

Тема: Разширяване на проект по Уеб технологии - Презентации по Web(календар) с AWS EC2, AWS RDS, AWS CodeCommit

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

Изготвил: Дияна Йорданова, фн: 81547, имейл: dijanaj@uni-sofia.bg

Лектор: доц. д-р Милен Петров, година: 2020

Съдържание

1	Условие	3
2	Въведение	3
3	Теория	4
4	Използвани технологии	5
5	Инсталация и настройки	5
5.1	AWS Relational Database Service	5
5.2	AWS CodeCommit	6
5.3	AWS Elastic Compute Cloud	8
5.4	Amazon Machine Images - Backup	11
6	Кратко ръководство за потребителя	12
7	Примерни данни	13
8	Описание на програмния код	13
8.1	Презентационен слой	13
8.2	Домейн слой	14
8.3	Структура на релационна база данни	15
8.4	Обща картина на Уеб приложението	15

9 Приноси на студента, ограничения и възможности за бъдещо развитие	16
10 Какво научих	16
11 Списък с фигури	17
12 Използвани източници	17

1 Условие

Разширяване на съществуващ проект по Уеб технологии с поне една услуга на AWS. Тех документация на разширения проект и диаграми, изобразяващи кратко ръководство за потребителя.

Условие на разширения проект: Уеб приложението подпомага организацията за представяне на презентации, по предмета Web технологии, от студенти. В системата е изградена функционалност за регистриране и вход на студенти. При вход в системата всеки студент трябва да въведе своя факултетен номер и парола. С тази функционалност всеки студент има права да редактира календара като избере час и дата, които са свободни. Ако няма регистриран студент, то неговите права са ограничени – той има права само да преглежда календара и информацията в него, но не и да извършва промени върху календара. Всеки регистриран потребител има възможност да запази дата за представяне на презентация. Запазването се осъществява само ако избраната дата вече не е запазена от друг студент. За всяка запазена дата от един студент има възможност за изтриване на датата и за избиране на нова такава.

2 Въведение

Документацията ще представи основните идеи за създаването на проекта и неговото разширение с AWS услуги.

Разширяването на проекта е вдъхновен от курса по Приложно-програмни интерфейси за работа с облачни архитектури с Амазон Уеб Услуги (AWS), в който бяха засегнати различни AWS услуги. За целта на проекта са използвани следните услуги:

- AWS Relational Database Service (RDS)
- AWS Elastic Compute Cloud (EC2)
- AWS CodeCommit

В проекта е засегнат проблем, чието решение подпомага участниците от курса по Web технологии. В курса всяка година студентите избират теми, върху които да разработят реферат. След завършване на реферата всеки студент представя и запознава своите колеги с неговата тема. За целта се запазва определен час и ден, в който това да се осъществи. На този етап за запазване на презентации се използва споделена таблица в Google Docs. Недостатъкът при използването на таблица в Google Docs, е че всяка година трябва да се създаде нова таблица и

да се въведат валидни дати и час, в които всеки студент да запише своите теми. Това затруднява работата на всички участници в курса. При въвеждане на данни от двама студенти за една дата се губят данни, което води до опасността студент да няма запазена дата за представяне на презентация. С помощта на създадената система, бързо и лесно, всеки студент може да запази дата за своята презентация.

3 Теория

В този раздел ще бъдат разгледани само по-ключови решения, засягащи архитектурата на приложението и AWS услугите, използвани за разширяването му.

За разширяването на Уеб приложението са използвани следните AWS услуги:

AWS Relational Database Service (RDS): услуга, която позволява лесно настройване, опериране и мащабиране на релационни бази данни като MariaDB, която се използва в разглеждания проект. AWS RDS се грижи за голяма част от административните задачи по поддържането на релационната база данни. Част от тези задачи са security updates, availability, data encryption, настройване на различни видове аларми и създаване на backup на базата. Това позволява на потребителя да се фокусира върху менажирането на съдържанието в базата.

AWS Elastic Compute Cloud (EC2): уеб услуга, която осигурява сигурен, преоразмерим изчислителен капацитет в cloud-а чрез използване на виртуални машини. Този подход осигурява пълен контрол върху всички компютърни ресурси и позволява работа в сигурна среда на Amazon. Също така, предоставя лесна интеграция с други AWS услуги като например AWS RDS чрез Virtual Private Cloud (VPC), което е частна виртуална мрежа в cloud-а на Amazon. Ресурсите, свързани към дадена мрежа имат видимост помежду си, като контролът на достъп се осъществява чрез security group, в която се добавят дадени правила, ограничаващи достъпа. Пример за такова правило е отваряне и затваряне на дадени портове.

AWS CodeCommit: услуга, която дава възможност за управление на source code-а на приложението. Тя предоставя подсигурени хранилища за код, базирани на Git. Достъпът до тези хранилища се контролира чрез Identity Access Management (IAM). Като повечето услуги на AWS, CodeCommit лесно се интегрира с други услуги като например EC2. Това позволява бързо публикуване на променения source code.

4 Използвани технологии

За изграждането на web приложението са използвани следните стандартни технологии и подходи:

- **Front-end:** HTML5, CSS3, JavaScript
- **Back-end:** PHP version 'PHP 7.3.11'
- **Database:** MySQL (MariaDB) version '10.4.8-MariaDB'
- **Cloud:**
 - AWS RDS (MariaDB)
 - AWS EC2 'Amazon Linux 1 AMI 2.0.20200406.0 x86-4 HVM gp2'
 - AWS CodeCommit (Git)

5 Инсталация и настройки

5.1 AWS Relational Database Service

Създава се MySQL (MariaDB) база данни с необходимата за проекта архитектура. Изпълняват се следните стъпки:

Security group ->

inbound rules ->

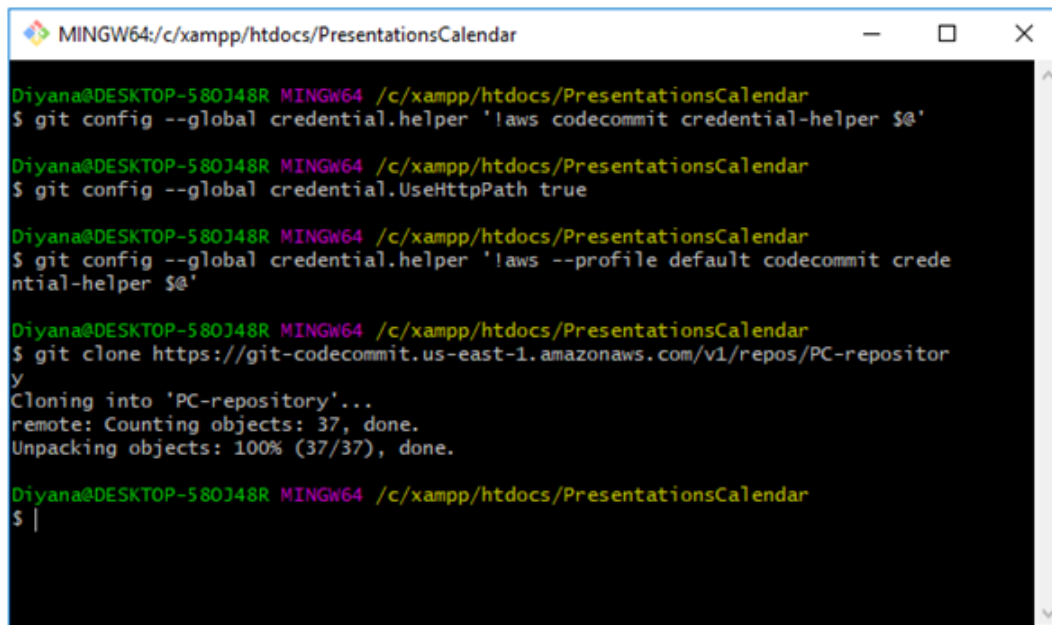
add rule ->

CP traffic (custom rule) with default IPs

Свързването към базата се осъществява с помощта на допълнителни инструменти като например MySQL Workbench. В инструмента се въвежда endpoint и порт на използваната база. Предоставят се username и password, създадени при инициализацията на базата. Import-ва се .dump файл, който създава всички таблици със съответните ограничения и въвежда всички данни, налични до този момент в тях. След успешното изпълнение на стъпките, базата данни е готова за употреба. За по-голяма сигурност AWS предоставя възможност за създаване на Snapshot на базата. Това е прост механизъм за Backup, който запазва текущото състояние на базата и при настъпил проблем позволява лесно възвръщане към предходен Snapshot.

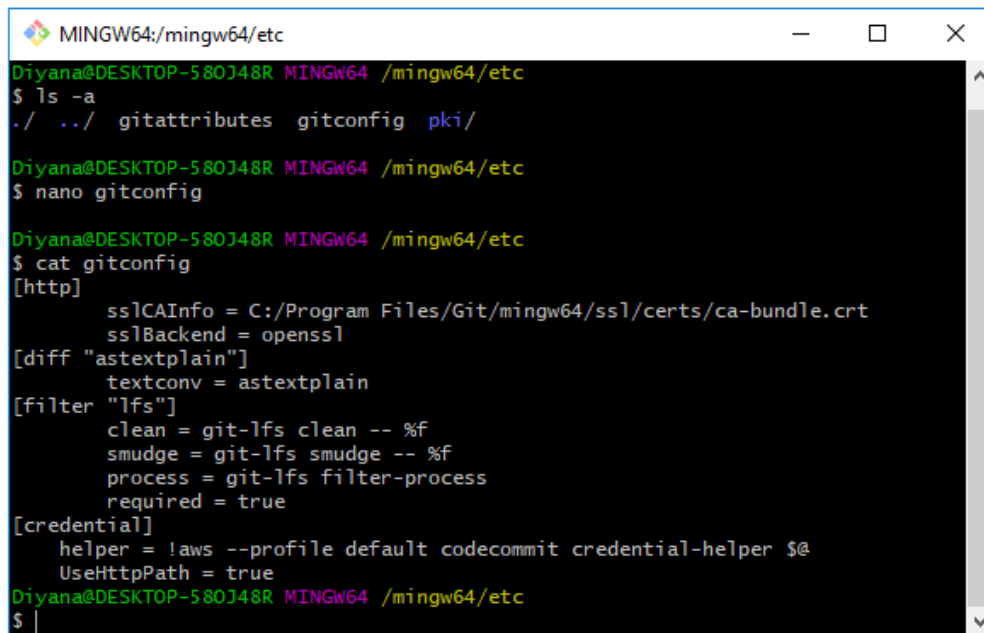
5.2 AWS CodeCommit

Създава се хранилище(AWS CodeCommit - Repository) от AWS Console. За клониране на хранилището се добавят следните git команди от фигура 1. Те синхронизират конфигурационните файлове, което дава възможност за authentication в CodeCommit.



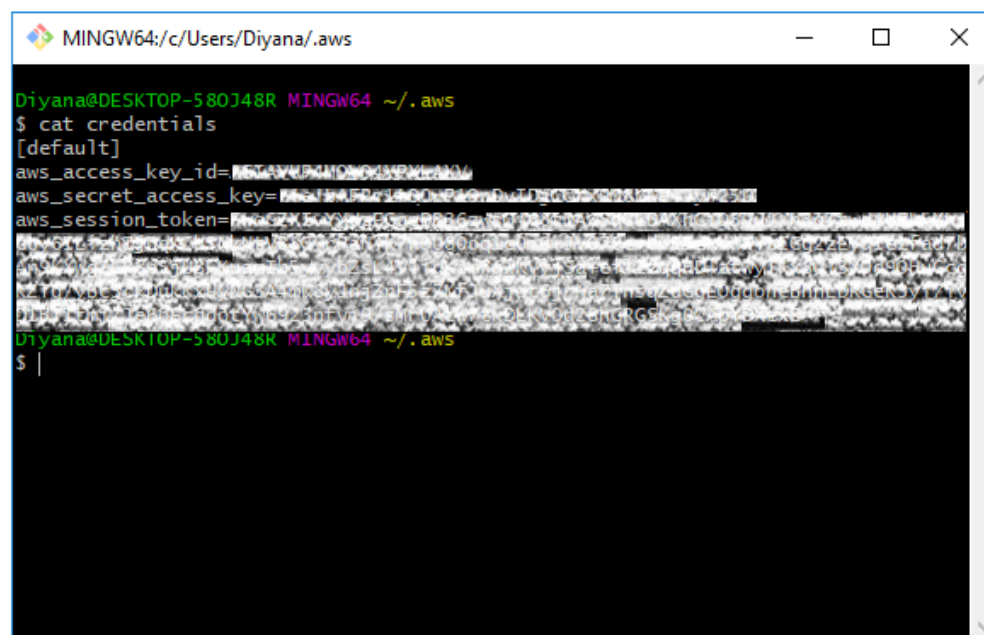
```
MINGW64:/c:/xampp/htdocs/PresentationsCalendar
Diyana@DESKTOP-580J48R MINGW64 /c:/xampp/htdocs/PresentationsCalendar
$ git config --global credential.helper 'aws codecommit credential-helper $@'
Diyana@DESKTOP-580J48R MINGW64 /c:/xampp/htdocs/PresentationsCalendar
$ git config --global credential.UseHttpPath true
Diyana@DESKTOP-580J48R MINGW64 /c:/xampp/htdocs/PresentationsCalendar
$ git config --global credential.helper 'aws --profile default codecommit credential-helper $@'
Diyana@DESKTOP-580J48R MINGW64 /c:/xampp/htdocs/PresentationsCalendar
$ git clone https://git-codecommit.us-east-1.amazonaws.com/v1/repos/PC-repository
Cloning into 'PC-repository'...
remote: Counting objects: 37, done.
Unpacking objects: 100% (37/37), done.
Diyana@DESKTOP-580J48R MINGW64 /c:/xampp/htdocs/PresentationsCalendar
$ |
```

Фигура 1: Configuring and cloning the repository



```
MINGW64:/mingw64/etc
Diyana@DESKTOP-580J48R MINGW64 /mingw64/etc
$ ls -a
./ ../ gitattributes gitconfig pki/
Diyana@DESKTOP-580J48R MINGW64 /mingw64/etc
$ nano gitconfig
Diyana@DESKTOP-580J48R MINGW64 /mingw64/etc
$ cat gitconfig
[http]
  sslCAInfo = C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
  sslBackend = openssl
[diff "astextplain"]
  textconv = astextplain
[filter "lfs"]
  clean = git-lfs clean -- %f
  smudge = git-lfs smudge -- %f
  process = git-lfs filter-process
  required = true
[credential]
  helper = !aws --profile default codecommit credential-helper $@
  UseHttpPath = true
Diyana@DESKTOP-580J48R MINGW64 /mingw64/etc
$ |
```

Фигура 2: Local folder Git -> [credential]



```
MINGW64:/c/Users/Diyana/.aws
Diyana@DESKTOP-580J48R MINGW64 ~/.aws
$ cat credentials
[default]
aws_access_key_id=AKIAI44QH8DHBVSQJ3ATV
aws_secret_access_key=AKIAI44QH8DHBVSQJ3ATV
aws_session_token=AKIAI44QH8DHBVSQJ3ATV
Diyana@DESKTOP-580J48R MINGW64 ~/.aws
$ |
```

Фигура 3: Local folder .aws -> [default]

Извършва се локално клониране на целия проект в новосъздадена папка. За публикуването на проекта в cloud-a се използват стандартните git команди - commit и push.

ЗАБЕЛЕЖКА: След изтичане на сесията от AWS educate, файлът credential от папка .aws трябва да се обнови с новите credential данни.

5.3 AWS Elastic Compute Cloud

Създава се EC2 инстанция от AWS Console. В стъпка 3: Configure Instance Details добавяме следните команди за стартиране на основни сървиси като Apache и допълнителни за него инструменти:

▼ Advanced Details

Metadata accessible ⓘ	Enabled
Metadata version ⓘ	V1 and V2 (token optional)
Metadata token response hop limit ⓘ	1
User data ⓘ	<input checked="" type="radio"/> As text <input type="radio"/> As file <input type="checkbox"/> Input is already base64 encoded

```
#!/bin/bash
sudo yum install httpd -y
sudo service httpd start
sudo systemctl enable httpd
```

Фигура 4: Advanced Details - EC2

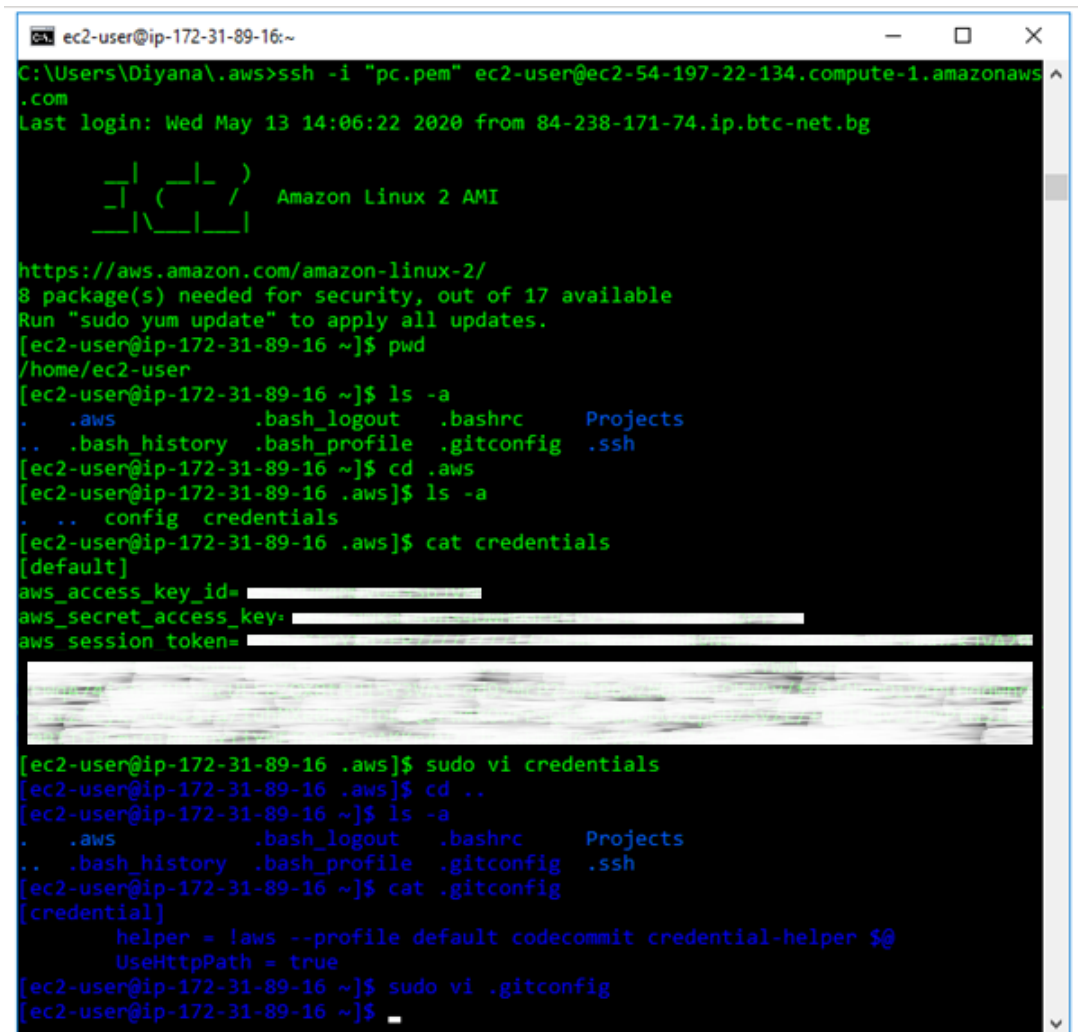
При създаване на инстанцията се създава нова security група, в която добавя-е следните правила:

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	0.0.0.0/0	-
HTTP	TCP	80	::/0	-
All traffic	All	All	sg-4999be67 (default)	-
SSH	TCP	22	0.0.0.0/0	-
SSH	TCP	22	::/0	-
MYSQL/Aurora	TCP	3306	0.0.0.0/0	-
MYSQL/Aurora	TCP	3306	::/0	-

Фигура 5: Security Group - Rules

ЗАБЕЛЕЖКА: Новосъздадена инстанция EC2 и RDS трябва да се намират в един и същи Virtual Private Cloud(VPC).

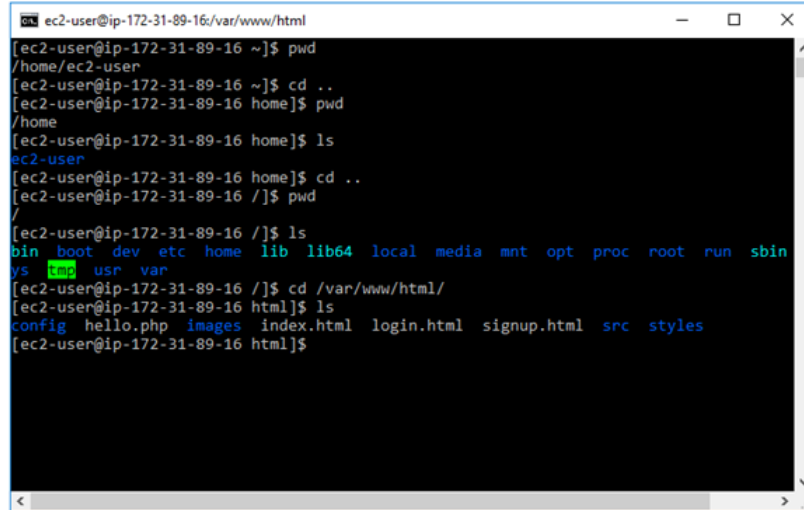
След успешното създаване на инстанция EC2 се свързваме с нея. Аналогично на конфигурацията на локалната среда, конфигурираме файла `gitconfig` и файла `credentials` от папка `.aws` на машината в cloud-a. Това позволява клонирането на хранилището.



```
ec2-user@ip-172-31-89-16:~  
C:\Users\Diyana\.aws>ssh -i "pc.pem" ec2-user@ec2-54-197-22-134.compute-1.amazonaws.com  
Last login: Wed May 13 14:06:22 2020 from 84-238-171-74.ip.btc-net.bg  
  
  _ _ _ _  
  | | | |  
  |_|_|_| Amazon Linux 2 AMI  
  
https://aws.amazon.com/amazon-linux-2/  
8 package(s) needed for security, out of 17 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-89-16 ~]$ pwd  
/home/ec2-user  
[ec2-user@ip-172-31-89-16 ~]$ ls -a  
. .aws .bash_logout .bashrc Projects  
.. .bash_history .bash_profile .gitconfig .ssh  
[ec2-user@ip-172-31-89-16 ~]$ cd .aws  
[ec2-user@ip-172-31-89-16 .aws]$ ls -a  
.. config credentials  
[ec2-user@ip-172-31-89-16 .aws]$ cat credentials  
[default]  
aws_access_key_id= [REDACTED]  
aws_secret_access_key= [REDACTED]  
aws_session_token= [REDACTED]  
  
[ec2-user@ip-172-31-89-16 .aws]$ sudo vi credentials  
[ec2-user@ip-172-31-89-16 .aws]$ cd ..  
[ec2-user@ip-172-31-89-16 ~]$ ls -a  
. .aws .bash_logout .bashrc Projects  
.. .bash_history .bash_profile .gitconfig .ssh  
[ec2-user@ip-172-31-89-16 ~]$ cat .gitconfig  
[credential]  
    helper = !aws --profile default codecommit credential-helper $@  
    UseHttpPath = true  
[ec2-user@ip-172-31-89-16 ~]$ sudo vi .gitconfig  
[ec2-user@ip-172-31-89-16 ~]$
```

Фигура 6: Configure EC2

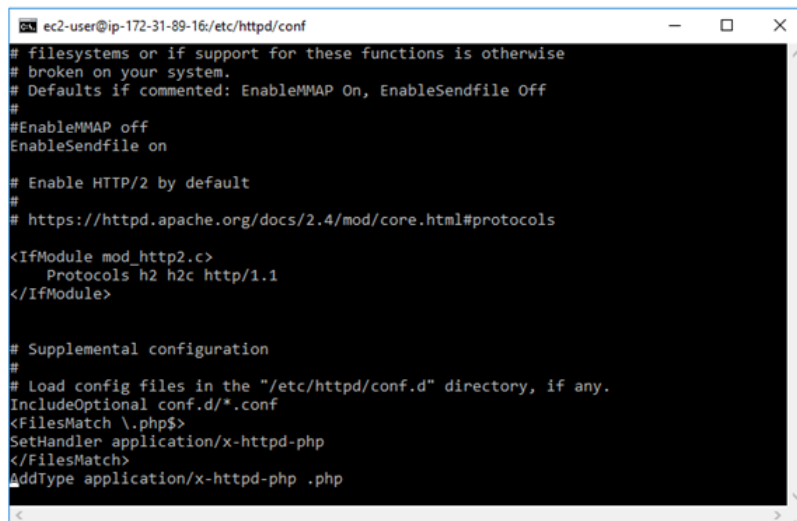
За стартирането на проекта през Apache Server от cloud-a, всички файлове от проекта се преместват в директория `/var/www/html` с команда:
`sudo cp -r "new-directory-name" /var/www/html`



```
ec2-user@ip-172-31-89-16:/var/www/html
[ec2-user@ip-172-31-89-16 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-89-16 ~]$ cd ..
[ec2-user@ip-172-31-89-16 home]$ pwd
/home
[ec2-user@ip-172-31-89-16 home]$ ls
ec2-user
[ec2-user@ip-172-31-89-16 home]$ cd ..
[ec2-user@ip-172-31-89-16 /]$ pwd
/
[ec2-user@ip-172-31-89-16 /]$ ls
bin  boot  dev  etc  home  lib  lib64  local  media  mnt  opt  proc  root  run  sbin
ys  tmp  usr  var
[ec2-user@ip-172-31-89-16 /]$ cd /var/www/html/
[ec2-user@ip-172-31-89-16 html]$ ls
config  hello.php  images  index.html  login.html  signup.html  src  styles
[ec2-user@ip-172-31-89-16 html]$
```

Фигура 7: Project files that would be run by the Apache server

Добавя се допълнителна конфигурация към файла **httpd.conf** на Apache, намиращ се в директория **/etc/httpd/conf**:



```
ec2-user@ip-172-31-89-16:/etc/httpd/conf
# filesystems or if support for these functions is otherwise
# broken on your system.
# Defaults if commented: EnableMMAP On, EnableSendfile Off
#
#EnableMMAP off
EnableSendfile on

# Enable HTTP/2 by default
#
# https://httpd.apache.org/docs/2.4/mod/core.html#protocols

<IfModule mod_http2.c>
    Protocols h2 h2c http/1.1
</IfModule>

# Supplemental configuration
#
# Load config files in the "/etc/httpd/conf.d" directory, if any.
IncludeOptional conf.d/*.conf
<FilesMatch \.php$>
    SetHandler application/x-httpd-php
</FilesMatch>
AddType application/x-httpd-php .php
```

Фигура 8: Configure Apache

Изпълняват се команди за допълнително конфигуриране и стартиране на PHP:

```
sudo yum install php
sudo yum install php-pdo
sudo yum install php-pdo_mysql
sudo service httpd restart
```

Приложението се достъпва чрез **Public DNS (IPv4)** или от **IPv4 Public IP** на създадената инстанция EC2.

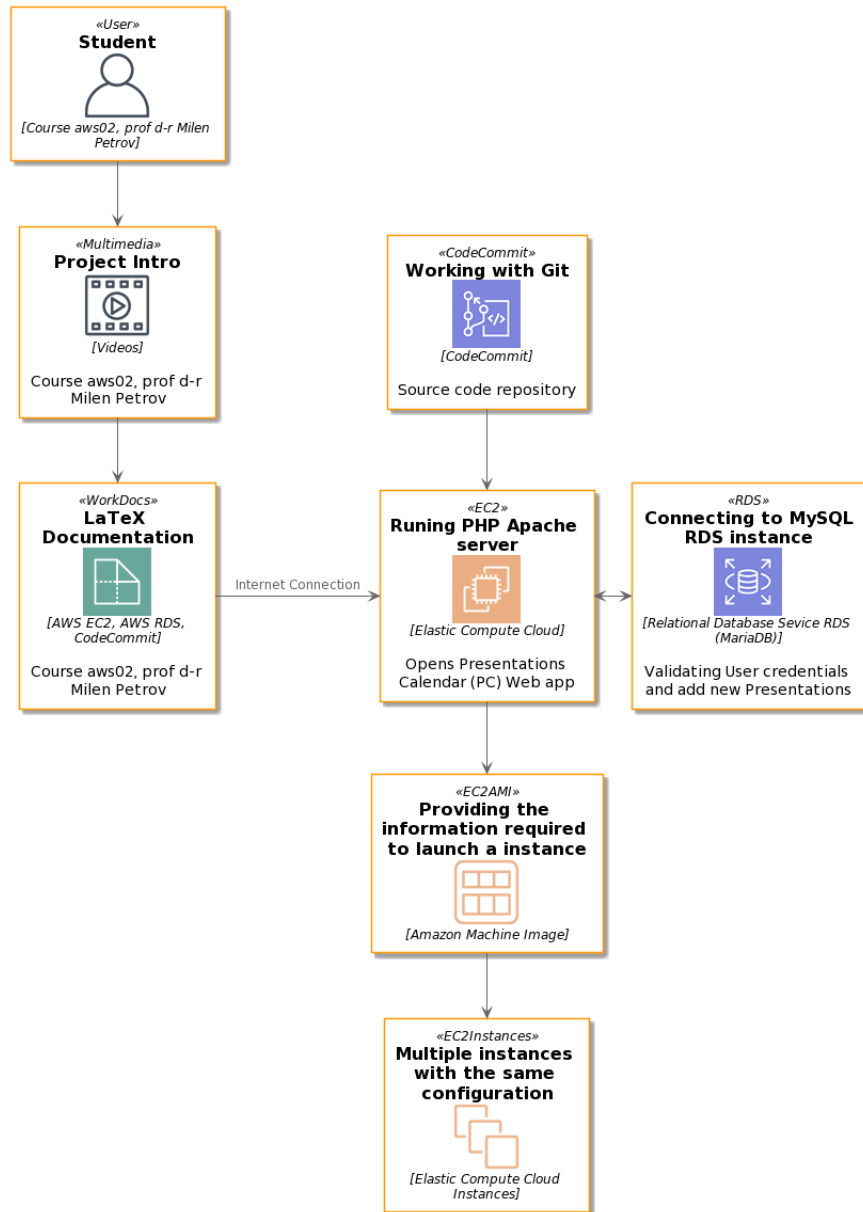
5.4 Amazon Machine Images - Backup

AWS AMI е услуга, която позволява бързо и лесно да се създаде Backup на вече създадена и конфигуриране инстанция AWS EC2. В AMI предварително са конфигурирани Virtual Machine Image (VMI), съдържащ операционна система(ОС) нужен за създаване на инстанция. AWS предоставя и готови AMI-та, но предоставя и функционалност за създаване на ново.

За създаването на ново AMI от вече създадена AWS EC2 инстанция се изпълняват следните команди от AWS конзолата:

```
AWS services ->
EC2 ->
Instances ->
Click on instance ->
Actions ->
Create Image (Create AMI) ->
AMI is ready for use
```

6 Кратко ръководство за потребителя



Фигура 9: Use Guide PlantUML Diagram

7 Примерни данни

	ID	Subject	Date	Time	StudentID
	7	HTML: Part 1 Semantic Elements	2019-12-04	11:15:00	4
	8	Work with sessions and cookies	2019-12-04	11:30:00	8
	9	CSS: layouts, box model	2020-01-15	09:30:00	7
	10	Test with Selenium web driver	2020-01-15	09:45:00	6
	12	Shadow DOM 101	2020-01-15	10:45:00	9
	13	Shadow DOM 201, CSS and Styling. Advanced ...	2020-01-16	10:30:00	10
	15	CSS: layouts, flexbox	2020-01-16	10:45:00	12
	16	HTML Imports	2020-01-16	11:00:00	13
	17	Google - Web Performance Best Practices	2020-02-11	11:30:00	16
	29	Spring (java)	2020-02-11	09:00:00	16
	30	URL rewriting	2020-02-11	10:15:00	16
	33	Multi-touch Web Development	2020-02-11	09:15:00	15
	34	MySQL: Part 1	2020-01-21	10:30:00	15
	35	Web	2020-04-20	09:00:00	5

Фигура 10: Presentations data

	ID	FullName	Identity	Email	Password
►	13	Test User 2	81004	test2@fmi.uni-sofia.bg	test2
	14	Test User	88888	test.user@gmail.com	testuser
	15	Pesho	81811	pesho.pesho@gmail.com	81811pesho
	16	Jonh Smith	90186	j.smith@gmail.com	123456

Фигура 11: Students data

8 Описание на програмния код

Кодът е разделен в няколко логически обособени части.

8.1 Презентационен слой

HTML файловете в главната папка на проекта, файловете в папките css (файлове, отговарящи за стилизацията), images (изображения, използвани в страниците).

HTML файловете са:

- index.html - основна страница на Уеб приложението.
- login.html - страница за ВХОД на потребителя.
- signup.html - страница за Регистриране на нов потребител.

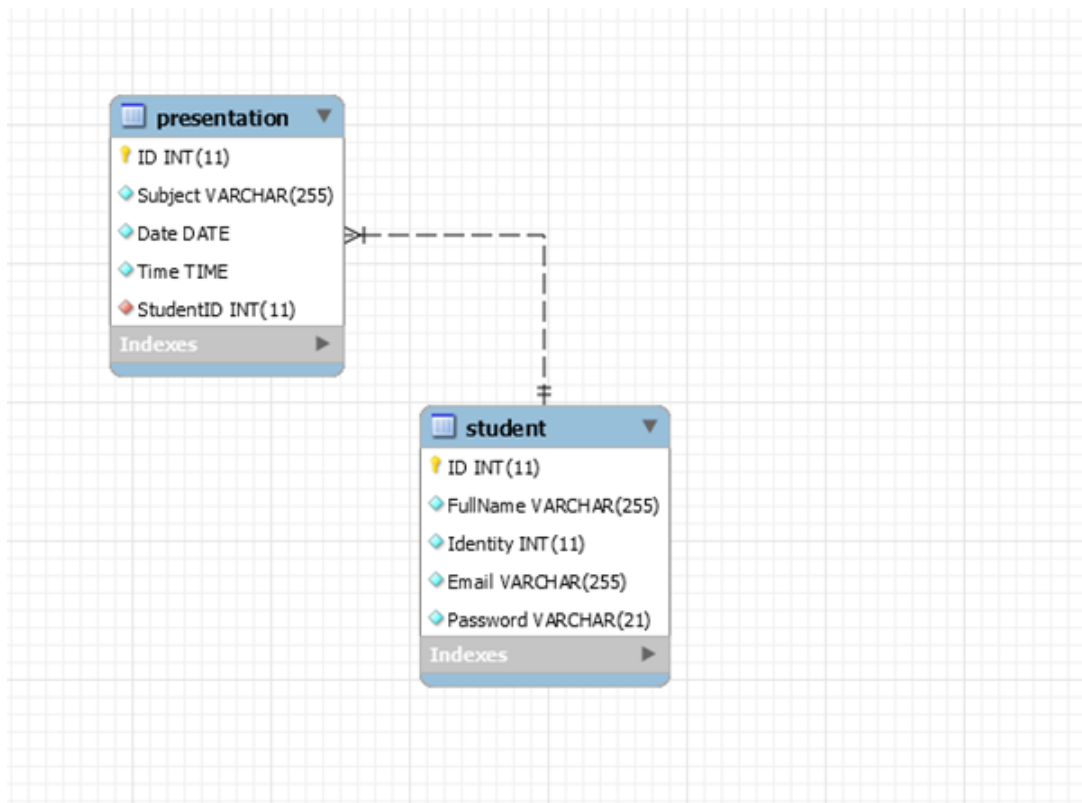
8.2 Домейн слой

RНР скриптове в подпапката `src` на главната папка на проекта:

- `addPresentation.php` - скрипт, в който се извиква метод от `db.php` за добавяне на нов запис в таблицата `calendardb.Presentations`.
- `authenticate.php` - скрипт за удостоверяване на валидността на подадени данни при ВХОД на потребителя.
- `calendar.php` - визуализация на календара.
- `day.php` - визуализация на ден.
- `db.php` - имплементация на `class DataBase`, използващ се в целия проект и даващ достъп до база данни `calendardb`, с която работи уеб приложението.
- `deletePresentation.php` - скрипт, в който се извиква метод от `db.php` за премахване на запис от таблицата `calendardb.Presentations`.
- `deletePresentationForm.php` - форма за изтриване на записана презентация.
- `generateCalendar.php` - скрипт за динамично генериране на календар.
- `generateDay.php` - скрипт за динамично генериране на ден, съдържащ записи със запазени часове за представяне на презентации.
- `home.php` - начална страница на влезнали в системата потребители.
- `logout.php` - изход от системата.
- `presentation.php` - форма за записване на презентация.
- `signup.php` - скрипт за валидация и регистриране на нов потребител по въведени данни.

8.3 Структура на релационна база данни

Архитектурата на базата данни представлява:



Фигура 12: Database - Architecture

8.4 Обща картина на Уеб приложението

При регистрацията потребителят трябва да въведе свои лични данни – собствено и фамилно име, факултетен номер, имейл и парола. Системата проверява дали всички задължителни полета са попълнени, дали имейлът е валиден и дали факултетният номер, който е въвел студентът, е уникален. Ако не съществува вече регистриран потребител с подадения факултетен номер, то регистрацията на студента е успешна и той продължава към Вход в системата.

Вход в системата се осъществява с въвеждане на факултетен номер и парола на студента. След въвеждане на нужните данни, системата проверява дали тези данни са коректни – проверява дали за съответния факултетен номер съответства подадената парола.

Ако Входът в системата е завършил с неуспех, тогава потребителят може да навигира до календара и да преглежда съдържанието, което е предоставено. Всякакви функционалности като добавяне и изтриване на запазена дата за представяне на презентация са неактивни за нерегистриран потребител. При липса на регистрация, всеки може да създаде такава, по всяко време.

Ако Регистрацията на студента е завършила с успех, то неговите данни биват записани в таблицата Student. Така по идентификационен номер в таблицата, при Вход се проверява валидността на факултетния номер и парола на студента. След безпроблемно влизане в системата, всеки потребител има възможност да навигира към календара със запазени дати за представяне на презентации, да запази своя дата или да изтрие запазена дата за презентация, само ако текущия потребител я е запазил. Изтриването на чужди запазени дати е забранено за активните потребители. Всеки студент има достъп за триене само на своите презентации.

При добавяне на нова презентация в таблицата Presentation се запазва запис за запазената дата. Така визуализирането на запазена дата за представяне на презентация става мигновено.

9 Приноси на студента, ограничения и възможности за бъдещо развитие

Представената функционалност в този документ е изработена изцяло от автора на този документ. Съществуват много възможности за бъдещо развитие като добавяне AWS CloudWatch и S3 при разширяване на функционалности на разгледания проект.

10 Какво научих

Благодарение на изпълнението на текущия проект за пръв път създадох цялостен web сайт, който да е качен в cloud пространството. При реализирането му не са използвани други framework и външни графични библиотеки. Запознах се с различни услуги, предоставени от AWS , MySql WorkBench при създаване на базата от данни, създаването на таблиците и съответно вкарването на разнообразните данни в тях. Подобрих познанията си в технологии като HTML, CSS, MySQL, PHP и Apache.

11 Списък с фигури

Списък на фигурите

1	Configuring and cloning the repository	6
2	Local folder Git -> [credential]	7
3	Local folder .aws -> [default]	7
4	Advanced Details - EC2	8
5	Security Group - Rules	8
6	Configure EC2	9
7	Project files that would be run by the Apache server	10
8	Configure Apache	10
9	Use Guide PlantUML Diagram	12
10	Presentations data	13
11	Students data	13
12	Database - Architecture	15

12 Използвани източници

- [1] AWS Elastic Compute Cloud
- [2] AWS Relational Database Service
- [3] CodeCommit
- [4] AWS FMI Course
- [5] Setup the Credential Helper on Git
- [6] AWS Documentation