Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет им. Г.И. Носова»

Многопрофильный колледж

УТВЕРЖДАЮ
Директор
_____ / С.А. Махновский
«09» февраля 2022 г.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ

по учебному предмету
ЭК.02 Основы облачных технологий

для обучающихся специальности

09.02.01 Компьютерные системы и комплексы
(базовой подготовки)

Магнитогорск, 2022

ОДОБРЕНО

Предметно-цикловой комиссией «Информатики и вычислительной техники»
Председатель И.Г. Зорина
Протокол № 5 от 19.01.2022

Методической комиссией МпК
Протокол № 4 от 09.02.2022

.

Составитель (и):
преподаватель ФГБОУ ВО «МГТУ им. Г.И.Носова» МпК С.М. Сайдильдина

Методические указания по выполнению практических работ разработаны на основе рабочей программы учебного предмета «ЭК.02 Основы работы с облачными технологиями».

СОДЕРЖАНИЕ

1 ВВЕДЕНИЕ

Важную часть теоретической и профессиональной практической подготовки обучающихся составляют практические занятия.

Состав и содержание практических занятий направлены на реализацию Федерального государственного образовательного стандарта среднего общего образования.

Ведущей дидактической целью практических занятий является формирование учебных практических умений (умений решать задачи по математике, физике, химии, информатике и др.), необходимых в последующей учебной деятельности.

В соответствии с рабочей программой учебного предмета «ЭК.02 Основы работы с облачными технологиями» предусмотрено проведение практических занятий.

В результате их выполнения, у обучающегося должны быть сформированы следующие результаты:

Предметные:

- ПР1 сформированность навыков выполнять автоматизированное развертывание VPC
- ПР2 сформированность навыков разрабатывать группы автоматического масштабирования и балансировщика нагрузки
- ПР3 сформированность умений разрабатывать объекты базы данных.

Метапредметные:

- МР1 умение самостоятельно определять цели деятельности и составлять планы деятельности; самостоятельно осуществлять, контролировать и корректировать деятельность; использовать все возможные ресурсы для достижения поставленных целей и реализации планов деятельности; выбирать успешные стратегии в различных ситуациях;
- МР3 владение навыками познавательной, учебно-исследовательской и проектной деятельности, навыками разрешения проблем; способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания;
- МР5 умение использовать средства информационных и коммуникационных технологий (далее - ИКТ) в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности;

Личностные:

- ЛР9 готовность и способность в образованию, в том числе самообразованию, на протяжении всей жизни, сознательное отношение к непрерывному образованию как условию успешной профессиональной и общественной деятельности
- ЛР13 осознанный выбор будущей профессии и возможностей реализации собственных жизненных планов; отношение к профессиональной деятельности как возможности участия в решении личных, общественных, государственных, общенациональных проблем;- _____.

Выполнение обучающихся практических и/или лабораторных работ по учебному предмету «ЭК.02 Основы работы с облачными технологиями» направлено на:

- обобщение, систематизацию, углубление, закрепление, развитие и детализацию полученных теоретических знаний по конкретным темам учебного предмета;
- формирование умений применять полученные знания на практике, реализацию единства интеллектуальной и практической деятельности;
- формирование и развитие умений: наблюдать, сравнивать, сопоставлять, анализировать, делать выводы и обобщения, самостоятельно вести исследования, пользоваться различными приемами измерений, оформлять результаты в виде таблиц, схем, графиков;
- приобретение навыков работы с различными приборами, аппаратурой, установками и другими техническими средствами для проведения опытов;

- развитие интеллектуальных умений у будущих специалистов: аналитических, проектировочных, конструктивных и др.;

- выработку при решении поставленных задач профессионально значимых качеств, таких как самостоятельность, ответственность, точность, творческая инициатива.

Практические занятия проводятся после соответствующей темы, которая обеспечивает наличие знаний, необходимых для ее выполнения.

2 МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Тема 1.1 Предпосылки перехода в «облака». Обзор парадигмы облачных вычислений.

Практическая работа № 1
Сравнительная характеристика моделей развертывания облаков

Цель:

- объяснить принцип работы облачных хранилищ и дать их общее определение
- сравнить несколько популярных облачных хранилищ
- выяснить, какое из них использовать удобнее всего
- выявить плюсы и минусы «облака» для обычного пользователя
- научиться создавать формы опроса и теста на сервисе Google

Выполнив работу, Вы будете:

уметь:

- различать виды облачных сервисов;
- создавать проходить процедуру авторизации в облачном сервисе Google;
- создавать формы опроса и теста в облачном сервисе Google;
- строить статистику результатов опроса и теста в облачном сервисе Google.

Материальное обеспечение:

MS Windows 7 (подписка Imagine Premium), MS Office 2007, 7 Zip, MS Visual Studio (подписка Imagine Premium), Visual Studio Code.

Задание:

1 составить сравнительную таблицу моделей развертывания облаков;

2 в соответствии с вариантом задания создать анкету или опросник (при необходимости с ответами) в соответствии со следующими требованиями:

- не более трех ответов Да/Нет в вопроснике;
- не менее 20 вопросов в анкете или опроснике;
- согласовать вопросник и тест с преподавателем.

Порядок выполнения работы:

1 составить таблицу в MS Word с основными характеристиками каждого облака;

2 завести логин и пароль в облаке Google;

3 создать форму опроса и форму теста;

4 обработать результаты.

Ход работы:

1 войти в облако Google. Войти в сервис «Документы». Создать форму опроса. Заполнить заголовок. Ввести дополнительные вопросы («ФИО опрашиваемого», «E-mail опрашиваемого», «Дата»). Ввести вопросы из вопросника, ввести ответы на вопросы.

2 проверить форму, при желании добавить оформление. Разослать форму по подгруппе. Получить ответы.

3 создать форму теста. Заполнить заголовок. Ввести дополнительные вопросы («ФИО тестируемого», «E-mail тестируемого», «Дата»). Ввести вопросы из теста, ввести ответы на вопросы. Проверить форму, при желании добавить оформление.

4 разослать форму по подгруппе. Получить ответы.
5 построить статистику результатов. Составить отчет.

Форма представления результата:
Таблица и отчет в электронной форме.

Критерии оценки:
Титульный лист отчета должен содержать всю информацию, необходимую для однозначной идентификации авторов и работы. Для этого на титульном листе указывают название дисциплины, тему и номер работы, вариант задания, номер группы, фамилии и инициалы студентов.

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 1.1 Предпосылки перехода в «облака». Обзор парадигмы облачных вычислений.

Практическая работа № 2 Облачные технологии. Сервис «Яндекс.Диск»

Цель: научиться создавать свой Яндекс.Диск, освоить технологию работы с сервисом Яндекса – Яндекс.Диск.

Выполнив работу, Вы будете:

уметь:

- создавать почтовый ящик Яндекс;
- работать с сервисом Яндекс.Диск;
- загружать документы на сервис Яндекс.Диск;
- делиться ссылкой на свой Яндекс.Диск.

Материальное обеспечение:

MS Windows 7 (подписка Imagine Premium), MS Office 2007, 7 Zip, MS Visual Studio (подписка Imagine Premium), Visual Studio Code.

Задание:

1 создайте свой почтовый ящик;

2 выполните вариант создания собственной папки в облаке.

Порядок выполнения работы:

1 создать Яндекс-аккаунт;

2 создать папку в облаке;

3 сохранить ссылку в документе Word, отправить на проверку.

Ход работы:

1) Создайте свой почтовый ящик.

Если вы хотите иметь 10Гбайт или даже больше памяти на серверах Яндекса для хранения резервных копий информации, размещённой на вашем компьютере, делиться событиями вашей жизни, запечатлёнными на фото и видео, тогда можно воспользоваться облачным сервисом Яндекс.Диск или другими подобными сервисами.

Для этого вам потребуется Яндекс-аккаунт, а точнее электронная почта в Яндексе.

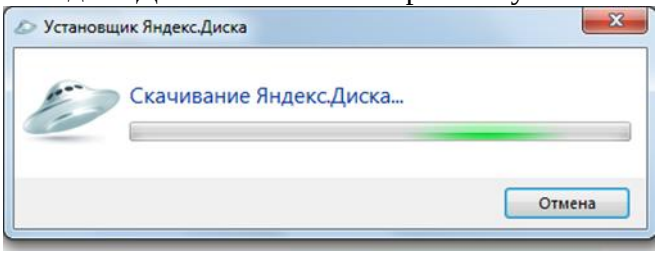
Сервисом Яндекс.Диск можно пользоваться двумя способами:

- Можно заходить в папку Яндекс.Диска по публичной ссылке (публичная ссылка – это ссылка на файлы или папки, предназначенные для общего доступа), отправленной вам преподавателем или другом, и пользоваться данными.
- Можно создавать собственные ресурсы, личные или предназначенные для общего доступа в облаке, установив Яндекс.Диск на свой компьютер.

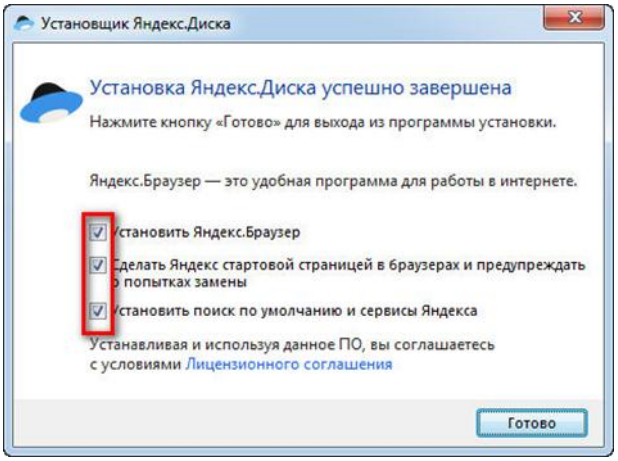
2) Выполните вариант создания собственной папки в облаке:

Для этого:

1. Перейдите на сервис Яндекс.Диска по ссылке http://disk.yandex.ru/

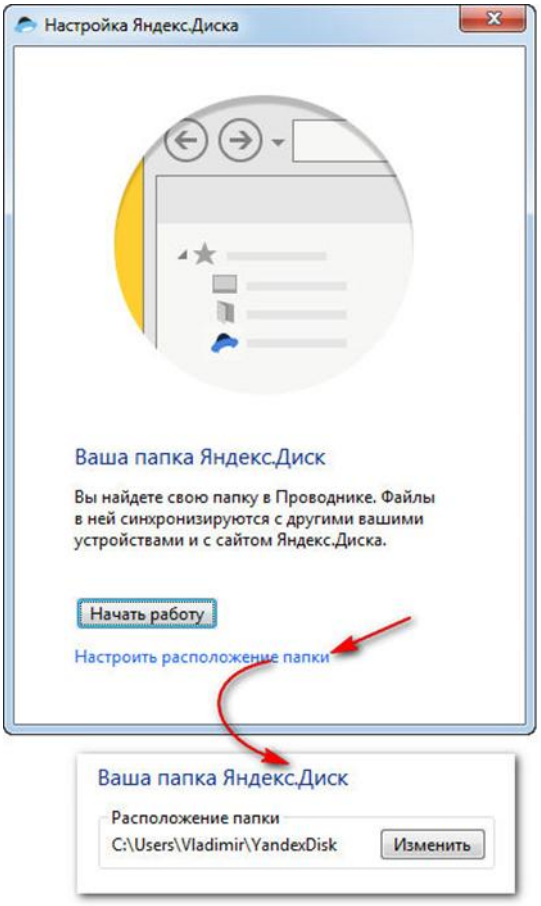


2. Скачайте необходимый для вашей операционной системы файл установки (обычно операционная система определяется автоматически) и установите на компьютер. Выделенные красной рамкой галочки можно отключить.



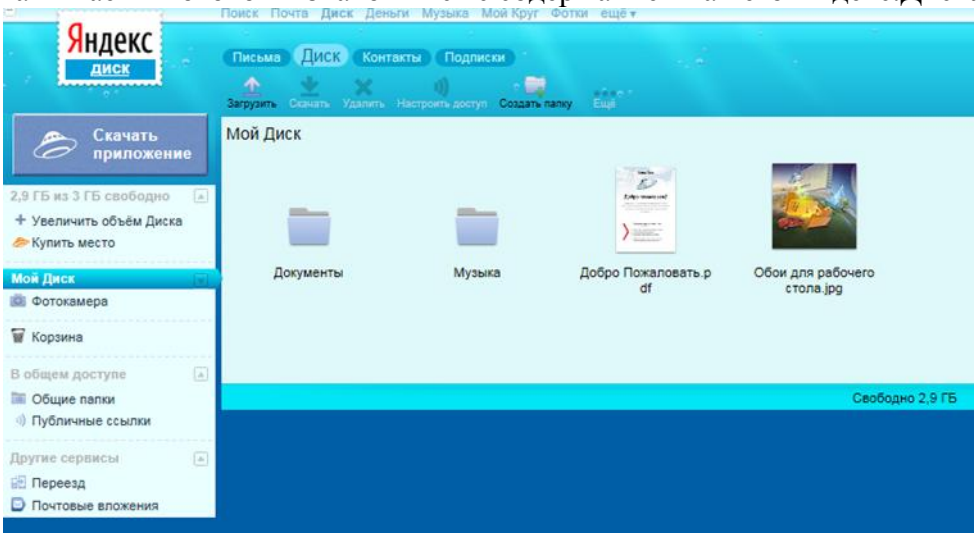
3. После установки программы, вам надо будет войти в аккаунт, используя почтовый ящик от Яндекс почты (логина и пароль от почтового аккаунта).

4. Следующий шаг — это выбор расположения папки для хранения файлов. По умолчанию она располагается на системном диске. Если вы хотите хранить файлы в папке по умолчанию, нажмите кнопочку «Начать работу». Для смены папки нажмите «Настроить расположение папки» и с помощью кнопки «изменить» указываете папку. Нажимаете кнопку «начать работу».

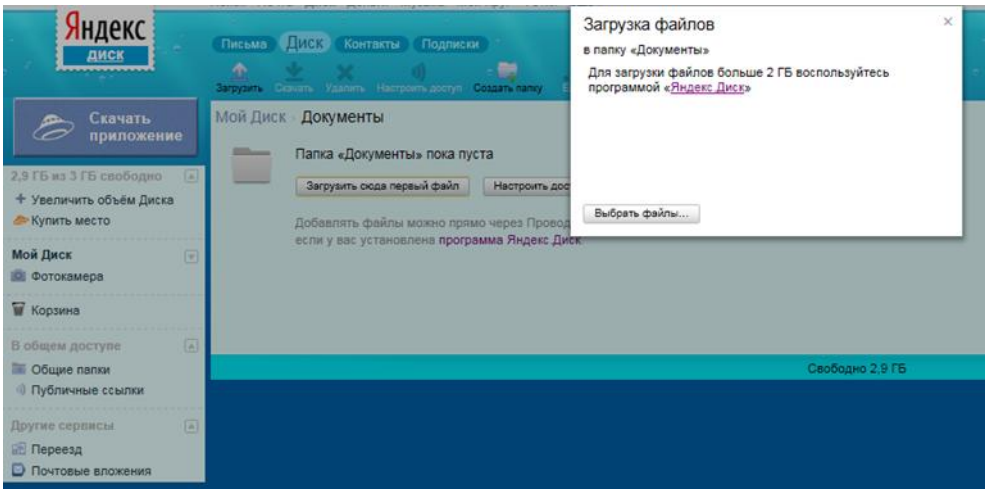


5. Если все настройки были сделаны правильно, программа автоматически начнет синхронизацию данных в облачном хранилище с вашим компьютером. Начнется скачивание всех хранящихся файлов на локальный диск (конечно, если они у вас там уже были). Вы заметите появление нового значка (летающей тарелки) в системном трее (в правом нижнем углу экрана).

6. Ждем. Нажимаем «Готово» и знакомимся с содержанием вашего Яндекс.Диска.



7. Загрузите на Яндекс.Диск любой файл из ваших файлов документов или рисунков.



8. Но, если мы хотим сразу увеличить доступное нам пространство, мы должны разместить в выделенной папке, хотя бы один файл. Загружаем файл.

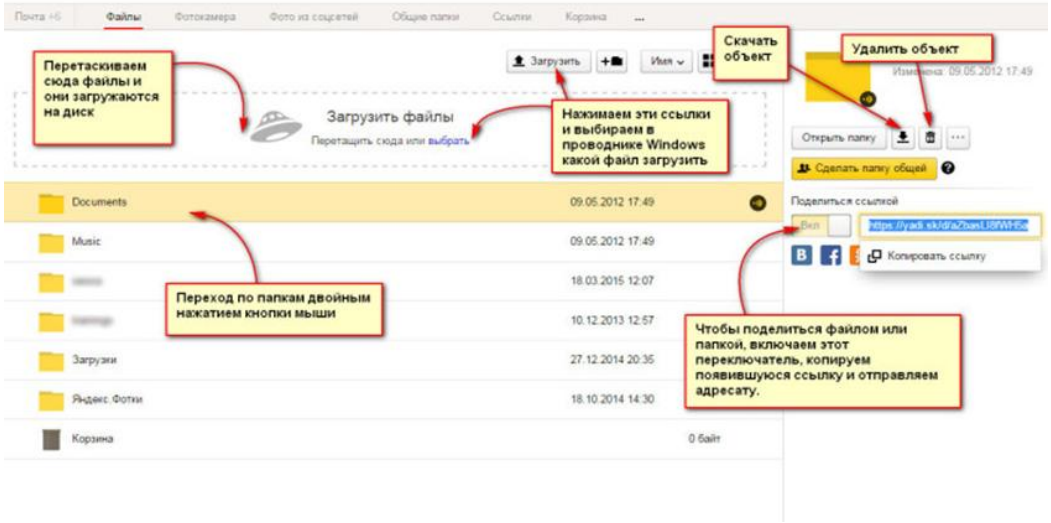
9. По умолчанию, новому пользователю предлагается 10 Гбайт дискового пространства в облаке. Создавая папку в 10 Гбайт вы должны быть уверены, что на вашем диске хватит места для размещения папки такого же размера.

10. Если вы не используете программу на компьютере для управления вашим диском, Яндекс предусмотрел возможность производить все операции с файлами и папками через вэб-интерфейс.

11. Для операций над файлом или папкой, необходимо выделить их. После выделения справа появиться контекстное меню с возможными действиями. Вы можете:

- Скачать файл или папку
- Удалить
- Поделиться (предоставить доступ к файлу/папке другому пользователю скопировав ссылку).

- Поделиться в социальных сетях.



12. Правой кнопкой мыши на значке вашего файла вызовите контекстное меню, выберите команду «поделиться», скопируйте появившуюся ссылку и отправьте ее преподавателю.

Форма представления результата:
Отчет в электронной форме. Ссылка в документе Word.

Критерии оценки:
Титульный лист отчета должен содержать всю информацию, необходимую для однозначной идентификации авторов и работы. Для этого на титульном листе указывают название дисциплины, тему и номер работы, вариант задания, номер группы, фамилии и инициалы студентов.

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 1.1 Предпосылки перехода в «облака». Обзор парадигмы облачных вычислений.

Практическая работа № 3 Облачный сервис Документы Google

Цель: научиться работать в сервисе Документы Google, освоить работу совместного доступа документа для пользователя.

Выполнив работу, Вы будете:

уметь:

- создавать аккаунт Google;
- создавать текстовый Google-документ;
- обеспечивать совместный доступ к документу.

Материальное обеспечение:

MS Windows 7 (подписка Imagine Premium), MS Office 2007, 7 Zip, MS Visual Studio (подписка Imagine Premium), Visual Studio Code.

Задание:

1 создать текстовый google-документ, наполнить его содержимым (2-3 абзаца). Открыть совместный доступ вашего документа.

Порядок выполнения работы:

1 создать аккаунт на сайте Google;

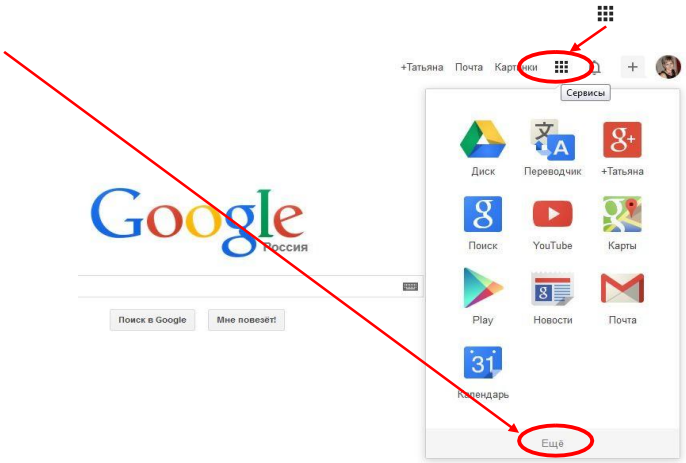
2 создать текстовый google-документ, наполните его содержимым (2-3 абзаца), написать сочинение о себе;

3 перевести ваш текст на английский язык, используя интернет словари;

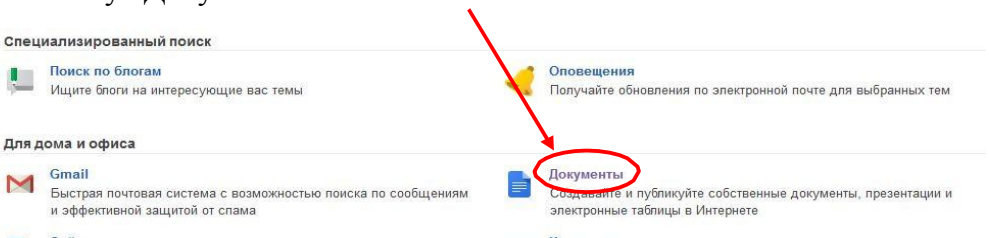
4 открыть совместный доступ для вашего документа для пользователя.

Ход работы:

Для создания Вашего первого текстового документа следует открыть главную страницу поисковой системы Google, расположенную по адресу www.google.ru, в верхней части страницы выбрать значок и нажать ссылку «Еще».

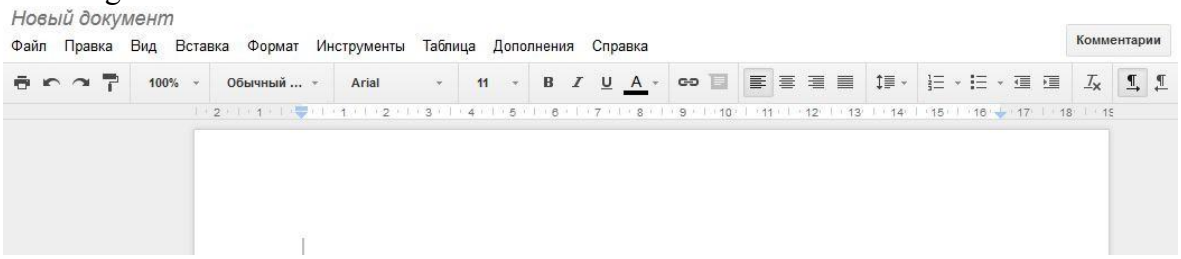


Откроется окно «Google-продукты», в котором следует найти раздел «Для дома и офиса» и выбрать ссылку «Документы».



Откроется новый, пока что пустой текстовый документ, к созданию которого Вы можете приступить. Знакомые всем функции копирования, вырезания и вставки реализуются при помощи команды меню «Правка» или посредствам сочетания клавиш: Ctrl+C – для копирования, Ctrl+X – для вырезания, Ctrl+V – для вставки. Следует заметить, что данные сочетания активны во всех веб-приложениях Google Docs.

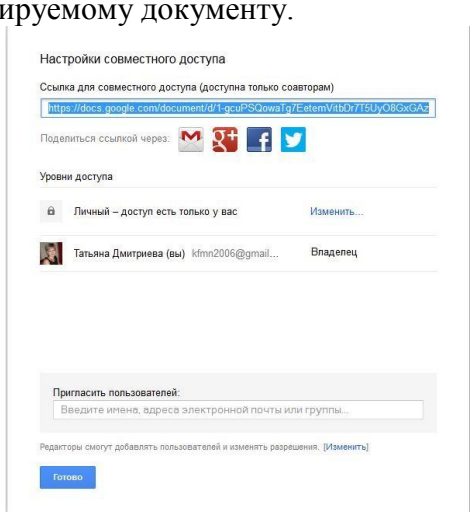
Для форматирования введенного текста воспользуйтесь веб-интерфейсом приложения документы Google.



Следует заметить, что рассматриваемое нами «облачное» приложение снабжено средством проверки орфографии. Никаких отдельных действий по запуску данного модуля не требуется, так как данная функция активирована по умолчанию. Все «незнакомые», а также неправильные слова будут подчеркнуты красной пунктирной линией. Для вывода возможных вариантов замены следует вызвать контекстное меню подчеркнутого слова и выбрать один из предложенных вариантов.

Обращаем внимание на то, что не следует переживать за сохранность введенных данных, любая информация будет автоматически сохранена в виде мгновенной копии. Таким образом, можно вернуть создаваемый документ к одному из первоначальных состояний. Просмотр истории изменений осуществляется при помощи команды меню «Файл» - «Просмотреть историю изменений».

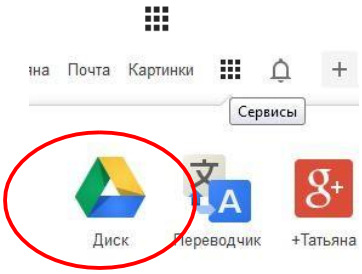
Одним из ярких достоинств «облачных» сервисов можно назвать возможность совместной работы над документами. Вы можете позволить Вашим соавторам просматривать Ваши документы, а при желании и редактировать данные. Для предоставления общего или избирательного доступа к редактируемому файлу, нажмите кнопку «Настройки доступа», расположенную в правом верхнем углу веб-интерфейса программы. Откроется диалоговое окно «Настройки совместного доступа», при помощи элементов управления которого Вы можете настроить права доступа к редактируемому документу.



Вы можете отправить редактируемый документ как вложение в сообщение электронной почты. Для этого выберите команду мены «Файл» -

«Прикрепить к сообщению эл. почты». Откроется диалоговое окно «Отправка сообщения», в котором Вам следует указать электронные адреса получателей (обязательно), а также ввести тему и текст отправляемого сообщения.

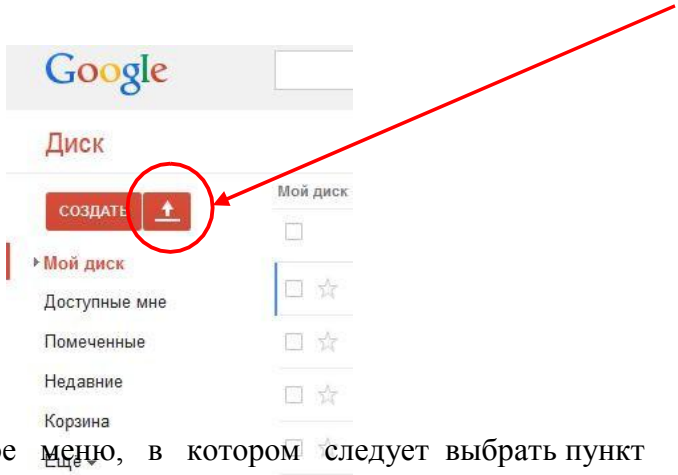
Созданный Вами документ сохраняется на Google-Диске. Посмотреть его содержимое можно следующим образом. Откройте главную страницу поисковой системы Google, расположенную по адресу www.google.ru, в верхней части страницы выберите значок и нажмите ссылку «Диск».



Вы можете загрузить на свой «Google-Диск» документ, созданный на

локальном компьютере. Разумеется, в дальнейшем Вы сможете редактировать его при помощи рассматриваемого приложения «Документы Google».

Для загрузки документа, созданного на локальном компьютере, следует открыть персональную страницу Google Диск и нажать кнопку «Загрузить», расположенную в левом верхнем углу веб-страницы.



Откроется дополнительное меню, в котором следует выбрать пункт

«Файлы». На следующем этапе откроется стандартное диалоговое окно открытия файла «Открыть», в котором Вам следует выбрать загружаемый файл. В следующем диалоговом окне установите флажок «Преобразовывать документы, презентации, таблицы и рисунки в формат Документов Google», так как данная опция необходима для последующего редактирования Вашего файла.

Форма представления результата:

Отчет в электронной форме. Ссылка в документе Word.

Критерии оценки:

Титульный лист отчета должен содержать всю информацию, необходимую для однозначной идентификации авторов и работы. Для этого на титульном листе указывают название дисциплины, тему и номер работы, вариант задания, номер группы, фамилии и инициалы студентов.

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 1.2 Настройки учетной записи - безопасность корневых пользователей.

Практическое занятие № 4 Знакомство с платформой Amazon Web Services.

Цель:

познакомиться с облачной платформой AWS;
изучить основные сервисы AWS;
поработать с калькулятором цен AWS.

Выполнив работу, Вы будете:

уметь:
- осуществлять вход на сервис AWS;
- отличать AWS от других облачных систем;
- находить официальную ссылку на сервис AWS;

Материальное обеспечение:

MS Windows 7 (подписка Imagine Premium), MS Office 2007, 7 Zip, MS Visual Studio (подписка Imagine Premium), Visual Studio Code.

Задание:

1. Составить сравнительную таблицу по продуктам, предоставляемым облачным хранилищам.
2. Рассчитать цену месячной подписки на калькуляторе AWS.
3. Составить список партнеров AWS по рейтингу.

Порядок выполнения работы:

1 Осуществить вход на AWS.
2 Открыть вкладку «Продукты».
3 Составить таблицу свободного вида.
3 Найти на сайте калькулятор цен, рассчитать цену на месяц пользования.
4 Открыть вкладку «Партнерская сеть», изучить, составить список.

Форма представления результата:

Тетрадь с выполненной практической работой, отчёт по заданиям 2 и 3.

Критерии оценки:

Титульный лист отчета должен содержать всю информацию, необходимую для однозначной идентификации авторов и работы. Для этого на титульном листе указывают название дисциплины, тему и номер работы, вариант задания, номер группы, фамилии и инициалы студентов.

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 1.2 Настройки учетной записи - безопасность корневых пользователей.

Практическое занятие № 5 Регистрация пользователя в AWS. Отработка безопасности учетной записи.

Цель:

ознакомиться с условиями регистрации на AWS;
пройти процедуру регистрации;
выполнить первый вход.

Выполнив работу, Вы будете:

уметь:
- осуществлять вход в личный кабинет AWS;
- продумывать безопасную аутентификацию для своего личного кабинета;
- настраивать свою учетную запись;

Материальное обеспечение:

MS Windows 7 (подписка Imagine Premium), MS Office 2007, 7 Zip, MS Visual Studio (подписка Imagine Premium), Visual Studio Code.

Задание:

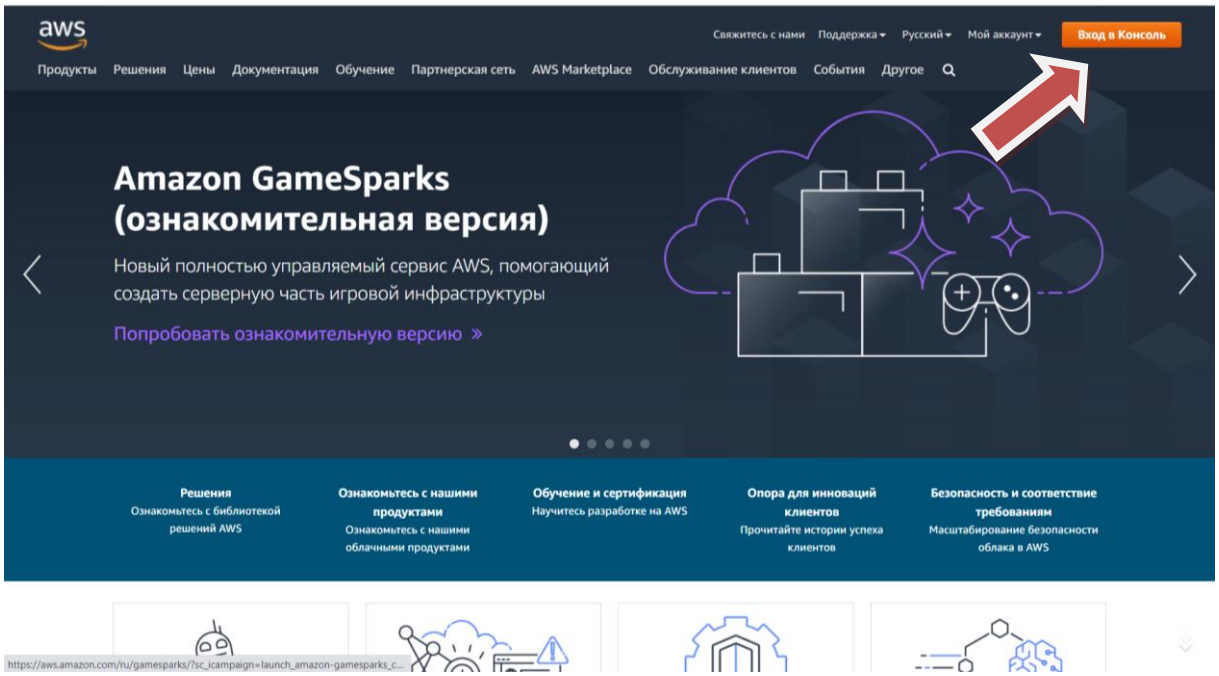
1. Создать учетную запись на облачном сервисе AWS.

Порядок выполнения работы:

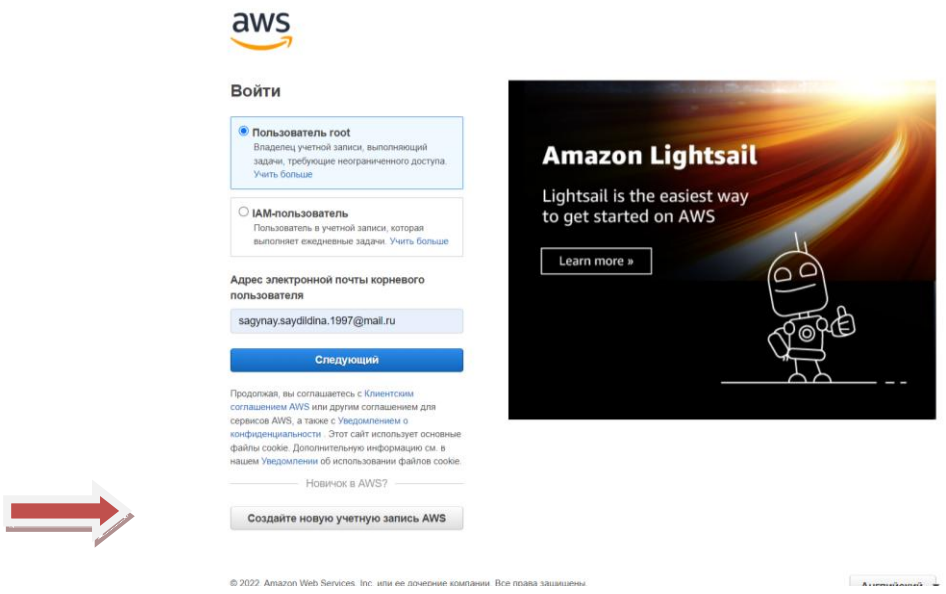
1 Осуществить вход на сервис AWS.
2 Ознакомиться с правилами регистрации.
3 Создать новую учетную запись.

Ход работы:

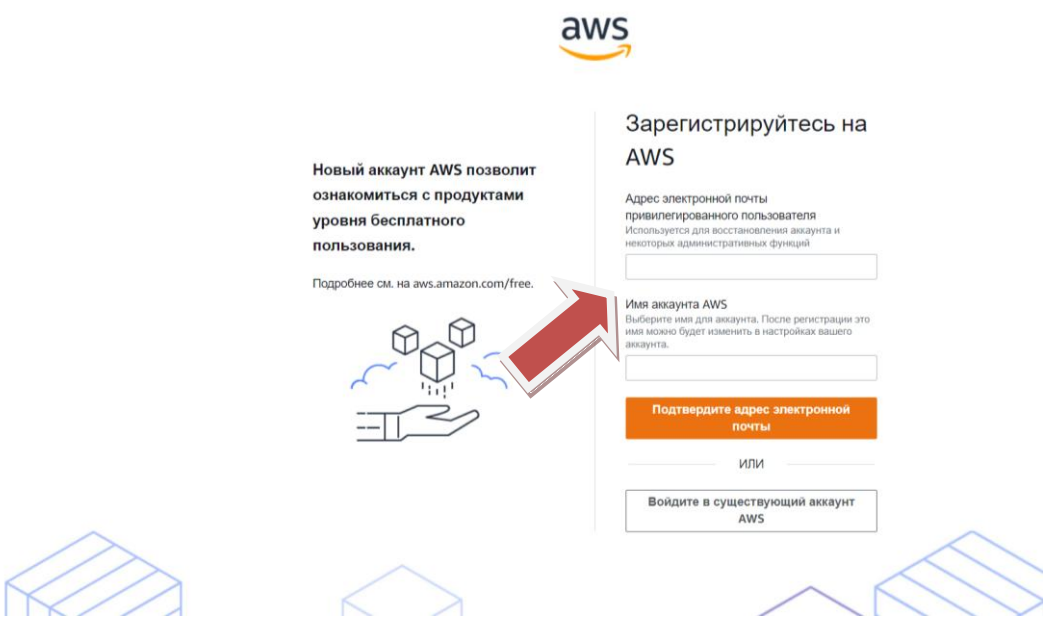
1. Нажать «Вход в консоль»



2. Далее нажать «Создайте ….»



3. Пройти все шаги процедуры регистрации.



Форма представления результата:

Продемонстрировать ход выполнения преподавателю. Показать свою учетную запись. Ответить на вопросы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 2.1 Создание виртуального компьютера и запуск экземпляра Amazon EC2.
Практическое занятие № 6 Создание Elastic IP address.

Цель:
Изучить назначение Elastic IP address;
Создать свой общедоступный адрес;
Познакомиться с понятием VPC и NAT Gateway.

Выполнив работу, Вы будете:
уметь:
создавать общедоступный адрес IPv4;
определять назначение NAT Gateway.

Материальное обеспечение:
MS Windows 7 (подписка Imagine Premium), MS Office 2007, 7 Zip, MS Visual Studio (подписка Imagine Premium), Visual Studio Code.

Задание:
Ваш VPC запускает NAT Gateway для предоставления интернет-доступа к частным ресурсам. Шлюзу NAT Gateway присваивается статический IP- адрес, который называется Elastic IP address. Ваша задача – создать Elastic IP address.

Порядок выполнения работы:
Elastic IP address это общедоступный адрес IPv4, доступ к которому можно получить из Интернета. Статический IP-адрес означает что IP-адрес не изменится. Вы можете связать Elastic IP address с ресурсом в VPC, например, NAT Gateway или экземпляром Amazon EC2. Вы сохраняете контроль над Elastic IP address до тех пор, пока не освободите его.

Ход работы:
1. В AWS Management Console, в меню Services выберите VPC.
2. В навигационной панели слева выберите Elastic IPs.
3. Нажмите Allocate Elastic IP address
4. На странице Elastic IP address settings нажмите Allocate

Отобразится ваш Elastic IP address. Вы будете использовать его в следующей задаче.

Форма представления результата:
Продемонстрировать ход выполнения преподавателю. Показать свою учетную запись. Ответить на вопросы.

Критерии оценки:
«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 2.1 Создание виртуального компьютера и запуск экземпляра Amazon EC2.
Практическое занятие № 7 Создание Amazon VPC. Исследование VPC.

Цель:
Изучить назначение Amazon VPC;
Создать свой VPC;
Познакомиться с видами и назначением подсетей;
Познакомиться с понятием таблица маршрутизации;
Исследовать свой VPC.

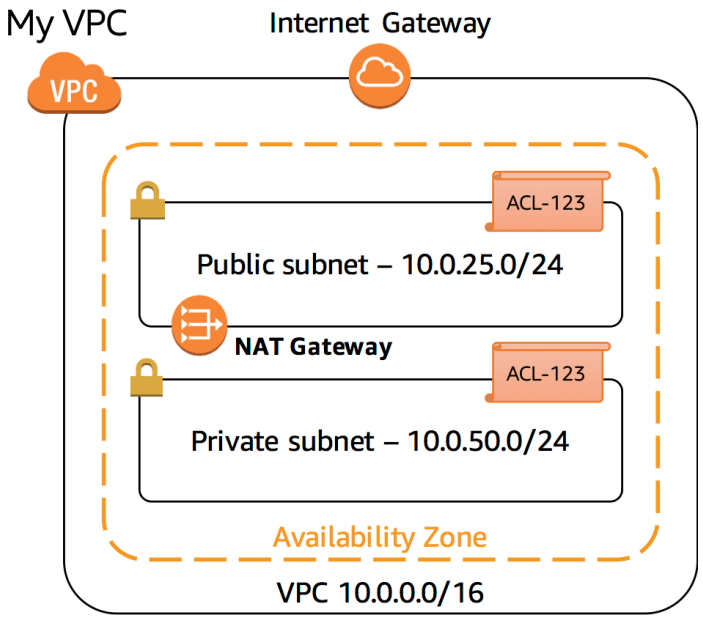
Выполнив работу, Вы будете:
уметь:
создавать VPC со всеми необходимыми характеристиками;
создавать таблицы маршрутизации;
отличать все компоненты VPC.

Материальное обеспечение:
MS Windows 7 (подписка Imagine Premium), MS Office 2007, 7 Zip, MS Visual Studio (подписка Imagine Premium), Visual Studio Code.

Задание:
1. В этой задаче вы создадите Amazon VPC с помощью мастера VPC wizard. Мастер автоматически создает VPC на основе параметров, которые вы указываете. Использование VPC Wizard гораздо проще, чем вручную создавать каждый компонент VPC.

2. Изучить компоненты VPC, созданные VPC Wizard

Ваша задача создать следующую инфраструктуру:



Ход работы:
1. Выберите **VPC Dashboard** в левом верхнем углу.
2. Нажмите **Launch VPC Wizard**

Мастер предлагает четыре заранее определенные конфигурации.

Нажмите на каждую опцию в мастере, чтобы просмотреть их назначение:

- **VPC with a Single Public Subnet (VPC с одной публичной подсетью):** единая публичная подсеть, подключенная к Интернету. Этот вариант идеально подходит для приложений, которые работают исключительно в облаке AWS.
- **VPC with Public and Private Subnets (VPC с публичными и частными подсетями):** публичная подсеть для интернет-ресурсов и частная подсеть для бэк-энд-ресурсов. Для обеспечения доступа в Интернет для ресурсов в частной подсети используется NAT Gateway. Это идеально подходит для изолирования частных ресурсов от Интернета.
- **VPC with Public and Private Subnets and Hardware VPN Access (VPC с публичными и частными подсетями и аппаратным доступом VPN):** публичная подсеть и частная подсеть, а также виртуальное частное подключение к существующему корпоративному центру обработки данных. Этот вариант идеально подходит, когда у вас есть инфраструктура в центре обработки данных, которая может подключиться к облаку AWS в качестве объединенной сети.
- **VPC with a Private Subnet Only and Hardware VPN Access (VPC с только частным подсети и аппаратным доступом VPN):** частная подсеть, подключенная к корпоративному центру обработки данных через VPN-соединение. Этот вариант идеально подходит для того, *чтобы передавать данные в облако AWS,* оставаясь при этом полностью защищенными от доступа в Интернет. Эта конструкция часто используется для разработки и тестирования, где прямой доступ в Интернет не требуется.

Используйте шаблон **VPC with Public and Private Subnets.**

3. Нажмите **VPC with Public and Private Subnets** (второй вариант).
4. Нажмите **Select**

Теперь вам представлены параметры для настройки конфигурации VPC. Настройте следующие настройки, оставив другие поля на значениях по умолчанию:

- **VPC name:** My VPC
- **Public subnet's IPv4 CIDR:** 10.0.25.0/24
- **Public Availability Zone:** Выберите первую зону доступности в списке
- **Private subnet's IPv4 CIDR:** 10.0.50.0/24
- **Private Availability Zone:** Выберите ту же зону доступности, что и public subnet
- **Elastic IP Allocation ID:** Выберите Elastic IP Address, созданный ранее

5. Нажмите **Create VPC**

Теперь будет создан ваш VPC. Окно состояния отображает прогресс. Когда создание VPC завершится, окно состояния покажет успешное создание VPC. Создание может занять несколько минут.

6. Нажмите **OK**, чтобы закрыть окно состояния и вернуться к панели мониторинга VPC.

Ваш созданный VPC теперь отображается в **панели мониторинга VPC**.

7. Скопируйте значение VPC ID вашего VPC и вставьте его в текстовый редактор.

8.В верхнем левом углу, под **Filter by VPC**, выберите поле **Select a VPC**, затем нажмите **My VPC**.

Это ограничивает отображение консоли только для компонентов, связанных с созданным VPC.

9. Если MyVPC не отображается под **Filter by VPC**, обновите страницу, а затем нажмите **Select a VPC** снова.

10. В левой навигационной панели нажмите **Internet Gateways**.

Будет отображаться интернет-шлюз для вашего VPC.

Интернет-шлюз соединяет ваш VPC с Интернетом. Если бы интернет- шлюза не было,

то У VPC не *было бы* подключения к Интернету.

Шлюз Интернета является горизонтально масштабируемым, избыточным и высокодоступным компонентом VPC. Таким образом, он не налагает никаких рисков доступности или ограничения пропускной способности на сетевой трафик.

11. В левой навигационной панели нажмите **Subnets**. Подсеть является подмножеством VPC. Подсеть:

- Принадлежит к определенному **VPC**

- Существует в единой **зоне доступности (в** то время как VPC может охватывать несколько зон доступности)
- Имеет ряд **IP-адресов** (известный как диапазон CIDR, который означает бесклассовую меж доменную маршрутизацию)

Для вашего VPC будут отображаться две подсети: публичная подсеть и частная подсеть.

12. Выберите **Public subnet**.

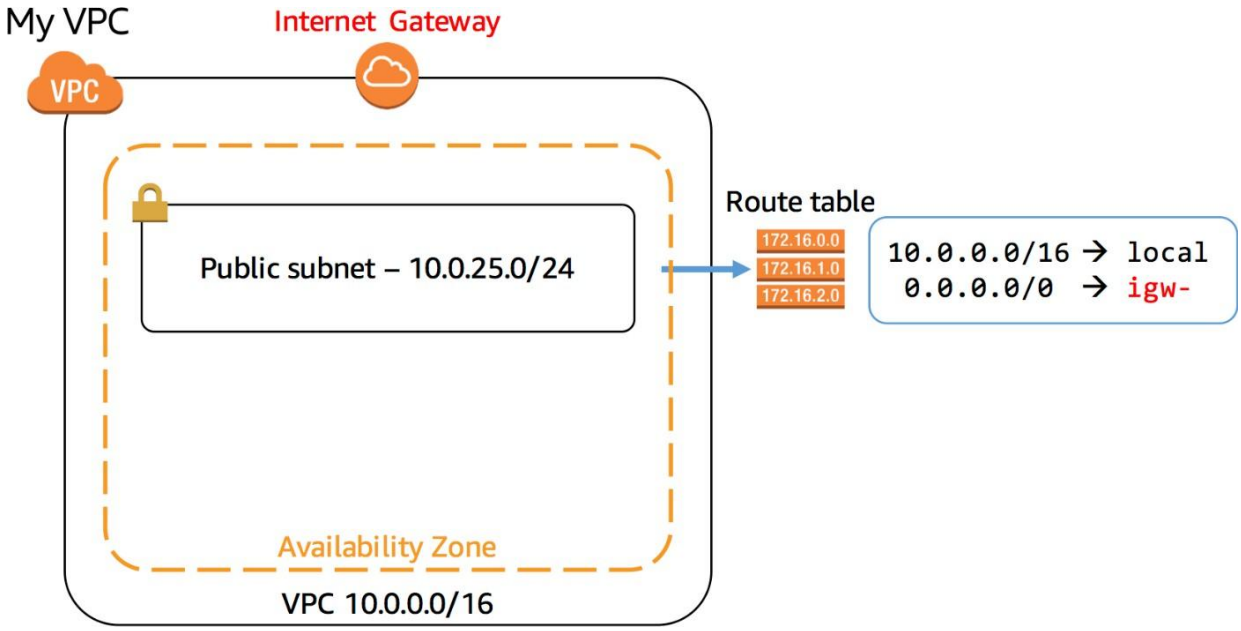
Изучите информацию, отображаемую в нижней части окна:

- Каждой подсети присваивается уникальный **идентификатор подсети.**
- **IPv4 CIDR** *от 10.0.25.0/24 означает,* что подсеть содержит диапазон IP- адресов *от 10.0.25.0* до *10.0.25.255*. (IPv6 также поддерживается)
- Подсеть имеет только 250 **доступных IP-адресов** из 256 возможных адресов. Это связано с тем, что в каждой подсети есть несколько зарезервированных адресов, а один IP-адрес был использован NAT Gateway.

Почему эта подсеть считается публичной? Ответ заключается в маршрутизации.

13. Выберите вкладку **Route table**.

Каждая подсеть связана с **Route table (таблицей маршрутизации)**, в которой указаны маршруты исходящего трафика. Представьте себе адресную книгу, в которой перечислено куда направить трафик в зависимости от его назначения.



- В таблице маршрутов есть два маршрута, связанных с публичной подсетью:
- **Route 10.0.0.0/16 | local** направляет трафик, предназначенный для других мест в VPC (который имеет диапазон *10.0.0.0.0/16*) локально в пределах VPC. Этот трафик никогда не покидает VPC.
- **Route 0.0.0.0/0 | igw-** направляет весь трафик на интернет-шлюз.

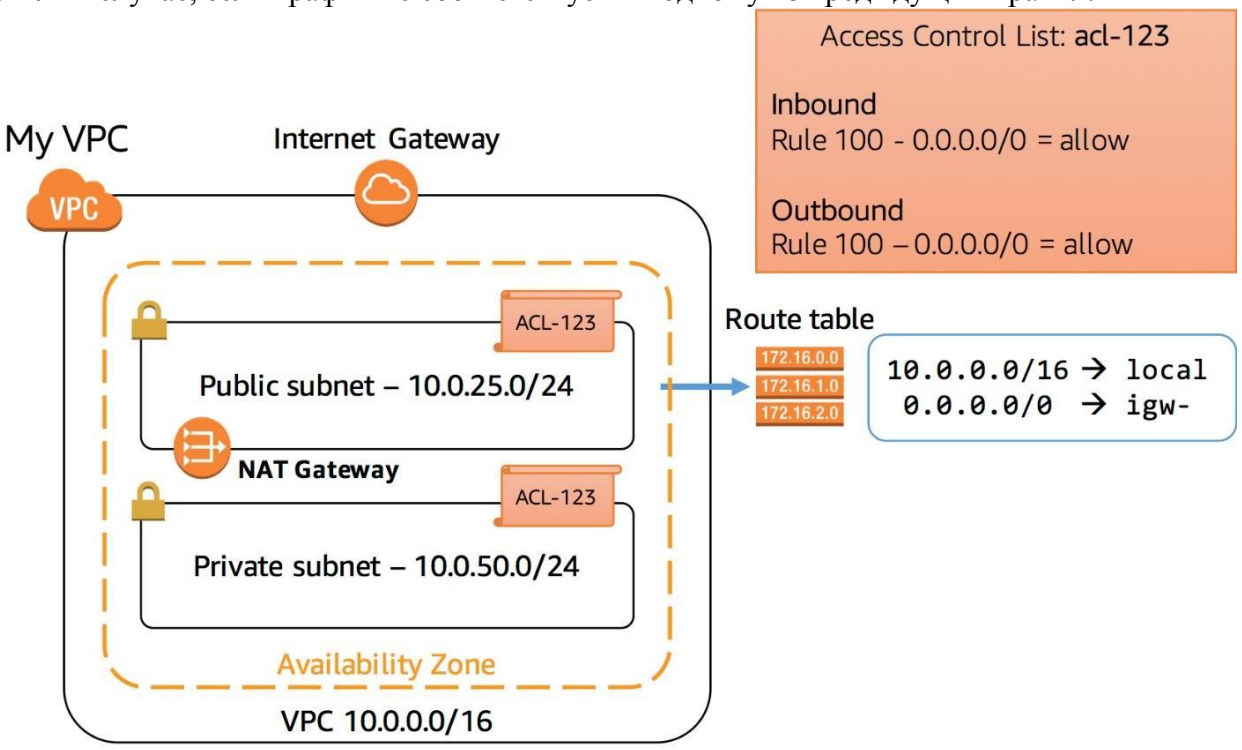
Правила маршрутизации ранжируются от самых ограничительных (с большим числом после слэш) до наименее ограничительных *(что составляет 0.0.0.0/0, так* как это относится ко всему Интернету). Таким образом, трафик сначала отправляется в пределах VPC, если он попадает в диапазон VPC, в противном случае он отправляется в Интернет. Правила могут быть дополнительно отредактированы на основе конкретной конфигурации сети.

Тот факт, что эта подсеть *связана с таблицей маршрутизации, которая имеет маршрут к шлюзу Интернета,* делает ее ***публичной подсетью.*** То есть, *достижимой из Интернета*.

14. Выберите вкладку **Network ACL**.

Список управления доступом к сети (ACL) является дополнительным слоем безопасности для вашего VPC, который выступает в качестве брандмауэра для управления трафиком в и из подсетей. Сетевые ACL обычно остаются с настройками по умолчанию, которые разрешают весь трафик в и из подсетей:

- **Rule 100 Inbound** разрешает весь входящий трафик в **Public Subnet**.
- **Rule 100 Outbound** разрешает весь исходящий трафик из **Public Subnet.**
- Вторая строка в каждом правиле – звездочка, которая действует как правило *catch-all* в случае, если трафик не соответствует ни одному из предыдущих правил.



15. В левой навигационной панели нажмите **Subnets**.

16. В верхней части окна выберите **Private subnet** and и убедитесь, что она является единственной выбранной строкой.

17. Нажмите на вкладку **Tags**.

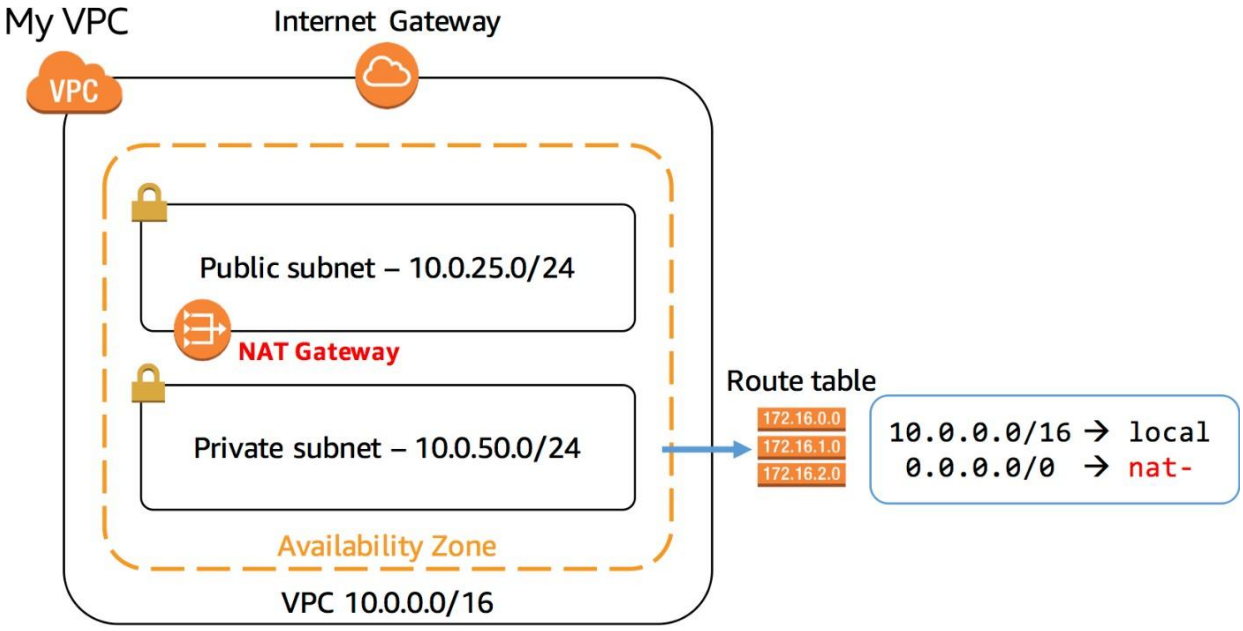
Подсеть была помечена ключом **Name** с значением **Private subnet**. Теги помогают управлять и идентифицировать ресурсы AWS.

18. Нажмите вкладку **Route Table**.

Таблица маршрутизации для частной подсети имеет конфигурацию:

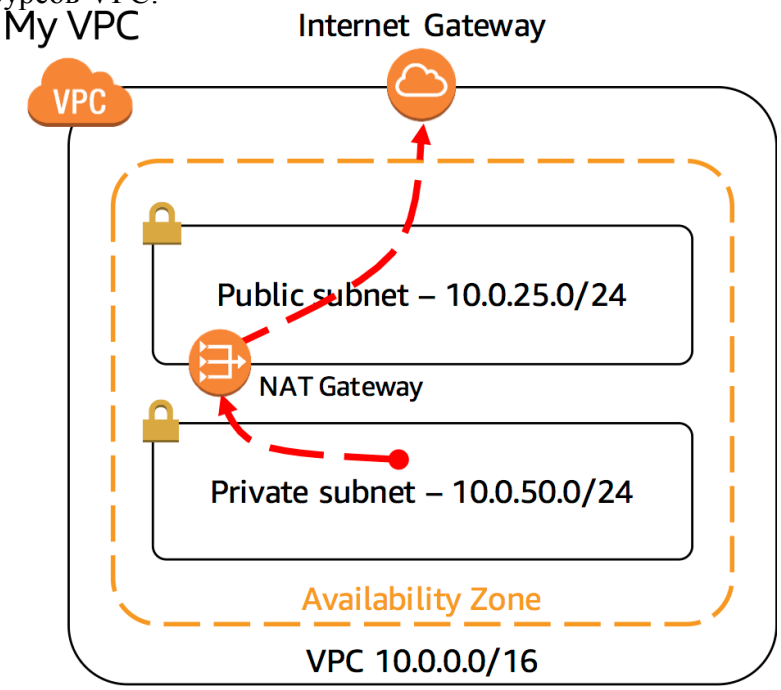
- **Route 10.0.0.0/16 | local** как и публичная подсеть.
- **Route 0.0.0.0 | nat-** направляет трафик на шлюз NAT.

Эта подсеть не имеет маршрута к интернет-шлюзу. Таким образом, это ***частная подсеть***.



19. В левой навигационной панели нажмите **NAT Gateways**.
Отображается шлюз NAT.

Network Address Translation (NAT) Gateway позволяет ресурсам в частной подсети подключаться к Интернету и другим ресурсам за пределами VPC. Это подключение только *для исходящих, что* означает, что соединение должно быть инициировано из частной подсети. Ресурсы в Интернете не могут инициировать входящие соединения. Таким образом, это средство ограничения частных ресурсов и повышения безопасности ресурсов VPC.



20. В левой навигационной панели нажмите **Security Groups**.

21. Выберите группу безопасности, которая совпадает с **VPC ID** который вы скопировали в текстовый редактор и выберите вкладку **Inbound rules**.

Группы безопасности выступают в качестве виртуального брандмауэра для ваших ресурсов для управления входящим и исходящим трафиком. При запуске инстанса Amazon EC2 в VPC можно назначить инстансу до пяти групп безопасности. Группы

безопасности действуют на уровне инстанса, а не на уровне подсети. VPC автоматически поставляется с группой безопасности по умолчанию. Если при запуске инстанса Amazon EC2 не указана группа безопасности, то будет использована группа безопасности по умолчанию.

Группа безопасности по умолчанию разрешает доступ *ко всем трафику* связанных ресурсов, но только в том случае, если Источник *является* группой безопасности по умолчанию. Эта самозависимость может показаться

странной, но эта конфигурация просто означает, что любой инстанс EC2, связанный с группой безопасности по умолчанию, может общаться с любым другим инстансом EC2, связанным с группой безопасности по умолчанию. Все остальные движения запрещены. Это очень безопасный параметр по умолчанию, поскольку он ограничивает любой доступ из других ресурсов.

При добавлении ресурсов в VPC можно создать дополнительные группы безопасности, чтобы разрешить желаемый доступ к ресурсам, таким как веб- серверы, серверы приложений и серверы баз данных.

Форма представления результата:

Продемонстрировать ход выполнения преподавателю. Показать свою учетную запись. Ответить на вопросы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 2.1 Создание виртуального компьютера и запуск экземпляра Amazon EC2.
Практическое занятие № 8 Запуск Amazon EC2 Instance.

Цель:
Запустить Amazon EC2 Instance;
Осуществить Мониторинг экземпляра;
Обновить группы безопасности и получить доступ к веб-серверу;
Изменить тип экземпляра и размер хранилища;
Исследовать ограничения EC2.

Выполнив работу, Вы будете:
уметь:
создавать EC2 Instance со всеми необходимыми характеристиками;
производить мониторинг и исследование созданного экземпляра.

Материальное обеспечение:
MS Windows 7 (подписка Imagine Premium), MS Office 2007, 7 Zip, MS Visual Studio (подписка Imagine Premium), Visual Studio Code.

Задание:
В этой задаче вы запустите экземпляр Amazon EC2 с защитой от прерывания. Защита от прерывания предотвращает случайное прекращение экземпляра EC2. Вы развернете свой экземпляр со скриптом пользовательских данных, который позволит вам развернуть простой веб-сервер.

Порядок выполнения работы:
1. Откройте консоль AWS
2. Авторизуйтесь
3. Создайте виртуальное частное облако MyVPC с одной частной и одной публичной подсетями.

Ход работы:
1. В **AWS Management Console** в меню**Services** выберите **EC2**.
2. В правом верхнем углу экрана, активируйте переключатель **New EC2 Experience** если он не активирован по умолчанию.
3. Нажмите **Launch instance** > **Launch instance**.

Шаг 1: Выбор образа виртуальной машины машины Amazon Machine Image (AMI)

Amazon Machine Image (AMI) предоставляет информацию, необходимую для запуска экземпляра, который является виртуальным сервером в облаке. AMI включает в себя:

- Шаблон операционной системы для экземпляра (например, операционная система или сервер приложений с приложениями)

- Разрешения запуска, учетные записи AWS, которые могут использовать AMI для запуска экземпляров
- Объем дискового хранилища, которое присоединяется к экземпляру при его запусках

Список **Quick Start** содержит наиболее часто используемые AMI. Вы также можете создать свой собственный AMI или выбрать AMI из AWS Marketplace, интернет-магазина, где вы можете продавать или покупать программное обеспечение, которое работает на AWS.

4. Нажмите **Select** рядом с **Amazon Linux 2 AMI** (сверху списка).

Шаг 2: Выбор типа экземпляра

Amazon EC2 предоставляет широкий выбор типов *экземпляров, оптимизированных* для различных случаев использования. Типы экземпляров включают различные комбинации процессора, памяти, памяти и сетевых возможностей и дают вам гибкость в выборе соответствующего сочетания ресурсов для приложений. Каждый тип экземпляра включает в себя один или несколько размеров экземпляров, что позволяет масштабировать ресурсы до требований вашей целевой рабочей нагрузки.

5. Прокрутите вниз и выберите **t2.micro.**

t2.micro имеет 1 виртуальный процессор и 1 GIB памяти.

6. Нажмите **Next: Configure Instance Details Шаг 3: Настройка деталей экземпляра**

Эта страница используется для настройки экземпляра в соответствии с вашими требованиями. Это включает в себя настройки сетей и мониторинга.

Network – в какое виртуальное частное облако (VPC) вы хотите запустить экземпляр. Вы можете иметь несколько сетей, таких как различные для разработки, тестирования и производства.

7. Для **Network** выберите **MyVPC**, для **Subnet** выберите публичную подсеть.
8. Для **Enable termination protection**, выберите **Protect against accidental termination**.

Когда экземпляр Amazon EC2 больше не требуется, он может быть *прекращен,* что означает, что экземпляр остановлен и его ресурсы освобождены. Прерванный экземпляр не может быть запущен снова. Если вы хотите предотвратить случайное прекращение работы экземпляра, можно включить *защиту* от прерывания для экземпляра, что предотвращает его прекращение.

9. Прокрутите вниз и раскройте **Advanced Details**.

Появится поле **User data.**

При запуске экземпляра можно передать данные *пользователя в экземпляр,* который может быть использован для выполнения общих автоматизированных задач конфигурации и даже запустить скрипты после запуска экземпляра.

Ваш экземпляр работает под управлением Amazon Linux, поэтому вы предоставите скрипт *оболочки,* который будет работать при запуске экземпляра.

10. Скопируйте следующие команды и вставьте их в поле **User data**:

```
#!/bin/bash
yum -y install httpd systemctl enable httpd systemctl start httpd
echo '<html><h1>Hello From Your Web Server!</h1></html>' > /var/www/html/index.html
```

Скрипт выполнит:

- Установку веб-сервера Apache (httpd)
- Настройку веб-сервера для автоматического запуска при загрузке
- Запуск веб-сервера

- Создание простой веб-страницы 13.Нажмите **Next: Add Storage**

Шаг 4: Добавление хранилища

Amazon EC2 хранит данные на сетевом виртуальном диске под названием *Elastic Block Store.*

Вы запустите экземпляр Amazon EC2 с использованием объема диска 8 GiB по умолчанию. Это будет ваш корневой объем (также известный как "загрузочный" объем).

14. Нажмите **Next: Add Tags Шаг 5: Добавление тегов**

Теги позволяют классифицировать ресурсы AWS по-разному, например, по назначению, владельцу или среде. Это полезно, когда у вас много ресурсов одного типа — вы можете быстро определить конкретный ресурс на основе назначенных им тегов. Каждый тег состоит из ключа и значения, оба из которых вы определяете.

15. Нажмите **Add Tag** и укажите:

- **Key:** Name
- **Value:** Web Server

16. Нажмите **Next: Configure Security Group Шаг 6: Настройка групп безопасности**

Группа безопасности выступает в качестве виртуального брандмауэра, который контролирует трафик в течение одного или нескольких экземпляров. При запуске экземпляра вы связываете одну или несколько групп безопасности с экземпляром. Вы добавляете *правила к* каждой группе безопасности, которые позволяют трафик или из связанных с ним экземпляров. Вы можете изменить правила для группы безопасности в любое время; новые правила автоматически применяются ко всем экземплярам, связанным с группой безопасности.

17. На **Step 6: Configure Security Group**, настройте:

Сохраняя настройки по умолчанию, создайте **новую** группу безопасности.

- **Security group name:** Web Server security group
- **Description:** Security group for my web server

В этом задании вы не будете входить в экземпляр с помощью SSH. Удаление доступа к SSH повысит безопасность экземпляра.

18. Удалите существующее правило SSH. 19.Нажмите **Review and Launch**

Шаг 7: Обзор конфигурации экземпляра

Страница Обзор отображает конфигурацию для экземпляра, который вы запустите.

20. Нажмите **Launch**

Появится окно **Select an existing key pair or create a new key pair**.

Amazon EC2 использует криптографию с открытым ключом для шифрования и расшифровки информации для входа. Чтобы войти в экземпляр, необходимо создать ключ-пару, указать имя пары ключей при запуске экземпляра и предоставить частный ключ при подключении к экземпляру.

В этом задании вы не будете входить в ваш экземпляр, поэтому вам не требуется ключ пара.

НЕОБХОДИМО выбрать **Proceed without a key pair**. Если вы не выберете этот пункт, ваш экземпляр не будет запущен.

21. Нажмите на выпадающий список **Choose an existing key pair** и выберите **Proceed without a key pair**

22. Отметьте **I acknowledge that ....**

23. Нажмите **Launch Instances**

Теперь ваш экземпляр будет запущен.

24. Нажмите **View Instances**

Экземпляр будет отображаться в состоянии *pending ,* что означает его запуск. Затем

он будет изменен *на running,* что указывает на то, что экземпляр начал загрузку. Через короткое время, вы можете получить доступ к экземпляру.

Экземпляр получает общедоступную *public DNS name,* которое можно использовать для связи с экземпляром из Интернета.

Выберите **Web Server** , вкладка **Details** отображает подробную информацию о вашем экземпляре.

Чтобы просмотреть дополнительную информацию во вкладке **Details** , перетащите разделитель окна вверх.

Просмотрите информацию, отображаемую во вкладке **Details**. Она включает в себя информацию о типе экземпляра, настройках безопасности и настройках сети.

25. Подождите, пока ваш экземпляр отобразит следующее:

- **Instance State:** running
- **Status Checks:** 2/2 checks passed

Тема 2.2 Создание группы автоматического масштабирования и балансировщика нагрузки.

Практическая работа № 9 Добавление метрик в панель мониторинга.

Цель:

Запустить Amazon EC2 Instance;
Осуществить Мониторинг экземпляра;
Обновить группы безопасности и получить доступ к веб-серверу;
Изменить тип экземпляра и размер хранилища;
Исследовать ограничения EC2.

Выполнив работу, Вы будете:

уметь:
создавать EC2 Instance со всеми необходимыми характеристиками;
производить мониторинг и исследование созданного экземпляра.

Материальное обеспечение:

MS Windows 7 (подписка Imagine Premium), MS Office 2007, 7 Zip, MS Visual Studio (подписка Imagine Premium), Visual Studio Code.

Задание:

Мониторинг является важной частью поддержания надежности, доступности и производительности экземпляров Amazon Elastic Compute Cloud (Amazon EC2) и решений AWS. Осуществить мониторинг EC2 Instance.

Порядок выполнения работы:

1. Откройте консоль AWS
2. Авторизуйтесь
3. Продолжите выполнение на основе практической № 8.

Ход работы:

26. Выберите вкладку **Status Checks**.

С помощью мониторинга состояния экземпляров можно быстро определить, обнаружил ли Amazon EC2 какие-либо проблемы, которые могут помешать вашим экземплярам работать с приложениями. Amazon EC2 выполняет автоматизированные проверки на каждом запущенном экземпляре EC2 для выявления проблем с оборудованием и программным обеспечением.

Обратите внимание, что проверки **System reachability** и **Instance reachability** прошли успешно.

27. Выберите вкладку **Monitoring.**

Эта вкладка отображает метрики CloudWatch для вашего экземпляра. В настоящее время существует не так много метрик для отображения, поскольку экземпляр был недавно запущен.

Вы можете нажать на график, чтобы увидеть расширенное представление.

Amazon EC2 отправляет метрики в Amazon CloudWatch для экземпляров EC2. Базовый (пятиминутный) мониторинг включен по умолчанию. Вы можете включить подробный (одноминутный) мониторинг.

28. В меню **Actions**, выберите **Monitor and troubleshooting Get system log**.

System Log отображает вывод консоли экземпляра, который является ценным инструментом для диагностики проблем. Это особенно полезно для устранения неполадок и проблем с конфигурацией, которые могут привести к тому, что экземпляр прекратит работу или станет недоступным до того, как его SSH daemon может быть запущен. *Если вы не видите системный журнал, подождите несколько минут, а затем повторите попытку.*

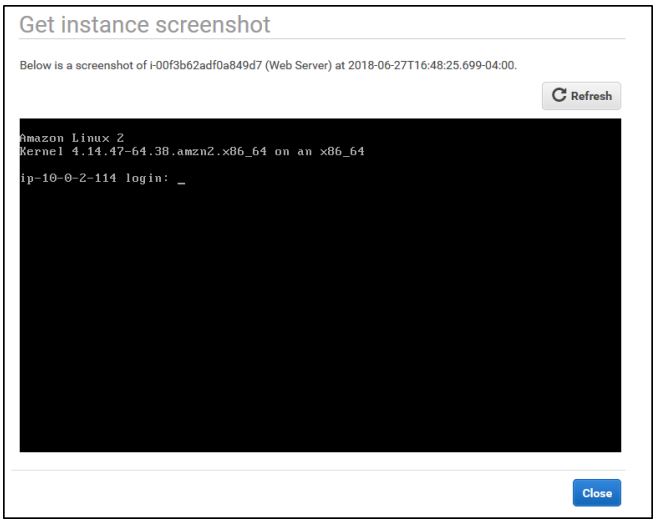
29. Прокрутите вывод и обратите внимание, что пакет HTTP был установлен из **user data**, которые вы добавили при создания экземпляра.

```
[   11.030892] cloud-init[3267]: ================================================================================
[   11.036062] cloud-init[3267]: Package                 Arch       Version
[   11.042524] cloud-init[3267]: ================================================================================
[   11.047523] cloud-init[3267]: Installing:
[   11.050541] cloud-init[3267]: httpd                   x86_64     2.4.33-2.amzn2.0.3
[   11.055701] cloud-init[3267]: Installing for dependencies:
[   11.059198] cloud-init[3267]: apr                     x86_64     1.6.3-5.amzn2
[   11.064186] cloud-init[3267]: apr-util                x86_64     1.6.1-5.amzn2
[   11.068902] cloud-init[3267]: apr-util-bdb            x86_64     1.6.1-5.amzn2
[   11.073681] cloud-init[3267]: generic-logos-httpd     noarch     18.0.0-4.amzn2
[   11.078698] cloud-init[3267]: httpd-filesystem        noarch     2.4.33-2.amzn2.0.3
[   11.083871] cloud-init[3267]: httpd-tools             x86_64     2.4.33-2.amzn2.0.3
[   11.089092] cloud-init[3267]: mailcap                 noarch     2.1.41-2.amzn2
[   11.094112] cloud-init[3267]: mod_http2               x86_64     1.10.18-1.amzn2.0
[   11.099019] cloud-init[3267]: Transaction Summary
[   11.102334] cloud-init[3267]: ================================================================================
[   11.107506] cloud-init[3267]: Install  1 Package (+8 Dependent packages)
[   11.111502] cloud-init[3267]: Total download size: 1.8 M
[   11.114908] cloud-init[3267]: Installed size: 5.0 M
[   11.118468] cloud-init[3267]: Downloading packages:
[   11.670023] cloud-init[3267]: --------------------------------------------------------------------------------
```

30. Нажмите **Cancel**

31. В меню **Actions** выберите **Monitor and troubleshoot Get instance screenshot**.

Будет выведен скриншот, как выглядела бы консоль Amazon EC2, если бы к ней был прикреплен экран.



Если вы не можете связаться с экземпляром через SSH или RDP, вы можете захватить скриншот вашего экземпляра и просмотреть его как изображение. Это обеспечивает видимость состояния экземпляра и позволяет быстрее устранения неполадок.

32. Нажмите **Cancel**

Задание 3: Обновление групп безопасности и получение доступа к веб-серверу

Когда вы запустили экземпляр EC2, вы предоставили скрипт, который установил веб-сервер и создал простую веб-страницу. В этой задаче вы будете получить доступ к содержимому с веб-сервера.

33. Нажмите на вкладку **Details**.
34. Скопируйте **Public IPv4 address** вашего экземпляра.
35. Откройте новую вкладку в веб-браузере, вставьте IP-адрес, который вы только что скопировали, а затем **нажмите Enter**.

В настоящее время вы **не** можете получить доступ к веб-серверу, поскольку *группа безопасности* не разрешает входящий трафик на 80 порту, который используется для веб-запросов HTTP. Это демонстрация использования группы безопасности в качестве брандмауэра для ограничения сетевого трафика, который разрешен в экземпляре и из его.

Чтобы исправить это, теперь вы обновите группу безопасности, чтобы разрешить веб-трафик в порту 80.

36. Оставьте вкладку браузера открытой, но вернитесь к вкладке **EC2 Management Console**.
37. В навигационной панели слева выберите **Security Groups**.
38. Нажмите **Security group ID** для группы безопасности с именем **Web Server security group**.

Группа безопасности в настоящее время не имеет правил.

39. Нажмите **Edit inbound rules**. 40.Нажмите **Add rule** и укажите:

- **Type:** *HTTP*
- **Source:** *Anywhere*
- Нажмите **Save rules**

Новое правило Inbound HTTP создаст запись как для IP-адреса IPV4 (0.0.0.0/0), так и для IP-адреса IPV6 (::/0).

Примечание: использование "Anywhere", а также 0.0.0.0/0 или ::/0 не рекомендуется лучшими практиками для производственных нагрузок.

41. Вернитесь к вкладке веб-сервера, которую вы ранее открыли, и обновите страницу.

Вы должны увидеть приветственное сообщение с вашего веб-сервера!

Задание 4: Изменение типа экземпляра и размера хранилища

По мере изменения потребностей может оказаться, что ваш экземпляр чрезмерно используется (слишком мал) или недостаточно используется (слишком большой). Если это так, можно изменить *тип экземпляра.* Например, если *экземпляр t2.micro* слишком мал для рабочей нагрузки, его можно изменить на экземпляр *t2.small.* Аналогичным образом можно изменить размер диска.

Остановить экземпляр

Прежде чем можно изменить размера экземпляра, необходимо *остановить* его.

При остановке экземпляра он выключается. Плата за остановленный экземпляр EC2 не взимается, но плата за хранение прикрепленных томов Amazon EBS остается.

42. В консоли **EC2 Management Console**, в навигационной панели слева выберите **Instances**.

Web Server уже должен быть выбран. 43.Выберите **Instance state Stop instance**. 44.Нажмите **Stop**

Ваш экземпляр будет выполнять нормальное отключение, а затем перестанет работать.

45. Подождите пока **Instance State** изменится на: stopped

Изменение типа экземпляра

46. Выберите **Web Server**

47. В меню **Actions**, выберите **Instance settings Change instance type**, затем укажите:

- **Instance type:** *t2.small*
- Нажмите **Apply**

Когда экземпляр запущен снова, это будет *t2.small,* который имеет в два раза больше памяти, чем экземпляр *t2.micro.*

Изменение объема хранилища

48. В навигационной панели слева выберите **Volumes.** 49.В меню **Actions** выберите **Modify Volume**.

Объем диска в настоящее время имеет размер 8 GiB. Теперь вы увеличите размер этого диска.

50.Измените размер на: 10 51.Нажмите **Modify**

52.Нажмите **Yes** чтобы подтвердить и увеличить размер объема. 53.Нажмите **Close**

Запуск измененного экземпляра

Теперь вы начнете экземпляр снова, который теперь будет иметь больше памяти и больше дискового пространства.

54.В навигационной панели слева нажмите **Instances**. 55.Выберите **Instance state Start instance**.

Примечание: Изменение объема хранилища проходит через несколько последовательных состояний: Modifying, Optimizing, и Complete.

Задание 5: Исследование ограничений EC2

Amazon EC2 предоставляет различные ресурсы, которые можно использовать. Эти ресурсы включают изображения, экземпляры, тома и снимки. При создании учетной записи AWS существуют ограничения по умолчанию на эти ресурсы в зависимости от региона.

56.В навигационной панели слева выберите **Limits**. 57.Нажмите **All limits** и выберите **Running instances**.

Обратите внимание, что существует ограничение для типа экземпляра в зависимости от количества требуемых vCPU. Например, для **For running On- Demand All Standard,**у вас есть ограничение на vCPU.

Вы можете запросить увеличение для многих из этих ограничений.

Задание 6: Проверка защиты от остановки экземпляра

Вы можете удалить свой экземпляр, когда он вам больше не нужен. Это называется *прекращением* экземпляра. Вы не можете подключиться или перезапустить экземпляр после его завершения.

В этой задаче вы узнаете, как использовать защиту *от прерывания.*

58.В навигационной панели слева выберите **Instances**. 59.В меню **Instance state** выберите **Terminate instance**.

60. В диалоговом окне **Terminate instance** нажмите **Terminate**

Обратите внимание, появилось сообщение: *Failed to terminate instances ...*

Это гарантия предотвращения случайного прекращения экземпляра.

Если вы действительно хотите прекратить экземпляр, вам нужно будет отключить защиту от прерывания.

61. Закройте отображаемое сообщение об ошибке.

62.В меню **Actions** выберите **Instance settings Change termination protection**.

63. Отключите **Enable**

64. Нажмите **Save**

Теперь вы можете остановить экземпляр.

65. В меню **Instance state** выберите **Terminate instance**.

66. В диалоговом окне **Terminate instance** нажмите **Terminate**

Форма представления результата:

Продемонстрировать ход выполнения преподавателю. Показать свою учетную запись. Ответить на вопросы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 2.2 Создание группы автоматического масштабирования и балансировщика нагрузки.

Практическое занятие № 10 Работа с AutoScaling Group.

Цель:

Изучить назначение AutoScaling Group;

Создать шаблон запуска и группы автоматического масштабирования;

Проверить шаблон запуска и группы автоматического масштабирования.

Выполнив работу, Вы будете:

уметь:

создавать группу автоматического масштабирования с помощью шаблона запуска;

проверять, что группа запустила экземпляр EC2.

Материальное обеспечение:

MS Windows 7 (подписка Imagine Premium), MS Office 2007, 7 Zip, MS Visual Studio (подписка Imagine Premium), Visual Studio Code.

Задание:

Прежде чем создать группу автоматического масштабирования с помощью шаблона запуска, необходимо создать шаблон запуска, который включает параметры, необходимые для запуска экземпляра EC2, такие как идентификатор Amazon Machine Image (AMI) и тип экземпляра.

В этой задаче необходимо создать шаблон запуска, настроить группу автоматического масштабирования с помощью одного экземпляра и осуществить проверку.

Порядок выполнения работы:

1. Откройте консоль AWS
2. Авторизуйтесь
3. Создайте виртуальное частное облако MyVPC с частной подсетью
4. Создайте группу безопасности MySecurityGroup с разрешенным входящим и исходящим трафиком по протоколу TCP, порт 443, источник любой.

Ход работы:

Задание 1: Создание шаблона запуска

1. В меню **Services** выберите **EC2**.

В правом верхнем углу экрана, активируйте переключатель **New EC2 Experience** если он не активирован по умолчанию.

2. В навигационной панели слева, ниже Instances, нажмите Launch Templates.
3. Выберите **Create launch template** и настройте следующим образом:
4. В секции **Launch template name and description** укажите:

- **Launch template name:** Lab-template
- **Template version description:** version 1

Вам будет предложено выбрать **Amazon Machine Image (AMI)**, который является шаблоном для загрузочного раздела экземпляра и может содержать операционную систему, сервер приложений и приложения. Вы используете AMI для запуска **экземпляра,** который является копией AMI, запущенной в качестве виртуального сервера в облаке.

AMI доступны для различных версий Windows и Linux. В этом задании вы запустите экземпляр под управлением *Amazon Linux.*

5. Для **AMI**, выберите *Amazon Linux 2 AMI.*

Первый вариант, сразу под разделом **Quick Start**.

6. Для **Instance type**, выберите *t2.micro*.

При запуске экземпляра **тип экземпляра** определяет оборудование, выделенное экземпляру. Каждый тип экземпляров предлагает различные возможности вычислений, памяти и хранения данных и сгруппирован в **группы экземпляров** на основе этих возможностей.

7. Для **Security groups** выберите *MySecurityGroup*.
8. Пролистните страницу вниз и нажмите **Create launch template**
9. Нажмите **View launch templates**

Задание 2: Создание группы автоматического масштабирования

Группа автоматического масштабирования содержит набор экземпляров EC2, которые имеют схожие характеристики и рассматриваются как логическая группировка для целей масштабирования и управления экземплярами. Например, если одно приложение работает в нескольких экземплярах, может потребоваться увеличить количество экземпляров в этой группе для повышения производительности приложения или уменьшить количество экземпляров для снижения затрат при низком спросе. Вы можете использовать группу автоматического масштабирования для динамического изменения числа экземпляров на основе критериев, которые вы указываете, или поддерживать фиксированное количество экземпляров, даже если экземпляр становится неработоспособным. Это автоматическое масштабирование и поддержание количества экземпляров в группе автоматического масштабирования является основной функциональностью службы автоматического масштабирования Amazon EC2.

В этой задаче вы настроите группу автоматического масштабирования с помощью одного экземпляра.

10. В навигационном меню слева, ниже **Auto Scaling**, нажмите **Auto Scaling Groups**.
11. Нажмите **Create an Auto Scaling group** и укажите:

- **Auto Scaling group name:** Lab-Group
- **Launch template:** выберите созданный вами шаблон запуска.
- Нажмите **Next**

12. В секции **Network** укажите:

- **VPC:** MyVPC
- **Subnets:** выберите частную подсеть 15.Нажмите **Next**

16. На странице **Configure advanced options** укажите:

- **Health check grace period:** 60
- **Monitoring:** *Enable group metrics collection within CloudWatch*
- Нажмите **Next**

По умолчанию период проверки работоспособности экземпляра установлен на 300. Так как это лабораторная среда, вы установили его до 60, чтобы избежать необходимости ждать очень долго, чтобы увидеть автоматическое масштабирование после первой проверки работоспособности.

17. На странице **Configure group size and scaling policies** укажите:

- **Minimum capacity:** 1
- **Maximum capacity:** 2
- Нажмите **Next**

По умолчанию автоматическое масштабирование будет **поддерживать эту группу на начальном уровне.** Это означает, что один экземпляр всегда будет работать. Если экземпляр выходит из строя, он автоматически заменяется.

18. Нажмите **Next** чтобы перейти на страницу **Review**. 1
19. Нажмите **Create Auto Scaling group**

Задание 3: Проверка группы автоматического масштабирования

Теперь, когда вы создали группу автоматического масштабирования, вы можете

проверить, что группа запустила экземпляр EC2.

Выберите вашу Auto Scaling Group.

Перейдите к **Group Details** для просмотра информации о группе автоматического масштабирования.

20. Выберите вкладку **Activity**.

Столбец «Статус» содержит текущее состояние экземпляра. При запуске экземпляра столбец состояния показывает *PreInService.* Статус изменяется на *Successful* при успешном запуске экземпляра. Вы можете нажать кнопку обновления, чтобы увидеть изменение состояние вашего экземпляра.

21. Выберите вкладку **Instance management**.

Вы можете видеть, что ваша группа автоматического масштабирования запустила экземпляр EC2 и находится в состоянии *InService*. В столбце «Health Status» показан результат проверки состояния работоспособности экземпляра EC2.

22. Выберите вкладку **Monitoring**. Здесь вы можете увидеть информацию, связанную с мониторингом для вашей группы автоматического масштабирования.

Задание 4: Проверка автоматического масштабирования

Попробуйте следующий эксперимент, чтобы узнать больше об автоматическом масштабировании. Минимальный размер для группы автоматического масштабирования составляет 1 экземпляр. Поэтому, если вы завершаете работу экземпляра, Auto Scaling должен запустить новый экземпляр, чтобы заменить его.

23. На вкладке **Instance management** нажмите **Instance ID.** Он выглядит примерно так: **i-1234abcd1234**.

Вы перейдете на страницу Instances сервиса Amazon EC2.

Если вы получите ошибку "AWS Compute Optimizer: This user is not authorized to call AWS Compute Optimizer", то просто игнорируйте ее.

24. Выберите ваш экземпляр EC2.
25. В меню **Instance state** нажмите **Terminate instance**
26. В диалоговом окне **Terminate instance** нажмите **Terminate**

Состояние экземпляра изменится на *shutting-down.*

Дождитесь пока состояние экземпляра изменится на *terminated.*

Нажимайте кнопку обновления для проверки изменения состояния.

27. В навигационном меню слева **Auto Scaling Groups**.
28. Нажмите на вашу группу автоматического масштабирования. 30.Выберите вкладку **Instance management.**

Статус экземпляра будет отображаться как *Terminating.* Вскоре после этого вы увидите новый экземпляр, который появится со статусом *Pending* или *InService.*

31.Выберите вкладку **Activity.**

Все действия масштабирования регистрируются здесь. Просмотрите записи запуска и прекращения первого экземпляра, а затем запись для запуска нового экземпляра.

Форма представления результата:

Продемонстрировать ход выполнения преподавателю. Показать свою учетную запись. Ответить на вопросы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 2.2 Создание группы автоматического масштабирования и балансировщика нагрузки.
Практическое занятие № 11 Настройка Elastic Load Balancer.

Цель:

Изучить назначение Elastic Load Balancer;

Запустить два экземпляра Amazon Linux EC2;

Создать подключение к каждому веб-серверу;

Создать балансировщик нагрузки;

Просмотреть метрики CloudWatch балансировщика нагрузки.

Выполнив работу, Вы будете:

уметь:

создавать Elastic Load Balancer;

запускать два экземпляра Amazon Linux EC2 с веб- сервером Apache PHP и базовым приложением.

изучать метрики CloudWatch.

Материальное обеспечение:

MS Windows 7 (подписка Imagine Premium), MS Office 2007, 7 Zip, MS Visual Studio (подписка Imagine Premium), Visual Studio Code.

Задание:

В этой задаче необходимо запустить два экземпляра Amazon Linux EC2 с веб- сервером Apache PHP и базовым приложением, установленным во время инициализации.

Получить общедоступные записи DNS для обоих экземпляров EC2

Открыть в браузере каждую запись DNS, чтобы получить доступ к веб- серверу экземпляров

Создать простой балансировщик нагрузки для HTTP.

Использовать Amazon CloudWatch для мониторинга вашего ELB.

Порядок выполнения работы:

1. Откройте консоль AWS и авторизуйтесь
2. Создайте виртуальное частное облако MyVPC с публичной подсетью.

Ход работы:

Amazon EC2 предоставляет шаблоны, известные как *Amazon Machine Images*

(AMI), которые содержат конфигурацию программного обеспечения (например, операционная система, сервер приложений и приложение). Эти шаблоны используются для запуска *экземпляра,* который является копией AMI, запущенной в качестве виртуального сервера в облаке.

Вы можете запустить различные типы экземпляров из одного AMI. *Тип экземпляра* по существу определяет аппаратные возможности компьютера виртуального хоста для вашего экземпляра. Каждый тип экземпляра предлагает различные процессорные возможности и объем памяти. Выберите тип экземпляра в зависимости от объема памяти и вычислительной мощности приложения или программного обеспечения, которое вы планируете запустить на экземпляре. Можно запустить несколько экземпляров из AMI.

Ваш экземпляр продолжает работать до тех пор, пока вы не остановите или не удалите его, или пока он не выйдет из строя. Если экземпляр выходит из строя, можно запустить новый из AMI.

При создании экземпляра вам будет предложено выбрать тип экземпляра. Тип экземпляра, который вы выбираете, определяет, какая пропускная способность и вычислительные мощности доступны для вашего экземпляра.

1. В меню **Services** выберите **EC2**.
2. В навигационном меню слева нажмите **Instances**.
3. Нажмите **Launch Instance**
4. В строке **Amazon Linux 2 AMI**, нажмите **Select**

Тип экземпляра **t2.micro,** является самым дешевым вариантом, должен быть выбран автоматически.

5. Нажмите **Next: Configure Instance Details** и укажите:

- **Number of instances:** 2
- **Network:** *MyVPC*

6. Пролистните вниз и раскройте раздел **Advanced Details**.
7. Введите следующую **User data:**

```
#!/bin/sh

yum -y install httpd php chkconfig httpd on systemctl start httpd.service cd
/var/www/html
wget https://s3-us-west-2.amazonaws.com/us-west-2-aws-training/awsu- spl/spl-
03/scripts/examplefiles-elb.zip
unzip examplefiles-elb.zip
```

Этот скрипт позволит вам «оживить» ваш экземпляр. Он установит Apache и PHP и образец кода (PHP скрипты), необходимый для этого задания, когда экземпляр будет создан и запущен. Пользовательские данные предоставляют механизм передачи данных или скрипта службе метаданных Amazon, к которой экземпляры могут получить доступ во время запуска.

Совет Если вы вводите этот текст вместо того, чтобы копировать его, нажмите SHIFT-ENTER, чтобы создать новые строки в текстовом поле.

8. Нажмите **Next: Add Storage**

Используйте конфигурацию по умолчанию.

9. Нажмите **Next: Add Tags**
10. Нажмите **Add Tag** и укажите:

- **Key:** Name
- **Value:** MyLBInstances

Это имя, более известное как тег, появится в консоли при запуске экземпляра. Это упрощает отслеживание запусков машин в сложной среде. Используйте имя, которое вы можете легко распознать и запомнить.

11. Нажмите **Next: Configure Security Group** и укажите:
- **Security group name:** MyLBInstances
- **Description:** MyLBInstances

Группа безопасности выступает в качестве брандмауэра, который контролирует трафик, разрешенный в группу экземпляров. При запуске экземпляра Amazon EC2 его можно назначить одной или более группам безопасности. Для каждой группы безопасности в экземпляры группы добавляются правила, регулирующие разрешенной входящий трафик. Весь другой входящий трафик отбрасывается. Вы можете изменить правила для группы безопасности в любое время. Новые правила автоматически применяются для всех существующих и будущих экземпляров группы.

По умолчанию AWS создает правило, позволяющее безопасной оболочке (SSH) получить доступ с любого IP-адреса. Настоятельно *рекомендуется ограничить доступ* терминалов к диапазонам IP-адресов (например, IP-адреса, назначенные машинам внутри вашей компании), которые имеют необходимость в администрировании вашего экземпляра Amazon EC2.

12. Удалите правило для SSH.
13. Нажмите **Add Rule** и укажите:
- **Type:** *HTTP*
- **Source:** *Anywhere*
- Нажмите **Review and Launch**

Это добавит правило по умолчанию для HTTP, которое позволит обрабатывать запросы из любой точки Интернета, так как вы хотите, чтобы этот веб-сервер был доступен для широкой аудитории.

14. Проверьте настройки, и нажмите **Launch**

Вы можете увидеть предупреждение на этом экране, что " Your security group … is open to the world.». Это результат ограничения доступа SSH к вашей машине, как описано ранее. Для этого задания вы можете игнорировать предупреждение.

15. В окне **Select an existing key pair or create a new key pair**:
- Выберите **Proceed without a key pair**.
- Нажмите **I acknowledge that...**
- Нажмите **Launch Instances**

Пожалуйста, не запускайте свой экземпляр с парой ключей. Страница состояния уведомит вас о запуске экземпляров.

16. Нажмите **View Instances**
17. Дождитесь отображения:
- **Instance State:** *running*
- **Status Checks:** *2/2 checks passed*

Это означает, что ваш экземпляр теперь полностью доступен.

Это может занять несколько минут. Вы можете обновить статус экземпляров, нажав на значок обновления.

Задние 2: Подключение к каждому веб-серверу

В этой задаче вы:
- Получите общедоступные записи DNS для обоих экземпляров EC2
- Откроете в браузере каждую запись DNS, чтобы получить доступ к веб-

серверу экземпляров

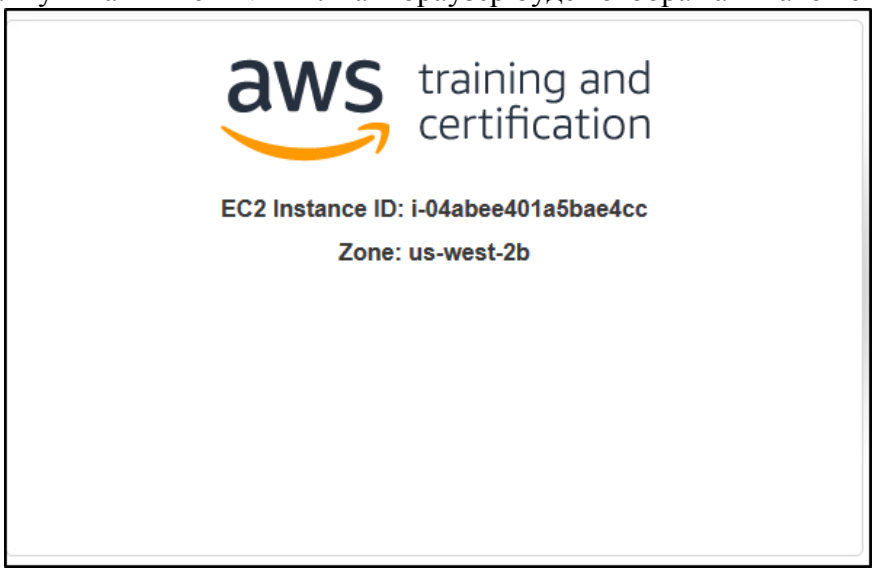
Все экземпляры Amazon EC2 получают два IP-адреса при загрузке: *частный* IP-адрес (RFC 1918) и *публичный IP-адрес,* которые непосредственно общаются друг с другом через Network Address Translation (NAT). Частные IP- адреса можно достичь только из сети Amazon EC2. Публичные адреса можно достичь из Интернета.

Amazon EC2 также предоставляет внутреннее *имя DNS и* публичное *имя DNS,* которые отражаются на частные и публичные IP-адреса, соответственно. Внутреннее имя DNS может быть доступно только в рамках Amazon EC2. Публичное имя DNS достижимо на общедоступном IP-адресе за пределами сети Amazon EC2 и на частном IP-адресе в сети Amazon EC2.

18. Выберите ваш первый экземпляр Amazon EC2. 21.Скопируйте ссылку **Public DNS (IPv4)** .

Она выглядит примерно так: *ec2-54-84-236-205.compute- 1.amazonaws.com.*

22. Откройте новое окно браузера, вставьте значение Public DNS в адресную планку и нажмите ENTER. Ваш браузер будет отображать такой экран:



Это веб-страница, возвращенная скриптом PHP, который был установлен при создании экземпляра. Это простой скрипт, который допрашивает службу метаданных на каждой машине и возвращает идентификатор экземпляра и название Зоны доступности, в которой работает экземпляр.

23. Повторите предыдущие два шага для второго экземпляра.

Обратите внимание, что каждая машина отображает разный идентификатор экземпляра. Это поможет вам определить, какой экземпляр обрабатывает ваш запрос, когда вы ставите балансер нагрузки перед ними.

Если при доступе к экземплярам из браузера вы видите ошибку вместо идентификатора экземпляра и зоны доступности, повторите попытку через пару минут. Вполне возможно, что скрипт загрузки все еще работает и еще не завершил установку и запуск веб-сервера и приложения PHP. Если ошибки сохраняются, убедитесь, что вы правильно ввели скрипт загрузки при запусках экземпляров и что группа безопасности разрешает доступ по порту 80.

Задание 3: Создание балансировщика нагрузки

Теперь у вас есть два веб-сервера. Теперь вам нужен балансировщик нагрузки перед этими серверами, чтобы дать пользователям единое местоположение для доступа к обоим и сбалансировать запросы пользователей

через них. В этой задаче вы создадите простой балансировщик нагрузки для HTTP.

24. В консоли **AWS Management Console**, в навигационном меню слева выберите **Load Balancers**.
25. Нажмите **Create Load Balancer**
26. Ниже **Application Load Balancer**, нажмите **Create** и укажите:

- **Name:** ELB
- **VPC:** *MyVPC*
- Выберите все зоны доступности
- Нажмите **Next: Configure Security Settings**

27. На странице **Step 2**, нажмите **Next:Configure Security Groups**
28. На странице **Step 3:**

- Снимите отметку **default**
- Выберите **MyLBInstances**
- Нажмите **Next: Configure Routing**

29. Для **Name** введите: myTarget
30. Раскройте **Advanced health check settings** и укажите:

- **Healthy threshold:** 3
- Нажмите **Next: Register Targets**

ELB будет периодически опрашивать каждый из экземпляров веб- службы для определения состояния работоспособности: код ответа 200 HTTP указывает на рабочее состояние, а любой другой код ответа указывает на ошибку. Если экземпляр неработоспособен и остается в этом состоянии следующее число проверок (*unhealthy threshold),*балансировщик нагрузки удаляет его из службы до тех пор, пока он не восстановится.

В этой конфигурации, что AWS проверяет корневой путь (/), который возвращает страницу по умолчанию, генерируемую PHP, которую вы видели ранее.

Healthy threshold – это количество успешных проверок, которые балансер нагрузки ожидает увидеть подряд, прежде чем ввести экземпляр в эксплуатацию за балансировщиком нагрузки. Более низкое значение ускоряет процесс.

31. Выберите оба экземпляра. 32.Нажмите **Add to registered** 33.Нажмите **Next: Review**
34. Проверьте конфигурацию и нажмите **Create**

Будет создан балансировщик нагрузки.

35. Нажмите **Close**

Это займет несколько минут, чтобы запустить балансировщик нагрузки, прикрепить веб-серверы, и пройти проверку работоспособности.

36. Подождите пока **State** изменится на *active*. 37.Скопируйте **DNS name .**
38. Откройте новую вкладку в браузере и:

- Вставьте скопированное DNS name
- Нажмите **Enter**

39. Обновите страницу несколько раз.

Вы заметите, что Amazon EC2 Instance IDs изменяется. Это означает, что повторные ответы возвращаются через два различных веб-сервера.

Балансировщики нагрузки могут охватывать зоны доступности, а также масштабироваться по мере необходимости для обработки спроса. Поэтому всегда следует получить доступ к балансировщику нагрузки по адресу DNS, а не по IP-адресу. Балансировщик нагрузки может иметь несколько IP-адресов, связанных с

его именем DNS.

40. В консоли **AWS Management Console**, в навигационном меню слева нажмите **Target Groups**

myTarget должна быть отмечена. 41.Перейдите на вкладку **Targets** .

Убедитесь, что оба экземпляра работоспособны.

Форма представления результата:

Продемонстрировать ход выполнения преподавателю. Показать свою учетную запись. Ответить на вопросы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 2.2 Создание группы автоматического масштабирования и балансировщика нагрузки.

Практическая работа № 12 Тестирование на устойчивость экземпляров EC2.

Цель:

Изучить назначение Elastic Load Balancer;

Запустить два экземпляра Amazon Linux EC2;

Создать подключение к каждому веб-серверу;

Создать балансировщик нагрузки;

Просмотреть метрики CloudWatch балансировщика нагрузки.

Выполнив работу, Вы будете:

уметь:

создавать Elastic Load Balancer;

запускать два экземпляра Amazon Linux EC2 с веб-сервером Apache PHP и базовым приложением.

изучать метрики CloudWatch. Осуществлять тестирование на устойчивость.

Материальное обеспечение:

MS Windows 7 (подписка Imagine Premium), MS Office 2007, 7 Zip, MS Visual Studio (подписка Imagine Premium), Visual Studio Code.

Задание:

В этой задаче необходимо запустить два экземпляра Amazon Linux EC2 с веб- сервером Apache PHP и базовым приложением, установленным во время инициализации.

Получить общедоступные записи DNS для обоих экземпляров EC2

Открыть в браузере каждую запись DNS, чтобы получить доступ к веб- серверу экземпляров

Создать простой балансировщик нагрузки для HTTP.

Использовать Amazon CloudWatch для мониторинга вашего ELB.

Порядок выполнения работы:

1.Откройте консоль AWS и авторизуйтесь

2.Создайте виртуальное частное облако MyVPC с публичной подсетью.

3. Работу продолжить на основе практики № 11.

Ход работы:

Задание 4: Просмотр метрик CloudWatch балансировщика нагрузки

В этой задаче вы будете использовать Amazon CloudWatch для мониторинга вашего ELB. Elastic Load Balancing автоматически сообщает о метриках балансировки загрузки в CloudWatch.

Amazon CloudWatch обеспечивает мониторинг облачных ресурсов AWS и приложений, которые работают на AWS. Разработчики и системные администраторы могут использовать его для сбора и отслеживания метрик, получения информации и немедленной реакции на бесперебойную работу своих приложений и предприятий. CloudWatch отслеживает ресурсы AWS, такие как Amazon EC2 и Amazon Relational Database Service (RDS) DB, а также может контролировать пользовательские метрики, генерируемые вашими приложениями и службами. С помощью CloudWatch вы получаете системную видимость использования ресурсов, производительности приложений и их работоспособности.

Amazon CloudWatch предоставляет надежное, масштабируемое и гибкое решение для мониторинга, которое можно начать использовать в течение нескольких минут. Вам больше не нужно делать настройку, управление или масштабирование собственных систем мониторинга и инфраструктуры. Используя CloudWatch, вы можете легко контролировать столько метрических данных, сколько вам нужно. CloudWatch позволяет программно извлекать данные мониторинга, просматривать графики и устанавливать сигналы тревоги, которые помогут вам устранению неполадок, точечным тенденциям и автоматизированным действиям в зависимости от состояния облачной среды.

42. В консоли **AWS Management Console**, в меню **Services** выберите **CloudWatch**.
43. В навигационном меню слева нажмите **Metrics**.
44. Нажмите **ApplicationELB**
45. Изучите метрики, которые выводит CloudWatch.

Показатели балансировки нагрузки включают задержку, количество запросов и здоровые и нездоровые подсчеты хоста. Метрики сообщаются по мере их встречи и могут занять несколько минут, чтобы появиться в CloudWatch.

Форма представления результата:

Продемонстрировать ход выполнения преподавателю. Показать свою учетную запись. Ответить на вопросы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 2.3 Создание ведра и стека.
Практическое занятие № 13 Хранение информации в S3.

Цель:

Создать ведро для хранения данных отчета EC2;
Загрузить объект в ведро;
Сделать объект общедоступным;
Осуществить версионирование.

Выполнив работу, Вы будете:

уметь:
загружать объекты в S3;
осуществлять хранение нескольких вариантов объекта в одном и том же ведре.

Материальное обеспечение:

MS Windows 7 (подписка Imagine Premium), MS Office 2007, 7 Zip, MS Visual Studio (подписка Imagine Premium), Visual Studio Code.

Задание:

- Amazon S3 является базовым сервисом для хранения информации.
- В этом задании вы создать ведро для хранения данных отчета EC2, а затем рассмотреть различные параметры конфигурации ведра.
- Загрузить объекты в свой reportbucket.
- Настроить разрешения на ведро и объект для проверки доступности.
- Получить доступ к объекту, чтобы подтвердить, что он является частным по умолчанию.
- Подключиться к экземпляру Amazon Elastic Compute Cloud (Amazon EC2) для проверки подключения и безопасности S3. Произвести версионирование.

Порядок выполнения работы:

1. Откройте консоль AWS и авторизуйтесь

Ход работы:

1. В AWS Management Console, в меню **Services** выберите S3.
2. Нажмите **Create bucket**

Имена ведра должны быть между 3 и 63 символами в длину и состоять только из букв нижнего регистра, чисел или дефисов. Название ведра должно быть глобально уникальным во всех Amazon S3, независимо от учетной записи или региона, и не может быть изменено после создания ведра. При вводе имени ведра отображается подсказка, показывающая любые нарушения правил именования.

3. В разделе General configuration назовите ведро: reportbucket(NUMBER)

Замените ***NUMBER*** в названии ведра случайным числом. Например:

reportbucket987987

* Оставьте в **Region** значение по умолчанию.

Выбор конкретного региона позволяет оптимизировать задержку, минимизировать затраты или выполнить нормативные требования. Объекты, хранящиеся в регионе, никогда не покидают этот регион, если вы не перенесите их в другой регион.

Прокрутите вниз и выберите **Create bucket**

7. Получите от инструктора файл new-report.png.
8. В S3 Management Console, найдите и выберите ведро, которое начинается с названия reportbucket.
9. Нажмите **Upload**

Это действие запускает мастер загрузки. Используйте этот мастер для загрузки файлов, выбрав их из диалога выбора файлов или перетащив их в окно S3.

10. Нажмите **Add files**
11. Выберите файл **new-report.png**. 12.Прокрутите вниз и нажмите **Upload**

Файл будет успешно загружен когда отобразится надпись **Upload succeeded** .

Если файл не отображается в ведре в течение нескольких секунд после его загрузки, возможно, потребуется выбрать кнопку обновления в правом верхнем направлении.

13. В секции **Upload: status** нажмите **Close**.
14. На странице reportbucket на вкладке с объектами найдите new-report.png и нажмите на имя файла new-report.png.

Открывается страница new-report.png. Обратите внимание, что навигация в верхнем левом обновлении со ссылкой, чтобы вернуться на страницу обзора ведра.

15. В разделе **Object overview** найдите и скопируйте ссылку **Object URL**. Ссылка должна выглядеть как: *https://reportbucket987987.s3-us-west-2.amazonaws.com/new-report.png*

16. Откройте новую вкладку браузера и вставьте ссылку object URL в поле адреса, а затем нажмите **Enter**.

Вы получаете ошибку **Access Denied.** Это связано с тем, что объекты Amazon S3 являются частными по умолчанию.

Теперь, когда вы убедились, что доступ в S3 по умолчанию является приватным, нужно сделать объект общедоступным.

17. Оставьте браузер с ошибкой Access Denied открытым и вернитесь к вкладке веб-браузера с **S3 Management Console**.

18. Вы все еще должны быть на вкладке **new-report.png** Object overview.

19. Нажмите кнопку **Object actions** и **Make public**, которая будет последней в списке.

Обратите внимание на предупреждение **Public access is blocked because Block Public Access settings are turned on for this bucket.** Эта ошибка отображается, потому что это ведро не имеет публичного доступа. Настройки ведра переопределяют любые разрешения, применяемые к отдельным объектам. Если вы хотите, чтобы объект был доступен для публичного просмотра, необходимо отключить Block Public Access (BPA).

20. Нажмите **Make public** прочитать предупреждение в верхней части окна о том, что "Failed to edit public access" т.к. выключено BPA.

21. Нажмите **Close** чтобы вернуться к обзору объекта.

22. Используйте навигацию в верхней части, чтобы вернуться к основной странице обзора reportbucket.

23. Перейдите на закладку **Permissions**.

24. Под **Block public access (bucket settings)**, нажмите **Edit** чтобы изменить настройки.

25. Отключите опцию **Block *all* public access** а затем отключите все остальные опции.

Обратите внимание, что все отдельные опции также остаются невыбранными. При выборе общего доступа необходимо выбрать отдельные параметры, которые применяются к вашей ситуации и целям безопасности. В рабочей среде рекомендуется использовать наименее разрешительные параметры.

26. Нажмите **Save changes**

27. В диалоговом окне в поле подтверждения введите *confirm* и нажмите **Confirm**

Сообщение **Successfully edited bucket settings for Block Public Access** отобразится сверху окна.

28. Перейдите на вкладку **Objects**.

29. Нажмите на имя файла **new-report.png**.

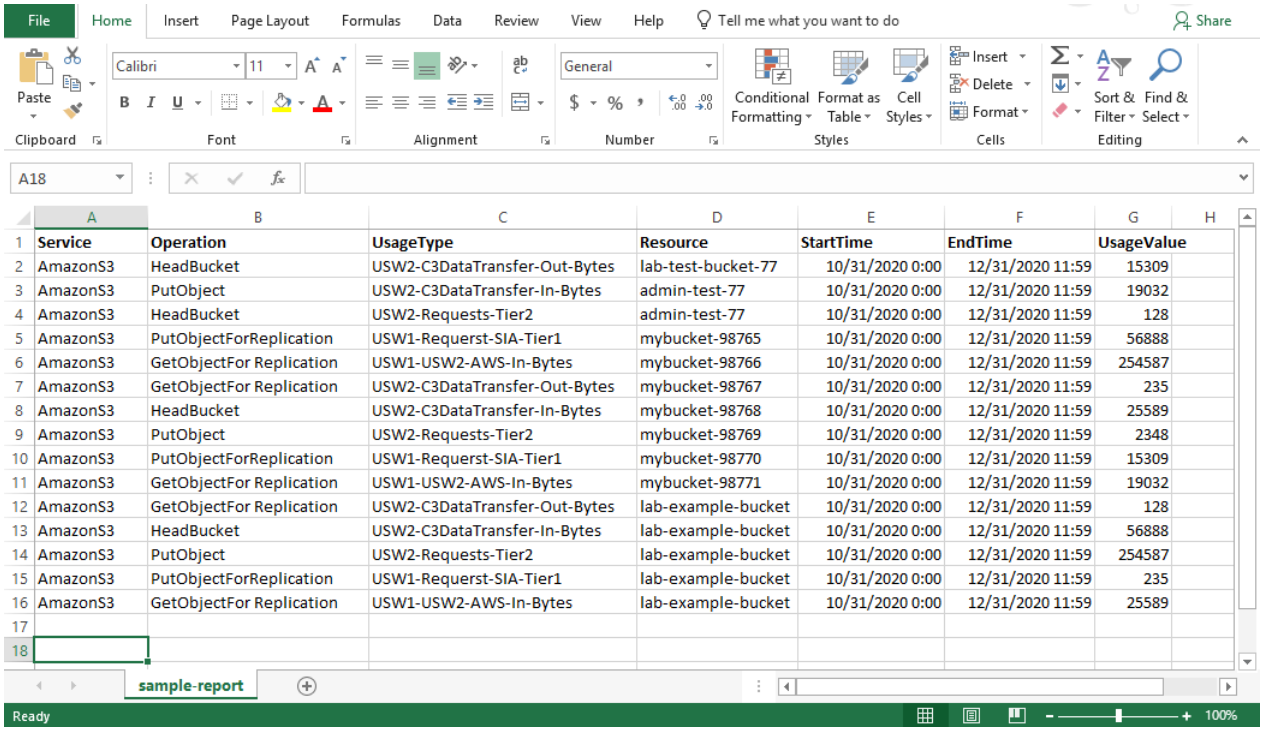
30. На странице new-report.png нажмите **Object actions** и выберите **Make public**.

Обратите внимание на предупреждение: **When public read access is enabled and not blocked by Block Public Access settings, anyone in the world can access the specified objects.** Сообщение напоминает, что если вы сделаете объект публичным, то каждый человек в мире сможет получить этот объект.

31. Нажмите **Make public** и вы увидите зеленое сообщение вверху окна

Successfully edited public access .

32. Нажмите **Close** чтобы вернуться к странице объекта.



	A	B	C	D	E	F	G
1	Service	Operation	UsageType	Resource	StartTime	EndTime	UsageValue
2	AmazonS3	HeadBucket	USW2-C3DataTransfer-Out-Bytes	lab-test-bucket-77	10/31/2020 0:00	12/31/2020 11:59	15309
3	AmazonS3	PutObject	USW2-C3DataTransfer-In-Bytes	admin-test-77	10/31/2020 0:00	12/31/2020 11:59	19032
4	AmazonS3	HeadBucket	USW2-Requests-Tier2	admin-test-77	10/31/2020 0:00	12/31/2020 11:59	128
5	AmazonS3	PutObjectForReplication	USW1-Requerst-SIA-Tier1	mybucket-98765	10/31/2020 0:00	12/31/2020 11:59	56888
6	AmazonS3	GetObjectFor Replication	USW1-USW2-AWS-In-Bytes	mybucket-98766	10/31/2020 0:00	12/31/2020 11:59	254587
7	AmazonS3	GetObjectFor Replication	USW2-C3DataTransfer-Out-Bytes	mybucket-98767	10/31/2020 0:00	12/31/2020 11:59	235
8	AmazonS3	HeadBucket	USW2-C3DataTransfer-In-Bytes	mybucket-98768	10/31/2020 0:00	12/31/2020 11:59	25589
9	AmazonS3	PutObject	USW2-Requests-Tier2	mybucket-98769	10/31/2020 0:00	12/31/2020 11:59	2348
10	AmazonS3	PutObjectForReplication	USW1-Requerst-SIA-Tier1	mybucket-98770	10/31/2020 0:00	12/31/2020 11:59	15309
11	AmazonS3	GetObjectFor Replication	USW1-USW2-AWS-In-Bytes	mybucket-98771	10/31/2020 0:00	12/31/2020 11:59	19032
12	AmazonS3	GetObjectFor Replication	USW2-C3DataTransfer-Out-Bytes	lab-example-bucket	10/31/2020 0:00	12/31/2020 11:59	128
13	AmazonS3	HeadBucket	USW2-C3DataTransfer-In-Bytes	lab-example-bucket	10/31/2020 0:00	12/31/2020 11:59	56888
14	AmazonS3	PutObject	USW2-Requests-Tier2	lab-example-bucket	10/31/2020 0:00	12/31/2020 11:59	254587
15	AmazonS3	PutObjectForReplication	USW1-Requerst-SIA-Tier1	lab-example-bucket	10/31/2020 0:00	12/31/2020 11:59	235
16	AmazonS3	GetObjectFor Replication	USW1-USW2-AWS-In-Bytes	lab-example-bucket	10/31/2020 0:00	12/31/2020 11:59	25589

33. Вернитесь к другой вкладке браузера, которая отображала **Access Denied** для new-report.png и обновите страницу.

new-report.png теперь отображается, потому что стал общедоступным.

34. Закройте вкладку веб-браузера, на которой отображается new-report.png и вернитесь на вкладку с Amazon S3 Management Console.

В этом примере предоставляется доступ только к одному конкретному объекту. Если вы хотите предоставить доступ ко всему ведру, вам нужно использовать политики ведра.

Дополнительные задания

Задание: Тестирование подключения из экземпляра EC2

В этой задаче вы подключаетесь к экземпляру Amazon Elastic Compute Cloud (Amazon EC2) для проверки подключения и безопасности S3.

35. В меню **Services** выберите **EC2**.

36. Создайте и запустите экземпляр с доступом по HTTPS, порт 443 и SSH порт 22

37. Выберите ваш экземпляр и нажмите **Connect**

38. В окне **Connect to instance**:

- Для **Connection method**, выберите **Session Manager**.

Session Manager позволяет подключиться к экземпляру без необходимости открытия определенных портов на брандмауэре или в группе безопасности Amazon Virtual Private Cloud (Amazon VPC).

39. Нажмите **Connect**

Откроется новая вкладка или окно браузера с подключением к экземпляру.

40. В окне терминала введите следующую команду для смены каталога и нажмите enter:

cd ~

41. Выполните следующую команду для перехода в домашний каталог: pwd

Результат должен быть примерно таким:

/home/ssm-user

Теперь вы находитесь в домашнем каталоге пользователя, где вы будете выполнять все остальные команды.

42. Введите следующую команду, чтобы посмотреть все доступные ведра S3.

aws s3 ls

Вывод должен быть примерно таким: 2020-11-11 22:34:46 reportbucket987987

Вы видите reportbucket который только что создали.

Примечание: для выполнения команд может понадобиться проверить Instance Profile и Role, которые отвечают за безопасность.

43. Введите следующую команду, чтобы перечислить все объекты в вашем ведре. Не забудьте изменить номер в конце имени reportbucket, чтобы соответствовать названию созданного ведра.

aws s3 ls s3://reportbucket(NUMBER)

Команда должна выглядеть примерно так: **aws s3 ls s3://reportbucket987987**

Результат должен быть примерно таким:

2020-11-11 15:46:34 86065 new-report.png

В настоящее время в вашем ведре есть только один объект, который называется new-report.png.

44. Выполните следующую команду чтобы создать новый файл. echo 'sample text' > test.txt

45. Выполните команду чтобы проверить, что файл создан. ls

Вывод будет примерно таким:

test.txt

46. Введите следующую команду чтобы скопировать файл в ведро. aws s3 cp test.txt s3://reportbucket(NUMBER)

Команда должна выглядеть примерно так: **aws s3 cp test.txt s3://reportbucket987987**

Вывод указывает на ошибку загрузки **upload failed**. Это связано с тем, что у нас есть только права на ведро и нет разрешений на проведение операции PutObject.

Версионирование является средством хранения нескольких вариантов объекта в одном и том же ведре. Вы можете использовать версии для сохранения, извлечения и восстановления каждой версии каждого объекта, хранящегося в вашем ведре Amazon S3. С помощью версий можно легко откатываться как от непреднамеренных действий пользователя, так и от сбоев приложения.

Для проверки и соответствия требованиям необходимо включить версионирование.

47. В меню ведра S3 переключитесь на вкладку **Properties**. 48.В разделе **Bucket Versioning** нажмите **Edit**

49. Выберите **Enable** и нажмите **Save changes**

Версионирование включается для всего ведра и всех объектов в ведре.

Оно не может быть включено для отдельных объектов.

Также следует понимать повышение затрат при предоставлении возможности версии.

50. У себя на компьютере измените файл new-report.png и сохраните его с тем же именем.

Хотя этот файл имеет то же имя, что и предыдущий файл, он содержит новый контент.

51. В консоли управления S3 выберите вкладку **Objects.**

В разделе **Objects** перейдите к **Show versions**.

52. Нажмите **Upload** и снова загрузите файл new-report.png.

53. Перейдите на вкладку браузера, которая отображает содержимое файла.

54.Обновите страницу.

Обратите внимание на изменения.

Amazon S3 всегда возвращает *последнюю версию* объекта, если версия не указана иным образом.

Вы также можете получить список доступных версий в консоли управления S3.

55. Закройте вкладку браузера с содержимым файла.

56. В S3 Management Console нажмите на new-report.png и перейдите на страницу объекта.

57. Выберите вкладку **Versions** , а затем выберите нижнюю версию, которая помечена *null* (Примечание: Это *не* последняя версия).

58. Выберите **Actions** и нажмите **Open**.

Теперь вы должны увидеть оригинальную версию файла.

Однако, если вы попытаетесь получить доступ к старой версии файла используя ссылку URL объекта, вы получите сообщение отказано в доступе. Это происходит, потому что политика ведра по умолчанию позволяет получить доступ только к последней версии объекта. Для доступа к предыдущей версии объекта необходимо обновить политику ведра, чтобы включить **разрешение "s3:GetObjectVersion".** Ниже приведен пример такой политики:

```
{
"Id": "Policy1557511288767", "Version": "2012-10-17",
"Statement": [
{
"Sid": "Stmt1557511286634",
"Action": [ "s3:GetObject", "s3:GetObjectVersion"
],
"Effect": "Allow",
"Resource": "arn:aws:s3:::mybucket45647467/*", "Principal": "*"
}
]
}
```

59. Для применения политики вернитесь в **S3 Management Console**.

60.Выберите ваше ведро, нажав на его название.

61. На странице ведра перейдите на вкладку **Permissions.**

62. В разделе **Bucket Policy** нажмите **Edit**

63. Скопируйте и вставьте текст политики в поле **Policy. Обратите внимание:** значение Resource должно содержать название вашего ведра.

64. Нажмите **Save Changes**

65. Вернитесь к вкладке **AWS Management Console** и выберите ссылку на имя ведра в левом верхнем части, чтобы вернуться к списку «Объекты ведра».

66. Выберите **Show versions** чтобы посмотреть версии объекта.

Теперь вы можете просмотреть доступные версии каждого объекта и определить, какая версия является последней.

Также обратите внимание, что теперь вы можете выбрать ссылку на название версии, чтобы перейти непосредственно к этой версии объекта в консоли.

67. Радом с **Show versions** переключите на стандартное представление объектов.

68. Выберите флажок напротив new-report.png 69.Нажмите **Delete**

70. Появится окно **Delete objects**.

71. Внизу окна в поле **Delete objects?** Необходимо написать *delete* для подтверждения удаления объекта. Нажмите **Delete objects**.

72. Нажмите **Close** чтобы вернуться к ведру.

Объект больше не отображается в ведре. Однако, если объект удален по ошибке, версия может быть использована для его восстановления.

73. Переключите **Show versions** для просмотра версий.

Обратите внимание, что объект отображается снова, но последняя версия — помечена **Delete marker.** Перечислены также две предыдущие версии. Если версионирование включено, объекты не удаляются немедленно. Вместо этого Amazon S3 вставляет маркер удаления, который становится текущей версией объекта. Предыдущие версии объекта не удаляются.

74. Выберите флажок слева от версии с **Delete marker**. 75.Для выбранного объекта нажмите **Delete**

76. Появится окно **Delete objects**.

77. Внизу секции **Permanently delete objects?** Нужно написать **permanently delete** и нажать **Delete objects**.

78. Нажмите **Close** чтобы вернуться к ведру.

79. Переключите **Show versions** в представление по умолчанию.

Обратите внимание, что объект был восстановлен в ведро. Удаление маркера удаления фактически восстановило объект в прежнем состоянии.

80. Чтобы удалить конкретную версию объекта переключите **Show versions** для просмотра версий.

Вы увидите две версии объекта.

81. Выберите флажок слева от последней версии объекта. 82.Для выбранного объекта нажмите **Delete**.

83. Появится окно **Delete objects**.

84. Внизу секции **Permanently delete objects?** Нужно написать **permanently delete** и нажать **Delete objects**.

85. Нажмите **Close** чтобы вернуться к ведру.

Обратите внимание, что в настоящее время существует только одна версия объекта. При удалении определенной версии объекта маркер удаления не создается. Объект удаляется навсегда.

86. Переключите **Show versions** чтобы вернуться к представлению по умолчанию.

87. Нажмите на имя объекта **new-report.png.** Откроется страница объекта. 88.Скопируйте ссылку **Object URL** в нижней части окна.

89.В новой вкладке браузера вставьте ссылку в поле адреса, а затем нажмите **Enter**.

Отобразится исходная версия объекта.

Форма представления результата:

Продемонстрировать ход выполнения преподавателю. Показать свою учетную запись. Ответить на вопросы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 2.3 Создание ведра и стека.
Практическое занятие № 14 Создание стека CloudFormation.

Цель:

Создать стек AWS CloudFormatio;
Изучить параметры CloudFormation;
Отследить создание стека CloudFormation;
Удалить стек.

Выполнив работу, Вы будете:

уметь:
Создавать стек CloudFormation со всеми необходимыми параметрами;
Осуществлять процесс отслеживания создания стека и удаления стека.

Материальное обеспечение:

MS Windows 7 (подписка Imagine Premium), MS Office 2007, 7 Zip, MS Visual Studio (подписка Imagine Premium), Visual Studio Code.

Задание:

Создать стек из заранее определенного шаблона AWS CloudFormation. Стек развернет экземпляр Amazon EC2 в VPC. Он также настроит его для запуска WordPress и MySQL. Осуществить мониторинг и удаление стека.

Порядок выполнения работы:

1. Откройте консоль AWS и авторизуйтесь.

Ход работы:

Создание стека CloudFormation

1. Получите у инструктора файл: wordpress.template

Шаблон содержит определение ресурсов, которые создадут блог WordPress.

2. В консоли AWS Management Console, в меню Services
3. Выберите CloudFormation.

Если вы видите уведомление в верхней части, которое сообщает, что переработанная консоль AWS CloudFormation доступна уже сейчас. Нажмите *Try it out now and provide us feedback* чтобы открыть обновленный интерфейс CloudFormation.

4. Нажмите **Create stack**

5. Нажмите **With new resources (standard)**
- Нажмите **Upload a template file**
- Нажмите **Choose file**
- Выберите файл *wordpress.template.*
- Нажмите **Next**

Изучите параметры CloudFormation

AWS CloudFormation имеет возможность указывать **параметры.** Значения, введенные в качестве параметров, передаются в стек и могут использоваться для настройки создания ресурсов, таких как настройка имени пользователя и пароля.

В шаблоне определены следующие параметры:

- **KeyName:** Имя пары ключей для доступа SSH
- **InstanceType:** Тип экземпляра EC2 для использования. В нем также указывается, какие типы экземпляров разрешены.
- **SSHLocation:** Диапазон IP-адресов, который может быть использован для SSH в экземплярах EC2
- *BName:** имя базы данных
- **DBUser:** Пользователь базы данных
- **DBPassword:** Пароль базы данных
- **DBRootPassword:** Root-пароль базы данных
- **VPCCIDR:** Блок CIDR для VPC, который будет создан
- **PublicSubnet1Param:** Значения, разрешенные в публичной подсети
- **PublicSubnet2Param:** Значения, разрешенные в публичной подсети

6. На странице **Create stack** укажите:
- **Stack name:** WordPress

7. Посмотрите на значение по умолчанию для **DBName**.

8. Откройте шаблон в текстовом редакторе и посмотрите на параметр **DBName**. DBName:
Default: wordpressdb
Description: The WordPress database name Type: String
MinLength: '1'
MaxLength: '64'
AllowedPattern: '[a-zA-Z][a-zA-Z0-9]*'
ConstraintDescription: must begin with a letter and contain only alphanumeric
characters.

Обратите внимание, что имя по умолчанию для **DBName** является *wordpressdb*. Поскольку шаблон использует свойство по умолчанию для **DBName,** *wordpressdb является текстом* по умолчанию, который появляется в разделе параметров при запуске шаблона. Кроме того, обратите внимание, что описание **DBName** отображается чуть выше поля ввода.

9. В консоли CloudFormation Management Console, посмотрите на значения по умолчанию для DBPassword, DBRootPassword, и DBUser.
Как видите, их нет.

10. В текстовом редакторе найдите те же параметры.
DBUser:
NoEcho: 'true'
Description: The WordPress database admin account username Type: String
MinLength: '1'
MaxLength: '16'
AllowedPattern: '[a-zA-Z][a-zA-Z0-9]*'

ConstraintDescription: must begin with a letter and contain only alphanumeric
characters. DBPassword:
NoEcho: 'true'
Description: The WordPress database admin account password Type: String
MinLength: '8'
MaxLength: '41' AllowedPattern: '[a-zA-Z0-9]*'
ConstraintDescription: must contain only alphanumeric characters. DBRootPassword:
NoEcho: 'true'
Description: MySQL root password Type: String
MinLength: '8'
MaxLength: '41' AllowedPattern: '[a-zA-Z0-9]*'
ConstraintDescription: must contain only alphanumeric characters.

Обратите внимание, что все три параметра не имеют свойства **по умолчанию**. Вот почему поле пусто. Тем не менее, все эти параметры имеют ограничения*(MinLength, MaxLength, AllowedPattern*). Эти ограничения также являются свойствами параметров.

11. В консоли **CloudFormation Management Console**, укажите:
- **DBPassword:** password1
- **DBRootPassword:** password2
- **DBUser:** admin

12. Посмотрите на параметр **InstanceType**.

Обратите внимание, **что t2.micro** выбирается по умолчанию. Однако, вы также можете выбрать *t1.micro* и *t2.nano*.

13. В текстовом редакторе найдите параметр **InstanceType.**

InstanceType:
Description: WebServer EC2 instance type Type: String
Default: t2.micro AllowedValues:
- t1.micro
- t2.nano
- t2.micro

ConstraintDescription: must be a valid EC2 instance type.

Обратите внимание, что параметр **InstanceType** имеет значение по умолчанию **t2.micro,** но он также позволяет выбрать *типы экземпляров t1.micro* и *t2.nano.*

14. В консоли **CloudFormation Management Console**, для **InstanceType** оставльте выбранным *t2.micro*.

15. Для **KeyName**, выберите доступную пару.

Если пара ключей отсутствует, можете создать ее используя Amazon EC2.

16. В текстовом редакторе найдите параметр **KeyName**

KeyName:
Description: Name of an existing EC2 KeyPair to enable SSH access to the instances
Type: AWS::EC2::KeyPair::KeyName
ConstraintDescription: must be the name of an existing EC2 KeyPair.
Обратите внимание, что он имеет тип *AWS::EC2::KeyPair::KeyName*.

При использовании типов параметров AWS любой, кто использует ваш шаблон для создания или обновления стека, должен указать существующие

значения AWS, которые находятся в их учетной записи и в регионе для текущего стека. Типы параметров AWS помогают гарантировать, что значения ввода для этих типов существуют и являются правильными до того, как AWS CloudFormation создаст или обновит любые ресурсы. Например, если вы используете тип параметра AWS::EC2::KeyPair::KeyName, AWS CloudFormation проверяет значение ввода в отношении существующих ключевых имен пар пользователей, прежде чем он создает какие-либо ресурсы, такие как экземпляры Amazon EC2.

17. В консоли **CloudFormation Management Console**, сравните остальные параметры с параметрами, определенными в шаблоне, а затем нажмите **Next**

Изучение параметров CloudFormation

Страница **Options** позволяет настраивать теги, роли и расширенные настройки.

- **Tags:** Теги являются произвольными парами значений ключей, которые могут быть использованы для идентификации вашего стека для таких целей, как распределение затрат.
- **Permissions:** Существующая роль службы управления идентификацией и доступом AWS (IAM), которую может взять на себя AWS CloudFormation.
- **Stack policy:** Определяет ресурсы, которые вы хотите защитить от непреднамеренных обновлений во время обновления стека. По умолчанию все ресурсы могут обновляться во время обновления стека.

18. Нажмите **Stack creation options**

- **Rollback on failure:** Указывается, следует ли откатить стек, если создание стека не удается. Как правило, следует принять значение по умолчанию Да. Выберите Нет, если вы хотите сохранить состояние стека, даже если создание не удается, например, при отладке шаблона стека.
- **Timeout:** указывается количество времени, в минутах, которое CloudFormation должен использовать для создания стеков. Если CloudFormation не может создать весь стек за отведенное время, выводится ошибка тайм-аута и откатывается стек. По умолчанию нет тайм-аута для создания стека. Однако отдельные ресурсы могут иметь свои собственные тайм-ауты в зависимости от характера службы, которую они реализуют. Например, если отдельный ресурс в стеке тайм- аут, создание стека также тайм-аут, даже если тайм-аут, указанный для создания стека, еще не достигнут.
- **Enable termination protection**: предотвращает случайное удаление стека. Если пользователь пытается удалить стек с включенной защитой прекращения, удаление не удается, и стек, включая его статус, остается неизменным.

Оставьте значения по умолчанию.

19. Нажмите **Next**
20. На странице **Review**:
- Проверьте настройки
- Нажмите **Create stack**

AWS CloudFormation создаст ресурсы по шаблону.

21. Выберите вкладку **Template**.

Вкладка **Template** отображает содержимое шаблона.

Если вы не видите вкладку **Template,** это может быть связано с шириной окна. Если это так, вы можете расширить окно так, чтобы шаблон появляется.

22. Перейдите на вкладку **Parameters**.

Вкладка **Parameters** отображает параметры и значения параметров, которые являются частью вашего шаблона.

23. Перейдите на вкладку **Events**.

Вкладка **Events** отображает каждое крупное событие в создании стека, отсортированного по времени каждого события, с последними событиями сверху. Вы можете увидеть различные события и их статус, такие как CREATE_IN_PROGRESS или CREATE_COMPLETE.

Исследование ресурсов CloudFormation

24. Перейдите на вкладку **Resources**

Вкладка **Resources** отображает ресурсы, которые являются частью стека. Первоначально вы можете увидеть только один или два ресурса. После создания стека вы увидите остальные ресурсы.

Для каждого ресурса отображается:

- **Logical ID:** Логический идентификатор должен быть алфавитным (A-Za-z0-9) и уникальным в шаблоне. Используйте логическое имя для ссылки на ресурс в других частях шаблона. Например, если вы хотите сопоставить объем Amazon Elastic Block Store с экземпляром Amazon EC2, вы ссылается на логические ID, чтобы связать магазины блоков с экземпляром.
- **Physical ID:** В дополнение к логическому идентификатору, некоторые ресурсы также имеют физический идентификатор, который является фактическим присвоенным именем для этого ресурса, например идентификатор экземпляра EC2 или имя ведра S3. Используйте физические списы для идентификации ресурсов за пределами шаблонов AWS CloudFormation, но только после создания ресурсов. Например, можно предоставить ресурсу экземпляра EC2 логический идентификатор MyEC2Instance; но когда AWS CloudFormation создает экземпляр, AWS CloudFormation автоматически генерирует и присваивает экземпляру физический идентификатор (например, i- 28f9ba55). Вы можете использовать этот физический идентификатор для идентификации экземпляра и просмотра его свойств (например, имя DNS) с помощью консоли Amazon EC2.
- **Type:** Тип ресурса определяет тип ресурса, который вы декларируете. Например, AWS::EC2::Instance объявляет экземпляр EC2.
- **Status:** Отображает код статуса ресурсов, которые создаются в стеке, таких как*: CREATE_IN_PROGRESS, CREATE_COMPLETE,* а также *CREATE_FAILED.*

Следующие ресурсы определяются в шаблоне, который вы развертываете.

- **WebServerSecurityGroup:** группа безопасности, которая позволяет получить доступ через порт 80 (HTTP) и порт 22 (SSH).
- **VPC:** Виртуальное частное облако, используемое для других ресурсов
- **InternetGateway:** шлюз для доступа в Интернет
- **AttachGateway:** способ прикрепить к интернет-шлюзу
- **PublicSubnet1: Ваша** первая публичная подсеть
- **PublicSubnet2: Ваша** вторая публичная подсеть
- **PublicRouteTable:** Таблица маршрутов для вашего VPC
- **PublicRoute:** общественный маршрут для таблицы маршрутов
- **PublicSubnet1RouteTableAssociation: способ связать свою первую подсеть с общедоступной таблицей маршрутов**
- **PublicSubnet2RouteTableAssociation:** способ связать вторую подсеть с общедоступной таблицей маршрутов
- **WebServer:** Amazon EC2 экземпляр, который создается как wordPress сервер

25. В текстовом редакторе найдите ресурс **WebServerSecurityGroup**.

WebServerSecurityGroup:

DependsOn: AttachGateway Type: AWS::EC2::SecurityGroup Properties:

GroupDescription: Enable HTTP access via port 80 locked down to the load balancer

+ SSH access VpcId: !Ref 'VPC'

SecurityGroupIngress:

- IpProtocol: tcp FromPort: '80'

ToPort: '80' CidrIp: 0.0.0.0/0

- IpProtocol: tcp FromPort: '22'

ToPort: '22'

CidrIp: !Ref 'SSHLocation'

Этот код определяет группу безопасности, которая разрешает трафик HTTP (порт 80) из Интернета (0.0.0.0/0) и SSH соединения (порт 22) из диапазона IP, который был указан как **SSHLocation** в разделе параметров.

26. В текстовом редакторе найдите ресурс **WebServer.**

WebServer:

Type: AWS::EC2::Instance DependsOn: WebServerSecurityGroup Metadata:

AWS::CloudFormation::Init:

install_wordpress:

packages:

yum:

php: []

php-mysql: []

mysql: []

mysql-server: [] mysql-devel: [] mysql-libs: []

httpd: []

Этот код определяет экземпляр Amazon EC2 с шагами для установки WordPress. Это включает в себя установку языка сценариев PHP, базы данных MySQL и веб-сервера HTTP. Представлен фрагмент кода.

27. Найдите раздел **Properties** вашего ресурса **WebServer.**

Properties:

ImageId: !FindInMap

- AWSRegionArch2AMI
- !Ref 'AWS::Region'
- !FindInMap

-AWSInstanceType2Arch

-!Ref 'InstanceType'

-Arch

Обратите внимание, что есть свойство **ImageId**. Шаблон создает экземпляр Amazon EC2 из предопределенного образа.

Свойство **ImageId** предоставляет уникальный идентификатор образа машины Amazon (AMI), который назначается во время регистрации. Обратите внимание, что **ImageId** расположен с помощью функции (! FindInMap). Функция ищет ваш **ImageId** с помощью двух карт в шаблоне **AWSRegionArch2AMI** и **AWSInstanceType2Arch**. Эти карты находятся в разделе **Mappings** вашего шаблона.

Изучение CloudFormation Mappings

28. В текстовом редакторе найдите раздел **Mappings**.

Дополнительный раздел Mappings соответствует ключу к соответствующему набору именованых значений. Например, если вы хотите установить значения на основе региона, можно создать отображение, которое использует имя региона в качестве ключа и содержит значения, которые вы хотите указать для каждого конкретного региона. Для получения значений на карте используется внутренняя функция Fn::FindInMap.

29. Найдите отображение **AWSInstanceType2Arch**.

Mappings: AWSInstanceType2Arch: t1.micro:
Arch: PV64 t2.nano:
Arch: HVM64 t2.micro:
Arch: HVM64

30. Какое значение, если используемый тип экземпляра **t2.micro**?

Это должен быть **HVM64**.

31. Найдите отображение **AWSRegionArch2AMI**. AWSRegionArch2AMI:
us-east-1:
PV64: ami-2a69aa47 HVM64: ami-6869aa05
HVMG2: ami-2e5e9c43 us-west-2:
PV64: ami-7f77b31f HVM64: ami-7172b611 HVMG2: ami-83b770e3

32. Какое значение для **us-west-2**?

Это должно быть **ami-7172b611**.

Изучение выходных параметров CloudFormation

33. В текстовом редакторе найдите раздел **Outputs.**

Outputs:
WebsiteURL:
Value: !Sub 'http://${WebServer.PublicDnsName}/wordpress' Description: WordPress Website
EC2IPAddress:
Value: !GetAtt WebServer.PublicIp

Дополнительный раздел **Outputs** объявляет значения вывода, которые можно импортировать в другие стеки (для создания перекрестных ссылок), возврат в ответ (для описания вызовов стеков) или просмотр на консоли AWS CloudFormation.

Код **Outputs** выше предоставляет ссылку на веб-сайт WordPress. Он также предоставляет публичный IP-адрес используемого экземпляра EC2.

34. В консоли CloudFormation Management Console, выберите вкладку Outputs.

Когда стек был развернут, он может отображать информацию о выходе.

Этот стек предоставляет ссылку на веб-сайт WordPress.

35. Скопируйте значение WebsiteURL и EC2IPAddress в текстовый редактор.

Подключитесь к сайту WordPress с помощью веб-браузера

36. Откройте новую вкладку веб-браузера и введите значение WebsiteURL, который соединяет вас с вашим новым веб-сайтом WordPress.

Вы должны быть направлены на страницу конфигурации WordPress.

37. Укажите следующие параметры:

- **Site Title:** Введите любое название, которое вы хотите
- **Username:** student
- **Password:** student123
- **Confirm Password:** Подтвердите использование слабого пароля d
- **Your Email:** student@example.com

- **Search Engine Visibility:** не меняйте 39.Нажмите **Install WordPress**

Вы должны увидеть надпись *Success*

40.Нажмите **Log In** и укажите учетные данные:

- **Username:** student
- **Password:** student123

Вы будете направлены на WordPress dashboard.

Теперь вы можете создать запись в блоге, чтобы добавить информацию на свой сайт.

Подключитесь к экземпляру EC2 через SSH и изучите изменения на экземпляре.

59. В консоли AWS Management console в меню Services выберите CloudFormation.
60. Выберите **WordPress**.
61. 61.Нажмите кнопку **Delete**.
62. 62.Нажмите **Delete stack**

Статус стека изменяется на DELETE_IN_PROGRESS. Таким же образом, как вы отслеживали создание стека, вы можете контролировать его удаление с помощью вкладки **Events .** Когда AWS CloudFormation завершает удаление стека, он удаляет стек из списка.

Форма представления результата:

Продемонстрировать ход выполнения преподавателю. Показать свою учетную запись. Ответить на вопросы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 2.4 Создание экземпляра RDS и DynamoDB. Кластер Redis.

Практическое занятие № 15 Создание базы данных Amazon RDS для MySQL и осуществление доступа к ней.

Цель:

Создать базу данных Amazon RDS для MySQL;
Осуществить доступ к базе данных;

Выполнив работу, Вы будете:
уметь:
создавать базу данных на сервисе;
подключаться к экземпляру EC2;
подключаться к базе данных RDS с помощью клиента mysql.

Материальное обеспечение:
MS Windows 7 (подписка Imagine Premium), MS Office 2007, 7 Zip, MS Visual Studio (подписка Imagine Premium), Visual Studio Code.

Задание:
Создать базу данных Amazon RDS для MySQL.

Порядок выполнения работы:
1. Откройте консоль AWS и авторизуйтесь

Ход работы:

1. В консоли **AWS Management Console**, в меню **Services** выберите **RDS**.
2. В навигационном меню слева выберите **Databases**.

Если пункт **Databases** не виден, нажмите значок навигации слева, а затем нажмите **Databases**.

3. Нажмите **Create database** и укажите:
- **Engine type:** *MySQL*
- **Templates:** *Dev/Test*

4. В секции **Settings** укажите:
- **DB instance identifier:** my-rds
- **Master username:** student
- **Master password:** Pass.123
- **Confirm password:** Pass.123

5. В секции **DB instance size** укажите:
- *Burstable classes*
- *db.t2.micro*

Если тип экземпляра *db.t2.micro* не отображается, включите кнопку переключения *Include previous generation classes* .

6. В секции **Availability & durability** для *Multi-AZ deployment*, отметьте **Do not create a standby instance**.
7. В секции **Connectivity** укажите:
- **Virtual Private Cloud (VPC):** *Create VPC*
- **Public accessible:** *No*
- **VPC security groups:** *Create new*

8. Раскройте секцию **Additional configuration**, и укажите:
- **Initial database name:** lab
- Снимите отметку **Enable automatic backups**
- Снимите отметку **Enable Enhanced monitoring**
- Снимите отметку **Enable auto minor version upgrade**

9. Прокрутите вниз страницы и нажмите **Create database**

На этой странице отображаются сведения для создаваемого экземпляра RDS. Создание экземпляра RDS займет около 10 минут.

Во время создания RDS был создан экземпляр Amazon EC2 Linux. Подключитесь к экземпляру EC2 используя SSH.

10. Скачайте **PEM-**сертификат для экземпляра EC2. 13.Поместите его в ваш рабочий каталог.

14. Скопируйте эту команду в текстовый редактор:
ssh -i KEYPAIR.pem ec2-user@EC2PublicIP

15. Замените *KEYPAIR.pem* на путь к PEM-файлу, который вы скачали.

16.Замените *EC2PublicIP* значением EC2PublicIP экземпляра к которому вы подключаетесь.

17.Скопируйте команду, вставьте ее в окно Терминала и выполните. 18.При запросе разрешить первое подключение к этому удаленному серверу SSH введите: *yes*.

Поскольку вы используете ключ-пару для проверки подлинности, вам не будет предложено ввести пароль.

Теперь вы будете подключаться к базе данных RDS с помощью клиента **mysql,** установленного на экземпляре EC2.

Для начала дождитесь возможности для создания нового соединения.

19. Вернитесь в **AWS Management Console**.

20. В навигационном меню слева выберите **Databases**.

21. Дождитесь пока для экземпляра **my-rds** статус изменится на **Available**.

Вы можете нажимать обновить каждые 60 секунд, чтобы обновить консоль.

22. Нажмите на экземпляр **my-rds**.

23. В разделе **Connectivity & security** скопируйте значение **Endpoint** в текстовый редактор.

Значение выглядит примерно так:
my-rds.cmq1uckiyvci.us-west-2.rds.amazonaws.com

24. В терминале, подключенном к консоли экземпляра EC2:

- Скопируйте и вставьте следующую команду:

mysql --user student --password --host ENDPOINT

- Замените **ENDPOINT** значением, которое вы скопировали в текстовый редактор
- Нажмите **Enter**
- При запросе пароля введите: Pass.123

Теперь вы вошли в консоль MySQL. Вы должны увидеть **mysql>**.

25. Скопируйте следующую команду, вставьте ее в терминал и выполните:
CREATE TABLE lab.staff (firstname text, lastname text, phone text);
INSERT INTO lab.staff VALUES ("John", "Smith", "555-1234");
INSERT INTO lab.staff VALUES ("Sarah", "Jones", "555-8866");

Эти команды создают новую таблицу и вставляют некоторые данные в базу данных.

Теперь вы можете выполнить запрос к базе данных.

26. Скопируйте следующую команду, вставьте ее в терминал и выполните:
SELECT * FROM lab.staff WHERE firstname = "Sarah";

Будут отображаться данные, относящиеся к Sarah.

Не стесняйтесь экспериментировать с другими запросами SQL.

Форма представления результата:

Продемонстрировать ход выполнения преподавателю. Показать свою учетную запись. Ответить на вопросы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 2.4 Создание экземпляра RDS и DynamoDB. Кластер Redis.
Практическое занятие № 16 Осуществление работы в DynamoDB, создание новых таблиц.

Цель:
Познакомиться с DynamoDB;
Создать новые таблицы и запрос к таблице;

Выполнив работу, Вы будете:
уметь:
создавать таблицы в DynamoDB;
изменять элементы в таблице;
осуществить запрос к таблице.

Материальное обеспечение:
MS Windows 7 (подписка Imagine Premium), MS Office 2007, 7 Zip, MS Visual Studio (подписка Imagine Premium), Visual Studio Code.

Задание:
В этом задании вы создатьновую таблицу в DynamoDB под названием Music. Каждая таблица требует первичный ключ, который используется для разделения данных на серверах DynamoDB. Таблица также может иметь ключ сортировки. Комбинация первичного ключа и ключа сортировки однозначно определяет каждый элемент в таблице DynamoDB.

Добавить данные в таблицу Music . Таблица представляет собой набор данных по конкретной теме.

Осуществить запрос таблицы DynamoDB. Удалить таблицу.

Порядок выполнения работы:
1. Откройте консоль AWS и авторизуйтесь

Ход работы:
1. В консоли AWS Management Console, нажмите **Services**, и выберите **DynamoDB**.

Примечание: включите переключатель **Switch to the new console** в верхней части страницы, если он выключен.

2. Нажмите **Create table**.
3. Для **Table name**, введите: *Music*
4. Для **Primary key**, введите: *Artist*, и оставьте выбранным тип **String**.
5. Нажмите **Add sort key**, в новом поле укажите: *Song*, и оставьте выбранным тип **String**.

В таблице будут использоваться настройки по умолчанию для индексов и резервирования емкости.

6. Нажмите **Create**.

Таблица будет создана менее чем за минуту.

Каждая таблица содержит несколько **item (элементов).** Элемент — это группа атрибутов, которые однозначно идентифицируются среди всех остальных элементов. Элементы DynamoDB во многом похожи на строки в других системах баз данных. В DynamoDB количество элементов, которые можно хранить в таблице, не ограничивается.

Каждый элемент состоит из одного или нескольких **attribute (атрибутов).** Атрибут является атомарным элементом данных, и не может быть декомпозирован. Например, элемент в таблице *Music* содержит такие атрибуты, как Песня и Исполнитель. Атрибуты в DynamoDB похожи на столбцы в других системах баз данных, но каждый элемент (ряд) может иметь различные атрибуты (столбцы).

При записи элемента в таблицу DynamoDB требуется только первичный ключ и ключ сортировки (если он используется). Помимо этих полей, таблица не требует схемы. Это означает, что можно добавить атрибуты в один элемент, который может отличаться от атрибутов на других элементах.

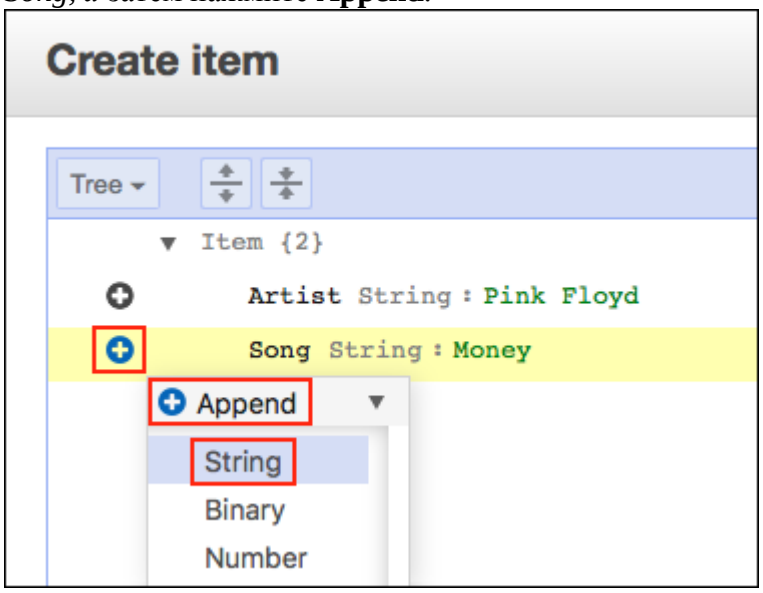
7. Выберите вкладку **Items** и нажмите **Create item.**

Если таблица все еще создается, обновите веб-страницу для отображения вкладки *Items* .

8. В строке **Artist** введите: Pink Floyd 11.В строке **Song** введите: Money

Это единственные необходимые атрибуты, но теперь вы можете добавить дополнительные атрибуты.

12. Чтобы создать дополнительный атрибут, нажмите знак плюс слева от *Song, a* затем нажмите **Append.**



13. В выпадающем меню выберите **String.**

Будет добавлена новая строка атрибутов.

14. Для нового атрибута укажите:

- В **FIELD** укажите: Album
- В **VALUE** укажите: The Dark Side of the Moon

15. Добавьте еще один новый атрибут, нажав знак плюс слева от *Album, a* затем нажмите **Append.**

16. В выпадающем меню выберите **Number.**

Будет добавлен новый атрибут с типом *number.*

17. Для нового атрибута:

- В **FIELD** укажите: Year
- В **VALUE** укажите: 1973

18. Нажмите **Save** для сохранения элемента с четырьмя атрибутами.

Элемент появится в консоли.

19. Теперь создайте второй **элемент,** используя следующие атрибуты:

Attribute Name	**Attribute Type**	**Attribute Value**
Artist	String	John Lennon
Song	String	Imagine
Album	String	Imagine
Year	Number	1971
Genre	String	Soft rock

Обратите внимание, что этот элемент имеет дополнительный атрибут, называемый *genre* (жанр). Это пример того, что каждый элемент способен иметь различные атрибуты без предварительного определения схемы таблицы.

20. Создайте третий **элемент**, используя эти атрибуты:

Attribute Name	**Attribute Type**	**Attribute Value**
Artist	String	Psy
Song	String	Gangnam Style
Album	String	Psy 6 (Six Rules), Part 1
Year	Number	2011
LengthSeconds	Number	219

Этот элемент имеет новый атрибут *LengthSeconds,* определяющий длину песни. Это свидетельствует о гибкости базы данных NoSQL.

Существуют также более быстрые способы загрузки данных в DynamoDB, такие как использование AWS Data Pipeline, программная загрузка данных или использование одного из бесплатных инструментов, доступных в Интернете.

Например, вы заметили, что в ваших данных есть ошибка. В этой задаче вы измените существующий элемент.

21. Выберите **Psy.**

22. Измените **Year** с *2011* на *2012*. 23.Нажмите **Save.**

Элемент будет обновлен.

Есть два способа запроса таблицы DynamoDB: *Query* и *Scan.*

Query находит элементы на основе первичного ключа и, при необходимости, ключа сортировки. Элементы индексируются по этим ключам, поэтому операция выполняется очень быстро.

24. В выпадающем списке, где указано **Scan** (расположен ниже кнопки *Create item*) выберите **Query**.

Будут отображены поля Partition Key (первичный ключ) и Sort Key (ключ сортировки).

25. Введите эти значения:

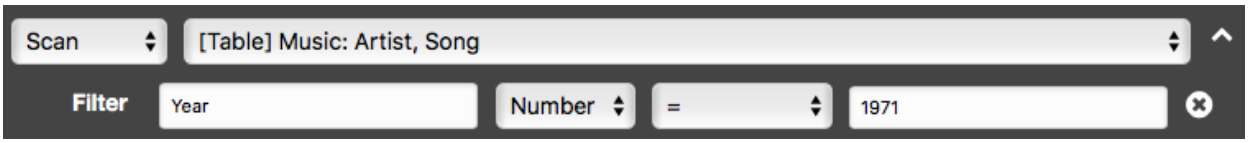
- **Partition key:** Psy

- **Sort key:** Gangnam Style 26.Нажмите **Start search**.

Песня быстро появляется в списке. *Query* является наиболее эффективным способом получения данных из таблицы DynamoDB.

Второй вариант – использование *scan* для выборки элементов. Этот механизм предполагает просмотр каждого элемента в таблице, поэтому менее эффективен и может занять значительное время для больших таблиц.

27.В выпадающем списке, где указано **Query** выберите **Scan**. 28.Нажмите **Add filter**, затем:



- Для **Enter attribute** укажите: Year
- Измените *String* на *Number*
- Для **Enter value** укажите: 1971
- Нажмите **Start search**

Отобразится только песня, выпущенная в 1971 году.

В этой задаче вы удалите таблицу *Music,* при этом удалятся все данные в таблице.

29.Нажмите **Delete table.** В поле подтверждения введите *Delete* и нажмите **Delete**

Таблица будет удалена.

Форма представления результата:

Продемонстрировать ход выполнения преподавателю. Показать свою учетную запись. Ответить на вопросы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 2.4 Создание экземпляра RDS и DynamoDB. Кластер Redis. Практическое задание № 17 Создание VPC по умолчанию.

Цель:
Создать VPC по умолчанию;
Создать пары ключей;
Создать и запустить экземпляр.

Выполнив работу, Вы будете:
уметь:
виртуальное частное облако с тремя подсетями;
создавать ключи;

Материальное обеспечение:
MS Windows 7 (подписка Imagine Premium), MS Office 2007, 7 Zip, MS Visual Studio (подписка Imagine Premium), Visual Studio Code.

Задание:
Создать виртуальное частное облако по умолчанию, создать ключи. Осуществить запуск экземпляра.

Порядок выполнения работы:
1. Откройте консоль AWS и авторизуйтесь

Ход работы:
1. В консоли **AWS Management Console** нажмите **Services** и выберите **VPC**
2. В навигационном меню слева выберите **Your VPCs**
3. В правом верхнем углу в меню Actions выберите **Create default VPC**

VPC по умолчанию позволяет запускать ресурсы Amazon EC2 без необходимости

создавать и настраивать собственные VPC и подсети. VPC по умолчанию имеет подсеть в каждой зоне доступности, интернет-шлюз и таблицу маршрутизации с маршрутом к интернет-шлюзу.

4. Нажмите **Create default VPC**

Будет создано виртуальное частное облако с тремя подсетями, таблицей маршрутизации, интернет-шлюзом, группой безопасности и списком контроля доступа.

- В консоли **AWS Management Console** нажмите **Services** и выберите **EC2**

- В навигационном меню слева выберите **Key Pairs**.
- Нажмите **Create key pair**.
- В поле **Name** введите: *my-key-pair*.
- В разделе **Private key file format** выберите **pem**.
- Нажмите **Create key pair**.

Файл ключа автоматически загружается вашим браузером. Имя файла — это имя, указанное в качестве имени вашей ключевой пары, а расширение имени файла определяется форматом файла, который вы выбрали. Сохраните частный файл ключа в вашем рабочем каталоге.

1. В навигационном меню слева нажмите **Instances**.
2. Нажмите **Launch Instances**.
3. На странице **Step 1: Choose an Amazon Machine Image (AMI)**, найдите Amazon Linux 2 AMI вверху списку и выберите **Select**.
4. На странице **Step 2: Choose an Instance Type**, выберите t2.micro и нажмите **Next: Configure Instance Details**.
5. На странице **Step 3: Configure Instance Details** укажите:
   - **Number of instances**: 1.
   - Остальные параметры оставьте без изменений
6. Нажмите **Next: Add Storage**.
7. Нажмите **Next: Add Tags**.
8. Добавьте тег Name для именования экземпляра и нажмите **Next: Configure Security Group**.
9. На странице **Step 6: Configure Security Group**, в разделе **Assign a security group** можно выбрать **Select an existing security group** и указать группу безопасности по умолчанию либо выбрать **Create a new security group** и создать новую группу безопасности.

Группа безопасности по умолчанию разрешает любой трафик. Однако для SSH требуется отдельное правило. При добавлении правила используйте следующие параметры:

- **Type:** SSH
- **Protocol:** TCP
- **Port Range:** 22
- **Source:** Anywhere 0.0.0.0/0 10.Нажмите **Review and Launch**. 11.Нажмите **Launch**.

1. Выберите созданный экземпляр и нажмите **Connect**.
2. Используйте SSH для подключения к терминалу экземпляра.
3. В терминале EC2 последовательно выполните следующие команды и изучите их назначение и результат:

sudo yum install git -y sudo yum install mysql -y sudo yum install python3 -y

pip3 install --user virtualenv

git clone https://github.com/aws-samples/amazon-elasticache-samples/ cd amazon-elasticache-samples/database-caching

virtualenv venv
source ./venv/bin/activate
pip3 install -r requirements.txt
Теперь экземпляр EC2 готов к работе.

Форма представления результата:

Продемонстрировать ход выполнения преподавателю. Показать свою учетную запись. Ответить на вопросы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Тема 2.4 Создание экземпляра RDS и DynamoDB. Кластер Redis.
Практическая работа № 18 Работа кластера Redis. Очистка мусора.

Цель:
Создать кластер Redis;
Провести очищение кластера;

Выполнив работу, Вы будете:
уметь:
создавать Redis;
создавать и заполнять базы данных MySQL;
проводить организацию кеширования и удаления.

Материальное обеспечение:
MS Windows 7 (подписка Imagine Premium), MS Office 2007, 7 Zip, MS Visual Studio (подписка Imagine Premium), Visual Studio Code.

Задание:
Проинициировать создание кластера Redis. Выполнить создание и заполнение базы данных несколькими способами. Осуществить стратегию кэширования и удалить кластер.

Порядок выполнения работы:
1. Откройте консоль AWS и авторизуйтесь

Ход работы:

1. В консоли AWS Management Console нажмите Services и выберите ElasticCache.
2. Нажмите **Get Started Now**.
3. Для **Cluster engine** выберите *Redis*

Проверьте что опция Cluster Mode enabled отключена.

4. В поле **Name** укажите название кластера Redis Cluster: *elc-lab.*
5. Измените **Node type** на *cache.t2.micro.*
6. Укажите **Number of replicas**: 1.

Это реплики только для чтения, они позволяют масштабировать операции чтения. В случае сбоя реплика может взять на себя роль основного узла.

7. Выберите опцию **Multi-AZ**.

В маловероятном случае сбоя первичного узла или зоны доступности или даже в случае запланированного обслуживания ElastiCache для Redis может заменить неудачный экземпляр, и реплика берет на себя роль основного узла. В результате время простоя сводится к минимуму.

8. Для **Subnet group** выберите VPC по умолчанию.
9. Для **Availability zones placement** укажите *No preference.*

В этом задании мы не будем использовать шифрование, но имеется возможность настроить шифрование как для хранимых данных, так и для передаваемых данных.

10. Выберите группу безопасности для кластера Redis.

Важно: убедитесь, что группа безопасности, которую вы выберете, позволяет

входящие соединения TCP по порту 6379 из экземпляра EC2. Если это не так, вы не сможете подключиться к узлам Redis.

В этом задании мы не будем загружать данные из S3, поэтому **Seed RDB file S3 location** можно пропустить.

Ежедневные резервные копии важны в большинстве случаев, и лучшие практики заключаются в том, чтобы использовать резервные копии с периодом удержания, который даст вам достаточно времени, чтобы действовать в случае, если что-то неожиданное произойдет. Для этого задания

11. Отключите **Enable automatic backups**.
12. Для **Maintenance window** укажите *No preference.*

Рекомендуется указывать день недели и время, когда ваше приложение имеет низкую рабочую нагрузку.

13. Нажмите **Create.**

Будет инициировано создание кластера Redis. Это может занять до 10 минут. Когда кластер Redis для ElastiCache будет готов к использованию, будет отображаться статус *available* и 2 узла.

1. В консоли **AWS Management Console** нажмите **Services** и выберите **RDS.**
2. Нажмите **Create database**.
3. Для **Choose a database creation method** выберите *Standard create*
4. Для **Engine type** выберите *MySQL*.
5. Для **Version** можно оставить рекомендуемую версию.
6. Для **Templates** укажите *Dev/Test*
7. В секции **Settings** укажите:

- **DB instance identifier**: my-rds
- **Master username**: student
- **Master password**: Pass.123
- **Confirm password**: Pass.123

8. В секции **DB instance size** укажите:

- *Burstable classes*
- *db.t2.micro*

Если тип экземпляра db.t2.micro не отображается, включите кнопку переключения Include previous generation classes .

9. В секции **Availability & durability** для **Multi-AZ deployment**, отметьте **Do not create a standby instance.**

10. В секции **Connectivity** укажите:

- **Virtual private cloud (VPC)**: Default VPC
- **Public access**: No
- Укажите группу безопасности

Важно: убедитесь, что выбранная группа безопасности позволяет входящие соединения TCP по порту 3306 из экземпляра EC2. Если это не так, вы не сможете подключиться к MySQL.

11. Раскройте секцию **Additional configuration**, и укажите:

- Снимите отметку **Enable automatic backups**
- Снимите отметку **Enable encryption**
- Снимите отметку **Enable Performance Insights**

- Снимите отметку **Enable Enhanced monitoring**
- Снимите отметку **Enable auto minor version upgrade**
- Снимите отметку **Enable deletion protection**

12. Прокрутите вниз страницы и нажмите **Create database**

Инициируется создание экземпляра RDS. Создание экземпляра RDS займет около 10 минут.

Пока создается экземпляр RDS, вы увидите баннер, предоставляющий учетные данные. Нажмите кнопку **View credential details** и копируйте учетные данные в текстовый редактор.

Выполнить создание и заполнение базы данных можно несколькими способами. Можно использовать MySQL Workbench, использовать командную оболочку mysql или консоль EC2.

Использование консоли EC2

При подготовке экземпляра EC2 был скопирован файл seed.sql. Чтобы выполнить этот sql-скрипт выполните в терминале EC2 следующую команду, вместо endpoint подставьте значение, которое вы скопировали в текстовый редактор:

mysql -h endpoint -P 3306 -u student -p < seed.sql

Если команда зависает, скорее всего, она заблокированы настройками группы безопасности. Убедитесь, что ваш экземпляр EC2 имеет доступ к группе безопасности, назначенной вашему экземпляру MySQL. Например, предположим, что экземпляр EC2 был назначен группе безопасности по умолчанию. Теперь вы можете изменить группу безопасности вашего экземпляра MySQL, отредактировать входящие правила и добавить правило MYSQL/Aurora, разрешая соединения по порту 3306 из любого экземпляра в группе безопасности.

При выполнении команды потребуется ввести пароль. Введите: *Pass.123*

Использование командной оболочки mysql

В консоли EC2 выполните следующую команду:

mysql -h endpoint -P 3306 -u student -p

При выполнении команды потребуется ввести пароль. Введите: *Pass.123*

Последовательно выполните следующие SQL-запросы (разделены пустой строкой) и изучите их назначение и результат:

CREATE database tutorial; USE tutorial;

CREATE TABLE planet (

id INT UNSIGNED AUTO_INCREMENT, name VARCHAR(30),

PRIMARY KEY(id));

INSERT INTO planet (name) VALUES ("Mercury"); INSERT INTO planet (name) VALUES ("Venus"); INSERT INTO planet (name) VALUES ("Earth"); INSERT INTO planet (name) VALUES ("Mars"); INSERT INTO planet (name) VALUES ("Jupiter"); INSERT INTO planet (name) VALUES ("Saturn"); INSERT INTO planet (name) VALUES ("Uranus"); INSERT INTO planet (name) VALUES ("Neptune");

Использование MySQL Workbench

При использовании MySQL Workbench исользуйте данные подключения, скопированные в текстовый редактор и запросы SQL, представленные выше (см.

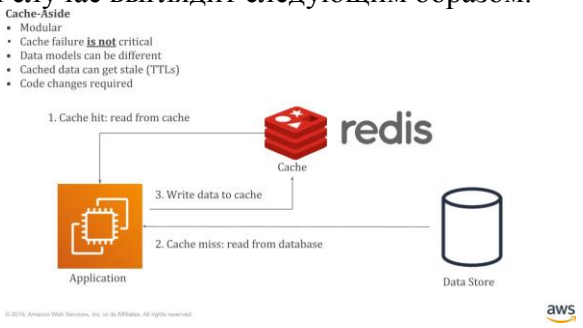
использование командной оболочки mysql).

В следующих заданиях будет использоваться таблица planet базы данных tutorial.

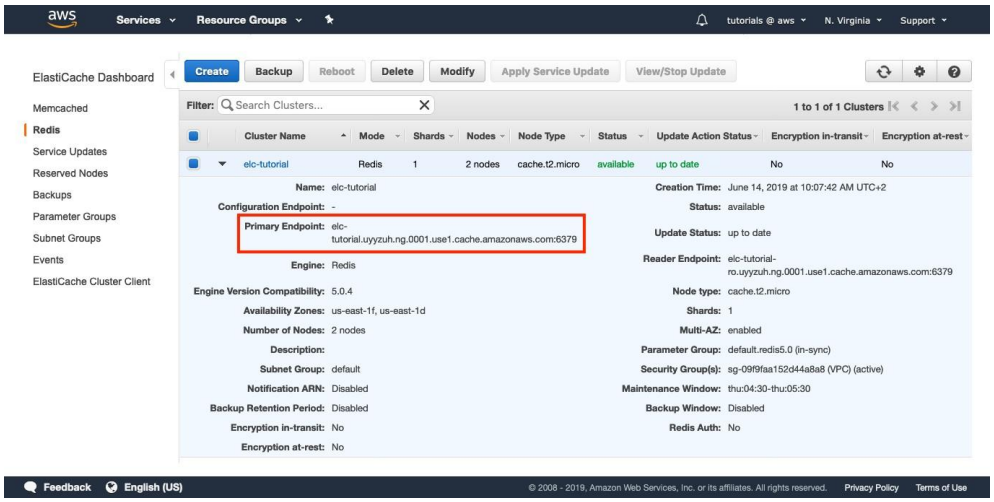
Реляционные базы данных являются краеугольным камнем большинства приложений. Однако когда речь заходит о масштабируемости и низкой задержке, можно сделать лишь очень многое для повышения производительности. Даже если вы добавите реплики для масштабирования чтения, существует физический предел, накладываемый дисковым хранилищем. Наиболее эффективной стратегией для преодоления этого ограничения является дополнение дисковых баз данных кэшированием в памяти.

Стратегия кэширования является одним из самых популярных вариантов повышения производительности базы данных. Когда приложению необходимо прочитать данные из базы данных, оно сначала запрашивает кэш. Если данные не найдены, приложение запрашивает базу данных и заполняет кэш результатом. Существует множество способов очистить кэш, если соответствующие записи были изменены в основной базе данных, Redis предоставляет функцию ограничения времени жизни (TTL, Time To Live).

Решение в данном случае выглядит следующим образом:



1. В консоли **AWS Management Console** нажмите **Services** и выберите **ElastiCache**.
2. В навигационном меню слева нажмите **Redis**
3. Выберите кластер *elc-lab*
4. Скопируйте значение **Primary Endpoint** в текстовый редактор. Значение **Primary Endpoint** выглядит примерно так:



Проверка подключения к Redis

5. Переключитесь к терминалу EC2

6. Введите следующую команду, чтобы запустить интерпретатор Python:
Python

7. Последовательно выполните следующие команды и изучите их назначение и результат, заменив **endpoint** на значение, которое вы скопировали в текстовый редактор:

import redis
client = redis.Redis.from_url('redis://**endpoint**:6379') client.ping()

При выполнении последней команды вы должны получить вывод *True.*

Это подтверждает доступ к Redis.

Примечание: Если выполнение зависает, это означает, что команда блокируется настройками Группы безопасности. Убедитесь, что ваш экземпляр EC2 имеет доступ к группе безопасности, назначенной вашему экземпляру ElastiCache. Например, предположим, что экземпляр EC2 был назначен группе безопасности по умолчанию. Вы можете изменить группу безопасности экземпляра Amazon ElastiCache и добавить правило Custom TCP, разрешая соединения по порту 6379 из любого экземпляра группы безопасности по умолчанию.

Настройка переменных окружения

При подготовке экземпляра EC2 был скопирован файл **example.py**. Для выполнения этого файла необходимо указать переменные окружения, хранящие данные подключения к Redis и MySQL.

8. В терминале EC2 выполните следующие команды, заменив значения на соответствующие данные, которые вы скопировали в текстовый редактор:

export REDIS_URL=redis://**your_redis_endpoint**:6379/ export DB_HOST=**your_mysql_endpoint**

export DB_USER=student

export DB_PASS=**your_admin_password** export DB_NAME=tutorial

Кеширование результатов SQL-запроса

9. Изучите следующий фрагмент кода на языке Python:

```
def fetch(sql):
result = cache.get(sql) if result:
return deserialize(result)
else:
result = db.query(sql)
cache.setex(sql, ttl, serialize(result)) return result
```

Redis в качестве ключа использует SQL-запрос, просматривает кэш, чтобы узнать, присутствует ли значение. Если значение не присутствует, то выполняется SQL-запрос к базе данных. Результат запроса хранится в Redis. Переменная ttl должна быть установлена на разумное значение, зависящее от характера приложения. Когда срок действия ttl истекает, Redis очищает ключ и освобождает связанную память.

С точки зрения стратегии, недостаток этого подхода заключается в том, что при изменении данных в базе данных изменения не будут доступны

пользователю, если предыдущий результат был кэширован и его ttl еще не истек.

Пример того, как можно использовать функцию извлечения данных из кеша:

```
print(fetch("SELECT * FROM planet"))
```

Результат будет следующим:

```
[{'id': 10, 'name': 'Mercury'},
{'id': 11, 'name': 'Venus'},
{'id': 12, 'name': 'Earth'},
{'id': 13, 'name': 'Mars'},
{'id': 14, 'name': 'Jupiter'},
{'id': 15, 'name': 'Saturn'},
{'id': 16, 'name': 'Uranus'},
{'id': 17, 'name': 'Neptune'}]
```

Форма представления результата:

Продемонстрировать ход выполнения преподавателю. Показать свою учетную запись. Ответить на вопросы.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Конечно, это очень простой пример, но ваше приложение будет работать гораздо быстрее, реализуя эту схему кэширования, где нет никакой разницы между результатом, поступающим из кэша, и результатом, поступающим прямо из базы данных.

Запись данных из кэша в базу данных

10. Второй пример – запись данных из кэша в базу данных. Изучите следующий фрагмент кода на языке Python:

```
def planet(id):
key = "planet:" + str(id) result = cache.hgetall(key)
if result:
return result
else:
sql = "SELECT `id`, `name` FROM `planet` WHERE `id`=%s" result =
db_record(sql, (id,))
if result:
cache.hmset(key, result) cache.expire(key, ttl)
return result
```

Ключевое отличие Redis – это «плоское» хранение, но есть последовательность пар «*ключ:значение*». Например ключом для записи с ID 1 будет "*planet:1*". Хотя этот фрагмент достаточно хорош для того, чтобы показать общий шаблон, возможно больше абстракции: один модуль может отвечать за генерацию ключей, другой может заботиться о создании записей SQL и т.д.

В примере происходит извлечение записи либо из кэша, либо из базы данных, и аналогичным образом реализуется функция, отвечающая за запись данных в базу данных.

Истечение срока годности

В двух примерах использовались Time To Live или ttl, после чего Redis удаляет ключ. Хотя в большинстве случаев этого достаточно, чтобы как можно скорее удалить устаревшие данные из кэша. Существуют и другие стратегии кэширование. Кроме команды EXPIRE, Redis также предоставляет EXPIREAT, которая позволяет указать точную дату и время, когда ключ должен быть удален. Он принимает в качестве параметра абсолютную дату в формате Unix (т.е. секунды, прошедшие с 1 января 1970 года).

Настройка механизма кеширования

Когда объем данных превышает установленную максимальную память, Redis может использовать различные способы реагирования в зависимости от выбранной политики очищения кэша. По умолчанию ElastiCache для Redis настроен для удаления из памяти наиболее давно использованных ключей с значением ttl. Параметр политики очищения называется *maxmemory-policy*, а значение по умолчанию в ElastiCache является *volatile-lru*. Другим вариантом является политика *volatile-ttl,* согласно которой Redis очищает память, удаляя ключи с наименьшим ttl.

Проверка

11. Выполните Python-скрипт example.py и изучите последовательность его работы.

12. Вы можете изменять скрипт для проверки различных режимов работы Redis.

Удаление кластера Redis

1. В консоли **AWS Management Console** нажмите **Services** и выберите **ElastiCache**.
2. В навигационном меню слева нажмите **Redis**
3. Выберите Redis-кластер *elc-lab*
4. Нажмите **Delete**.
5. В окне подтверждения на запрос **Create a final backup?** Выберите *No*. Статус кластера будет изменен на *deleting*.

Удаление базы данных RDS

6. В консоли **AWS Management Console** нажмите **Services** и выберите **RDS**.
7. В навигационном меню слева нажмите **Databases**
8. Выберите экземпляр базы данных *my-rds*
9. В меню **Actions** выберите **Delete**.
10. В окне подтверждения удаления отмените выбор **Create final snapshot?** и выберите соглашение «*I acknowledge…*».
11. В поле подтверждения введите *delete me* и нажмите **Delete**.