Тема: Разгръщане на система за споделяне не файлове в AWS

Предмет: Приложно-програмни интерфейси за работа с облачни архитектури с Амазон Уеб Услуги (AWS)

Изготвил: Даниел Пенчев, фн: 62114, имейл: danielpenchev@abv.bg

Лектор: Милен Петров, година: 2021

Съдържание

1 Условие

Модифициране на архитектурата на вече съществуваща система с цел мигрирането ѝ в/у клауд инфраструктура. Разделяне на системата на слоеве, всеки от който е независим от другите с цел по-добра модифицируемост и скалируемост.

2 Въведение

Целта на този проект е да се мигрира системата в/у клауда използвайките част от сървисите на AWS като:

- AWS EFS за съхраняване на голям обем от често достъпвани файлове
- AWS ELB за управление на натоварването в/у отделните инстанции на части системата
- AWS RDS за съхраняване на всички потребителски данни
- AWS Gateway за достъп на EC2 инстанциите до интернет

3 Теория

AWS EFS е услуга за съхранение на файлове, базирана в облака, за приложения и натоварвания, които се изпълняват в публичния облак на Amazon Web Services. AWS автоматично деплойва и управлява инфраструктурата на еластична файлова система, която се разпределя м/у неограничен брой сървъри, за да се избегнат проблеми в производителността. AWS EFS предоставя възможност за динамично променяне на капацитета за съхранение с цел поемане на натоварването от EC2 инстанциите и достъп до файлове чрез добре дефинирано API Тази услуга е проектирана да бъде силно достъпна и издръжлива за множество EC2 инстанции, използващи услугата. За да постигне тази цел, да бъде издръжлива и достъпна, тя съхранява всеки обект в множество AZ като ги репликира. Всяка инстанция Amazon EFS може бъда закачена на съществуваща файлова система на EC2 инстанция, като единственото условие е EC2 инстанцията да използва Linux AMI, и позволява управлението ѝ през AWS конзолата или AWS CLI.

AWS ELB е услуга предоставящата и управляващата Load Balancer инстанция, която автоматично разпределя входящия трафик между EC2 инстанциии и контейнери, работещи в е една или множество AZs. Освен се грижи за наблюдението над приемниците на входящия трафик по отношение на тяхната достъщност, като по този начин разпределя входящия трафик само между достъпните

приемници. ELB се грижи за скалирането на Load Balancer инстанцията взимайки предвид натоварването от входящия трафик. AWS услугата предлага различни разни разновидности на Load Balancer инстанция - Application Load Balancer, Network Load Balancer и Classic Load Balancer, като в проекта ще използваме единствено ALB. Application Load Balancer се позиционира на седмия слой (приложния слой) от ОСИ модела, като се грижи за разпределянето на HTTP/HTTPS трафика.

AWS RDS е услуга, улесняваща управлението и настройването и скалирането на релационна база данни в облака. Тя осигурява икономичен и променлив капацитет, като същевременно автоматизира времеемките административни задачи, като хардуерно осигуряване, настройка на база данни, закърпване и архивиране. Освобождава клиентите от отговорностите свързани с поддръжката на базата, като им позволява изцяло да се фокусират върху приложенията си. Amazon RDS се предлага на няколко типа инстанции на база данни - оптимизирани откъм памет, производителност или I/O.

AWS NAT Gateway е услуга отговорна за създаването и управлението на NAT инстанции. Обикновено се използва когато инстанциите в частни подмрежи трябва да имат достъп до ресурси, които не в VPC-то на Амазон, като същевременно инстанциите не могат да бъдат директно достъпвани отвън.

4 Използвани технологии

- **AWS EC2** - Amazon Linux 2 AMI
- **AWS EFS**
- **AWS RDS** - version PostgreSQL 11.11-R1
- **AWS ELB** - Application Load Balancer
- **AWS VPC**

5 Инсталация и настройки

1. Създаване на VPC
2. Създаване на Internet Gateway и закачането му към VPC-то
3. Създаване на публична подмрежа за EC2 инстанция служеща като NAT - "subnet-ngw"

4. Създаване на 2 публични подмрежа в различни AZ за слушачите на ALB-то - "web-app-pub-1"и "web-app-pub-2"

5. Създаване на 2 частни подмрежи в различни AZ за EC2 инстанции, в които работят системните сървъри - "web-app-prv-2"и "web-app-prv-1"

6. Създаване на 2 частни подмрежи в различни AZ за базите от данни създадени чрез AWS RDS - "subnet-prv-db-1"и "subnet-prv-db-2"

7. Създаване на рутираща таблица асоциирана с "subnet-ngw"с допълнително правило трафика отправен извън AWS VPC-то да минава през Internet Gateway-a

8. Създаване на рутираща таблица асоциирана с "web-app-pub-1"и "web-app-pub-2"с допълнително правило правило трафика отправен извън AWS VPC-то да минава през Internet Gateway-a

9. Създаване на рутираща таблица асоциирана с "web-app-prv-2"и "web-app-prv-1"с допълнително правило трафика отправен извън AWS VPC-то да бъде отправен през "subnet-ngw"

10. Създаване на рутираща таблица асоциирана с "subnet-prv-db-1"и "subnet-prv-db-2".

11. Създаване на security-group-и "ngw-sg "alb-sg "rds-sg"и "web-app-sg"

12. Добавяне на правила към security group-ата "ngw-sg"(отговаряща изцяло за NAT инстанцията), които позволява входен HTTP и SSH трафик от "web-app-sg"

13. Добавяне на правила към security group-ата "web-app-sg"(отговаряща за EC2 инстанциите със системните сървъри), които позволяват SSH трафик от "ngw-sg"и HTTP трафик и "alb-sg"

14. Добавяне на правила към security group-ата "alb-sg"(отговаряща за Application Load Balancer инстанцията), които позволяват входен HTTP трафик от всякъде и изходен HTTP трафик само към "web-app-sg"

15. Добавяне на правило към security group-ата "rds-sg"(отговаряща за бази от данни), които позволяват входен/изходен трафик на порт 5432 от/към "web-app-sg"

16. Създаване на EC2 инстанция с amazon nat ami. Тази инстанция служи като NAT gateway. При създаването на инстанцията се използва security group-ата "ngw-sg"и подмрежата "subnet-ngw"

17. Създаване на Postgres база + реплика чрез AWS RDS (едната от тях е read реплика), избирайки 2те частни подмрежи "subnet-prv-db-1"и "subnet-prv-db-2"и security group-ата "rds-sg"

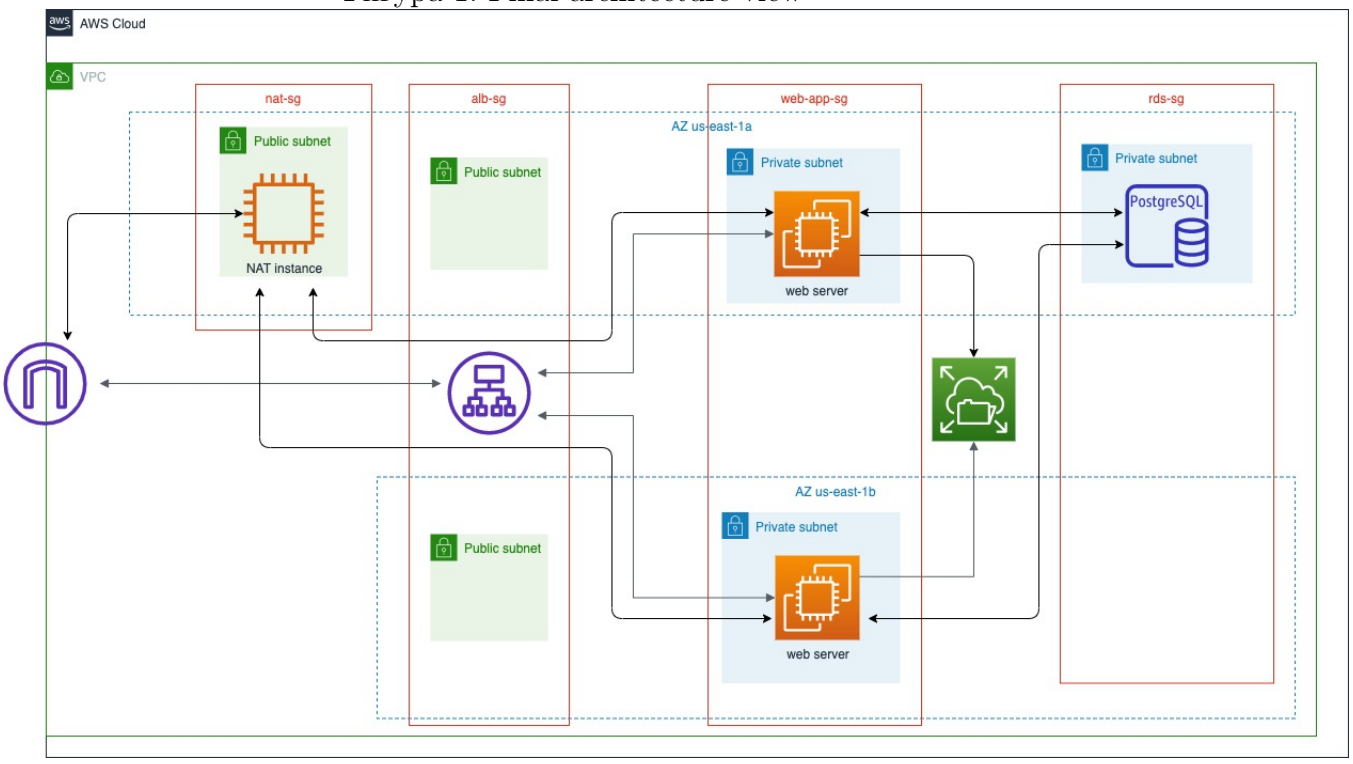
18. Създаване на файлова система използвайки AWS EFS

19. Създаване на 2 EC2 инстанции с амазон linux2 ami с скрипт за инстанлиране на надстройване на web server-а. При създаването на инстанциите се използват подмрежите "web-app-prv-2"и "web-app-prv-1"и security group-ата "web-app-sg"

20. Създаване на EC2 instance target група, избирайки VPC, то което създадохме на стъпка 1 и двете EC2 инстанции на системния сървър

21. Създаване на ALB чрез AWS ELB, използваки target групата от предната стъка, security групата "alb-sg"и подмрежите "web-app-pub-1"и "web-app-pub-2"

Фигура 1: Final architecture view



6 Кратко ръководство на потребителят

Системата предоставя множество функционалности за споделяне на файлове. Съществува само един вид потребител на системата - регистриран потребител. Регистрираните потребители могат да се групират и в зависимост от принадлежността си към дадена група могат да достъпват различни файлове, качени от членовете на тази група специално за тази група Даден потребител може да е член на 0..N групи, като не може една инстанция на файл да се сподели между групи - потребителят ако е член на двете групи може да качи файла поотделно за двете групи. Всяка група си има администратор и членове, като администратора е създателя на групата. Когато администратора си изтрие профила, то се избира произволен член, който да поеме тази роля. Системата може да се достъпва по 2 начина - директния начин е чрез използване на REST API-то на системата и другия вариант е чрез специален конзолен клиент. Повече информация относно системата и клиента може да бъде намерена тук.

7 Примерни данни

При регистрация потребителят трябва да въведе поне 8 символна парола, която трябва да съдържа 1 малка буква, 1 главна буква, 1 символ и една цифра.

При създаване на група трябва да се въведе уникално име на групата. След създаване на групата, потребителят е нейн собственик и може да качва/изтрива/сваля файлове от нея.

8 Описание на програмния код

8.1 Конзолен клиент

Кода на конзолния клиент и пълната му документация може да бъде намерена тук.

8.2 Уеб сървър

Кода на уеб сървъра и пълната му документация може да бъде намерена тук.

9 Приноси на студента, ограничения и възможности за бъдещо развитие

Системата е разработена от мен във връзка с курса "Въведение в Golang". Допълнителните усилия с цел качване на системата в/у AWS бяха главно свързани с настройването на инфраструктурата и модифициране на кода на сървъра с цел по-добра скалируемост. Това което може да се усъвършенства в системата, по конкретно в сървърната част е мястото от където се извършва изтриването на ресурсите на дадена група. С възможностите на AWS Lambda е възможно да се настрои периодична задача, чиято единствена цел е да изтрива тези ресурси. Всяка инстанция на сървъра съдържа в себе си подобна периодична задача и с AWS Lambda тази функционалност е се изнесе, като по този начин ще се постигне по-добро използване на ресурсите и няма да има нужда от синхронизационни механизми.

Списък на фигурите

Литература

[1] AWS Documentation