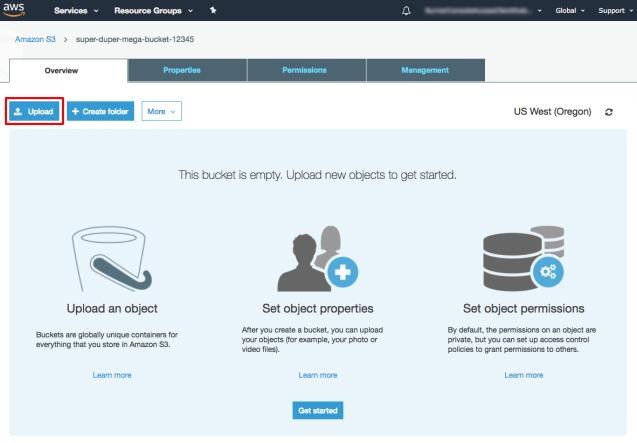
Шаг 3. Отправка файла

На этом шаге отправим файл в новую корзину Amazon S3.

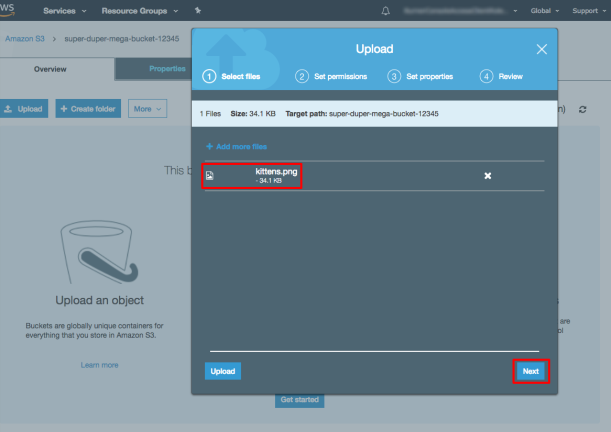
1. Созданная корзина появится в консоли S3. Чтобы перейти к корзине, нажмите ее название.



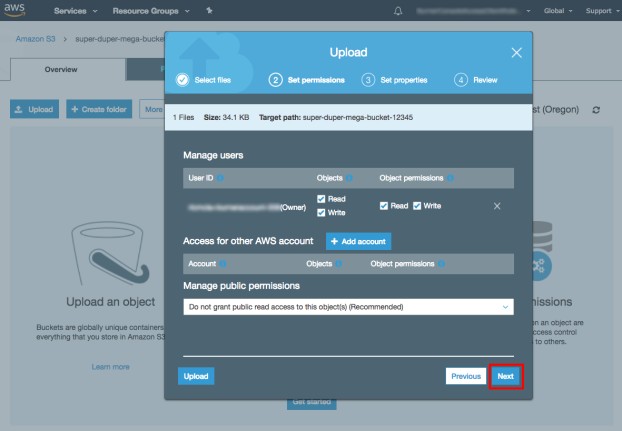
1. Откроется главная страница корзины. Нажмите Отправить.



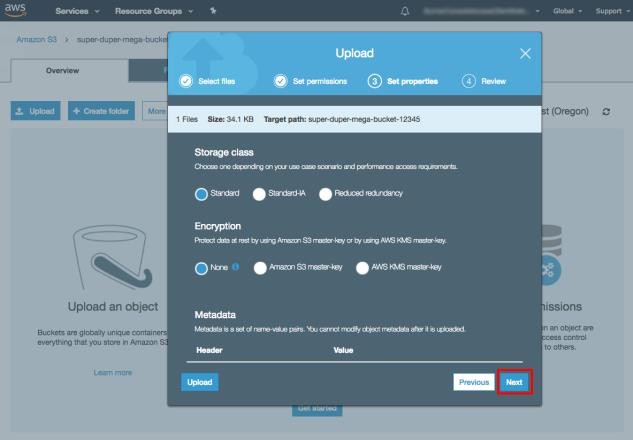
1. Чтобы выбрать файл для отправки, нажмите Добавить файлы и выберите образец файла для сохранения ИЛИ перетащите файл в область отправки. После выбора файла для отправки нажмите Далее.



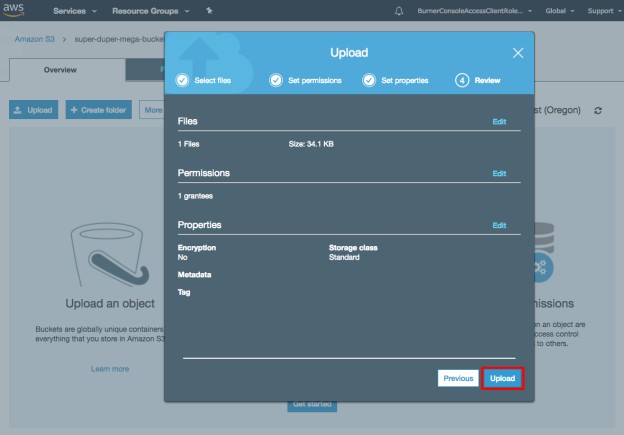
1. Для объекта можно установить набор разрешений. Для целей данного пособия мы оставим значения по умолчанию. Нажмите Далее.



1. Для объекта можно установить набор привилегий, такие как класс хранения, параметры шифрования и метаданные. Оставьте значения по умолчанию и нажмите Далее.



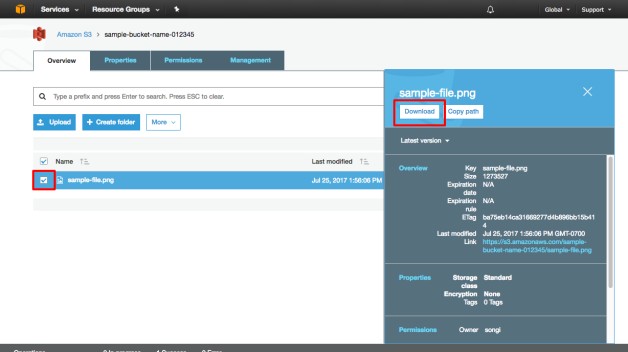
1. Проверьте параметры и нажмите Отправить. Объект появится на главном экране корзины.



Шаг 4. Извлечение объекта

На этом шаге мы скачаем файл из корзины Amazon S3.

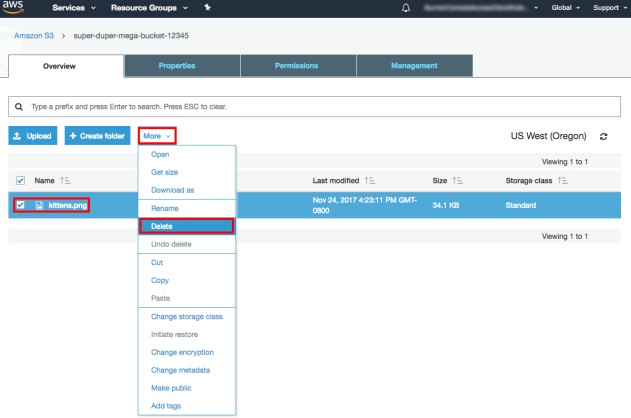
a. Поставьте галочку в поле напротив файла, который хотите скачать из корзины, и нажмите Скачать.



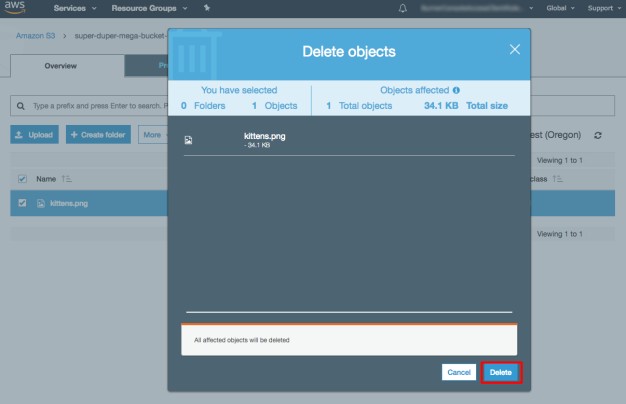
Шаг 5. Удаление объекта и корзины

Объект и корзину можно легко удалить с помощью консоли Amazon S3. Рекомендуется всегда удалять объекты, которые больше не используются, чтобы не платить за их хранение.

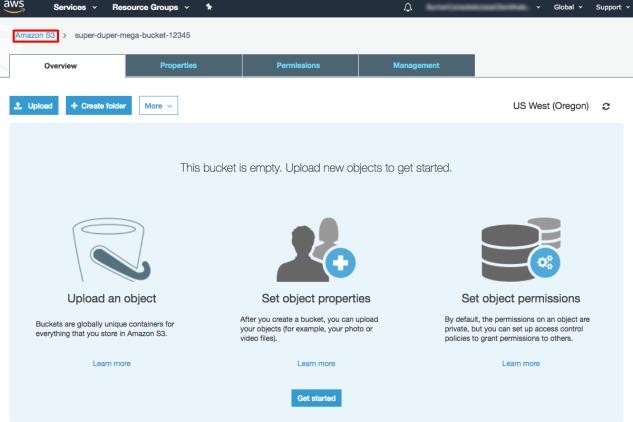
1. Сначала удалим объект. Поставьте галочку в поле напротив файла, который необходимо удалить, и выберите Дополнительно > Удалить.



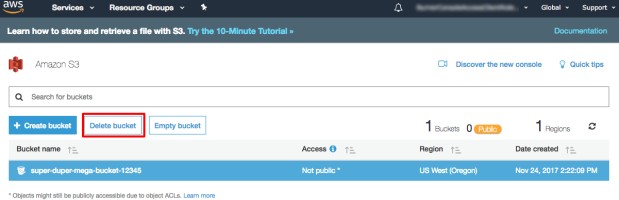
1. Проверьте, что выбрали нужный объект, и подтвердите удаление. Нажмите Удалить.



1. Нажмите на Amazon S3 для просмотра всех корзин в данном регионе.



1. Нажмите справа от имени созданной корзины, чтобы выбрать ее, а затем нажмите Удалить. Введите имя корзины и нажмите Подтвердить.



# 5. ТЕСТИРОВАНИЕ

* 1. План тестирования

План тестирования (тест-план) – это документ, описывающий весь объем работ по тестированию, начиная с описания тестируемых объектов, стратегии, расписания, критериев начала и окончания тестирования, до необходимого в процессе работы оборудования, специальных знаний, а также оценки рисков с вариантами их разрешения. Тест-план является важной составляющей любого грамотно-организованного процесса тестирования, так как содержит в себе всю необходимую информацию, описывающую данный процесс.

Условия для плана тестирования:

* возможность решения задач по тестированию;
* построение стратегии тестирования;
* возможность вести учёт всех требуемых ресурсов, как технических, так и человеческих;
* планирование использования ресурсов на тестирование;
* просчёт рисков, возможных при проведении тестирования.

Чек-лист плана тестирования:

* скорость загрузки страниц (интерфейса ПО);
* адаптивность;
* язык и система исчислений, формат времени представлены в привычном формате;
* удобный шрифт и качество контента;
* основной функционал понятный и его не нужно искать;
* отсутствие горизонтальной полосы прокрутки;
* однообразие интерфейса, есть карта сайта;
* присутствие информации о компании и есть рабочее лого;
* нет раздражительных элементов для пользователей (например, реклама со звуком);
* страница 404 сообщает время и причину ошибки, содержит информацию об основных разделах и контактах;
* удобство пользования функционалом;
* удобство пользования функционалом:
* мгновенный скроллинг (кнопки вверх/вниз на длинных страницах);
* варианты выбора динамически изменяются (при начале ввода есть список доступных результатов, также есть возможность не выбирать из списка, если нет подходящего варианта);
* всё, что можно сделать программно, делается;
* присутствует удобный поиск на сайте или в приложении;
* удобство кнопок (нажимается сама кнопка, а не текст на ней или близлежащая область);
* отсутствуют яркие вызывающие цвета и цветной текст;
* цвет зависит от восприятия подсознательного значения (красный – ошибка, зеленый – всё хорошо);
* оптимальный размер активных элементов (ссылки, баннеры, кнопки);
* пояснения всплывают, где это необходимо (например, почему кнопка/ссылка неактивна, не нужно их совсем скрывать);
* строка поиска, контакты, подзаголовки на странице расположены по схеме в виде буквы F (так скользит взгляд пользователя);
* оформление ссылок в однообразных цветах и стиле по всему приложению;
* единственное поле поиска, помощь при отсутствии результатов;
* возможна авторизация через социальные сети;
* есть удобная обратная связь с подтверждением вопроса, отправкой ответа;
* модуль поддержки с человеком, который также отображает время ожидания ответа;
* разбивка на тематические категории.
* обязательные для ввода поля регистрациия пользователя выделены, минимальное, но необходимое количество данных;
* простота заполнения полей регистрациия пользователя (использование подсказок);
* проверка валидности данных ввода по заполнению до отправки формы;
* информационное сообщение об ошибке (также выделение нужного поля);
* функция просмотра пароля;
* подтверждение регистрации;
* есть возможность отказаться от рассылок.
* единообразие в размерах однотипных форм и цветов;
* полезность (изображение с информацией или элемент дизайна);
* высокое качество.

Тестирование проводилось параллелльно с практической реализацией.

* 1. Тест-кейс

Тест-кейс № 1 приведён в таблице № 1.

Таблица № 1

|  |  |
| --- | --- |
| Название | Тест доступности сервиса |
| Функция | Соединение |
| Действие: открыть сайт  htttps:// | Ожидаемый результат: сайт открыт и доступен.  Фактический результат: сайт открыт и доступен |
| Действие: возврат на шаг  назад | Ожидаемый результат: отсутствие ошибки соединения.  Фактический результат: отсутствие ошибки соединения |
| Результат | Тест пройден |

Тест-кейс № 2 приведён в таблице № 2.

Таблица № 2

|  |  |
| --- | --- |
| Название |  |
| Функция |  |
| Действие |  |
| Действие |  |
| Результат |  |

* 1. Баг-репорты

Баг-репорты отсутствуют, т. к. отсутствуют ошибки.

*или*

Баг-репорт № 1:

|  |  |
| --- | --- |
| Краткое описание | Авторизация на главной странице с использованием кириллицы работает не правильно |
| Проект | https://aws.amazon.com/ |
| Компонент приложения | Страница авторизации |
| Номер версии | 0.001 |
| Важность:  S1 Блокирующая (Blocker)  S2 Критическая (Critical) S3 Значительная (Major) S4 Незначительная (Minor) | S3 Значительная (Major) |

|  |  |
| --- | --- |
| S5 Тривиальная (Trivial) |  |
| Приоритет:  P1 Высокий (High) P2 Средний (Medium) P3 Низкий (Low) | P2 Средний (Medium) |
| Статус | Новая |
| Автор | Ф. И. О. |
| Назначен на | Должность «Программист» |
| Шаги воспроизведения | Открываем главную страницу сайта https://aws.amazon.com/=> слева страницы находим раздел: «авторизация»(см. копию экрана):    Введите логин (например, «user»). Нажмите кнопку «Вход» |
| Фактический результат | Запрос не прошел валидацию. (смотри копию экрана): |
| Ожидаемый результат | Поиск прошел удачно, описание требуемого показано верно. |

* 1. Отчёт о тестировании

В отчёте о тестировании суммированы цели и результаты тестирования (описанные в плане тестирования).

Отчёт о тестировании имеет следующую структуру.

* Обозначение продукта тестирования;
* Вычислительные системы, использованные при тестировании (технические средства, программные средства и их конфигурация);
* Дата окончания тестирования;
* Результаты тестирования описания продукта, документации, программ и данных;
* Перечень несоответствий требованиямл, либо перечень несоответствий рекомендациям, либо перечень не учтенных в продукте рекомендаций, либо формулировка того, что продукт не был протестирован на соответствие рекомендациям;
* Формулировка, что результаты тестирования относятся только к протестированным компонентам продукта;
* Формулировка, что полная копия отчета о тестировании не может быть изготовлена без письменного разрешения тетировшиков.

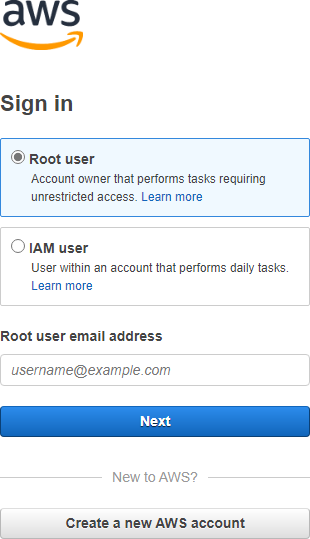
Отчёт о тестировании приведён в приложении Д.

# 5. РЕЗУЛЬТАТЫ

*Если сервис недоступен – раздел исключается.*

* 1. Создание бесплатного аккаунта Amazon Web Services

*Результаты создания бесплатного аккаунта Amazon Web Services с копиями экрана приведены в приложении А. Если сервис недоступен или аккаунт не требуется – пункт исключается.*



* 1. Результаты создания таблицы NoSQL и выполнения запросов к ней

*Результаты создания таблицы NoSQL и выполнения запросов к ней с копиями экрана приведены в приложении Б. Если сервис недоступен – пункт исключается.*

* 1. Результаты настройки документной базы данных

*Результаты создания таблицы NoSQL и выполнения запросов к ней с копиями экрана.*

*Если сервис недоступен – пункт исключается.*

* 1. Результаты хранения и извлечения файла с помощью Amazon S3

*Результаты хранения и извлечения файла с помощью Amazon S3 с копиями экрана. Если сераис недоступен – пункт исключается.*

* 1. Руководство пользователя

*Разработанное (разработанные) Руководство (руководства) пользователя.*

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование руководства пользователя** | **Приложение** |
| 1. | Руководство пользователя по созданию таблицы NoSQL и  выполнения запросов к ней | А |
| 2. | Руководство пользователя по настройке документной базы данных | Б |
| 3. | Руководство пользователя по хранению. и извлечению файла с  помощью Amazon S | В |

# ЗАКЛЮЧЕНИЕ

Целью разработки данного курсового проекта являлась разработка (подготовка) документации и отчётных форм для внедрения программных средств платформы Amazon Web Services интеграции распределённых приложений, миграции и передачи данных.

В ходе написания данной курсовой работы были получены практические навыки в области разработки (подготовки) документации и отчётных форм для внедрения программных средств платформы Amazon Web Services интеграции распределённых приложений, миграции и передачи данных.

Опыт, полученный при написании работы, подтвердил необходимость в правильной выработки требований к разработке (подготовке) документации и отчётных форм для внедрения программных средств платформы Amazon Web Services интеграции распределённых приложений, миграции и передачи данных.

В рамках курсового проектирования получен опыт создания среды AWS Cloud9 для работы с Amazon DocumentDB. Установлена оболочка Mongo, создан кластер Amazon DocumentDB, и подключение к нему, отправлено несколько запросов, вставлены данные и созданы запросы к документам JSON с помощью Amazon DocumentDB.

Создана резервная копия файла в облаке. Для этого создана корзинв Amazon S3 и отправлен в неё файл как объект S3. Amazon S3 обеспечивает надежность хранения объектов на уровне 99,999999999 %, что позволяет гарантировать доступ к ним в любой момент времени. Также извлечена сохраненная копия файла и удалиена корзина.

Созданы таблицы, запрошены таблицы и элементы, управление которыми осущетвлялось из консоли управления AWS.

Заявленные цели полностью достигнуты, однако, возможный недостаток, не влияющий на итоговый результат, проявляется в виде недочёта оформления поянительной записки.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 «Юрайт», 2020

2 «Юрайт», 2020

3 «Amazon», 2020

4 …….

1. https://
2. https://
3. https://
4. https://

9 ……

# ПРИЛОЖЕНИЕ А

## (обязательное)

Руководство пользователя по созданию таблицы NoSQL и выполнения запросов к ней

# ПРИЛОЖЕНИЕ Б

## (обязательное)

Руководство пользователя по настройке документной базы данных

# ПРИЛОЖЕНИЕ В

## (обязательное)

Руководство пользователя по хранению. и извлечению файла с помощью Amazon S3

# ПРИЛОЖЕНИЕ Г

## (обязательное) Отчёт о тестировании

### Наименование подсистемы

Сервис Amazon S3 облачной платформы Amazon Web Services.

### Краткая характеристика

Сервис. Amazon S3 – это безопасное, надежное облачное хранилище с большими возможностями масштабирования для объектов, составляющих статический веб-сайт. Примеры объектов, которые можно хранить: HTML-страницы, файлы CSS, изображения, видеофайлы и объекты JavaScript. Простой веб-интерфейс упрощает использование объектного хранилища Amazon S3 и обеспечивает передачу данных на хранение и их извлечение из любой точки Интернета. Это означает, что надежный доступ к сохраненным данным обеспечен для всех конечных пользователей.

### Вычислительные системы, использованные при тестировании

Компьютер . Операционная система . Программное обеспечение (например, браузер, Office).

### Дата тестирования

14 декабря 2020 г.

### Состав рабочей группы

1. Ф. И. О. (студент (студентка) группы -\_\_, специальность – « »)

### Выявленные в ходе тестирования ошибки (проблемы, замечания)

Ошибок (проблем, замечаний) не выявлено.

*или*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Ошибка** | **Описание** | **Способ устранения** | **Ошибка устранена (да/нет)** |
| 1 | Инстанс | Причины | Процедура восстановления |  |
|  | перестал | подобных | инстанса, который перестал |
|  | отвечать | проблем, как | отвечать на запросы, зависит от |
|  |  | правило, | типа инстанса (на основе EBS |
|  |  | зависят от | или хранилища инстансов). |
|  |  | особенностей | Сначала изучите информацию, |
|  |  | внутренней | которую инстанс выводит в |
|  |  | конфигурации | консоль, чтобы определить, |
|  |  | инстанса | почему перезагрузка повлияла на |
|  |  |  | инстанс. Иногда информации, |
|  |  |  | выведенной в консоль, |
|  |  |  | достаточно, чтобы определить |
|  |  |  | причину ошибки инстанса. При |
|  |  |  | работе в Консоли управления |
|  |  |  | AWS требуется сделать |
|  |  |  | следующее выберите нужный |
|  |  |  | инстанс; для просмотра |
|  |  |  | связанной информации в меню |
|  |  |  | «Instance Actions» выберите |
|  |  |  | пункт «View System Log». Если |
|  |  |  | вы используете инструменты API |
|  |  |  | Amazon EC2 выполните команду |
|  |  |  | ec2-get-console-output. |
| 2 | Потеряны | Чаще всего | В некоторых случаях служба |  |
|  | данные | восстановить | поддержки AWS Support может |
|  |  | данные в | восстановить часть данных при |
|  |  | хранилище | условии, что работа инстанса не |
|  |  | инстансов | была завершена и базовое |
|  |  | невозможно. | оборудование в порядке. Процесс |
|  |  |  | восстановления данных не |
|  |  |  | гарантирует положительного |
|  |  |  | результата и может занять |
|  |  |  | несколько дней, поэтому не |
|  |  |  | следует полагаться на |
|  |  |  | восстановление данных службой |
|  |  |  | поддержки AWS Support как на |
|  |  |  | основную стратегию резервного |
|  |  |  | копирования. |
| 3 | Ошибки | При переходе | Иногда ошибки корневого тома |  |
|  | корневого | инстанса в | для некорректно |
|  | тома | спящий режим | загружающегося инстанса на |
|  |  | сохраняются | основе EBS можно исправить |
|  |  | данные | вручную. Ручное исправление |
|  |  | оперативной | ошибок является сложным |
|  |  | памяти. В | процессом и не рекомендуется |
|  |  | случае | для пользователей, не |
|  |  | остановки | обладающих опытом системного |
|  |  | инстанс | администрирования. Примеры |
|  |  | отключается, а | действий, которые пользователи |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Ошибка** | **Описание** | **Способ устранения** | **Ошибка устранена (да/нет)** |
|  |  | данные | предпринимают на основе |  |
| оперативной | анализа данных консоли после |
| памяти | ошибки инстанса, включают |
| удаляются. В | выполнение команды fsck на |
| обоих случаях | соответствующем томе, |
| данные из | отключение модуля SELinux и |
| корневого | исправление ошибок в файле |
| тома EBS и | fstab. |
| любых |  |
| подключенных |  |
| томов данных |  |
| EBS |  |
| сохраняются. |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |

Результаты тестирования относятся только к протестированным компонентам продукта.

Полная копия отчета о тестировании не может быть изготовлена без письменного разрешения тетировшиков.