

# Тема: Разширение на проект от УЕБ: PIC2MAP - сайт за позициониране на снимки по GPS координати

Предмет: AWS

Изготвил: Виолета Павлова, фн: 81872, имейл:  
violetaip@uni-sofia.bg

Лектор: доц. д-р Милен Петров, година: 2022

## Съдържание

1	Условие	3
2	Въведение	3
3	Теория	4
4	Използвани технологии	4
5	Инсталация и настройки	5
5.1	IAM . . . . .	5
5.2	RDS . . . . .	5
5.3	EC2 . . . . .	5
5.4	S3 . . . . .	6
6	Кратко ръководство за потребителя	7
7	Примерни данни	7
8	Описание на програмния код	8
9	Приноси на студента, ограничения и възможности за бъдещо развитие	8
10	Какво научих	8
11	Списък с фигури и таблици	9



# 1 Условие

Настоящото задание е създаването на приложение, подобно на <https://www.pic2map.com/tnbzb.html>, което съдържа следната функционалност:

- Регистрация в приложението
- Вход в приложението
- Възможност за преглед върху публична карта (напр. openstreet map). Картата се поставя на заден фон, когато потребителят кликне върху някоя отметка и се визуализира съответстващата ѝ картинка.
- Снимките могат да бъдат качвани като частни или публични, с възможност да се преглеждат на карта;
- Качване на снимка чрез специална за целта форма или чрез кликуване върху картата. Снимката може да се 'оразмерява' по зададен формат (insta formats). Ако качената снимка съдържа информация за местоположението на заснемане, то тези координати се взимат предвид.

# 2 Въведение

Целта на проекта е да се разработи уеб приложение за споделяне на снимки. След поредното си пътуване, човек ще може да качи любими снимки от местата, на които е бил. Много от тези снимки вече имат GPS координати, но в случай, че нямат, човек може да избере къде да постави своята снимка на картата. Ще може да избере дали иска неговата снимка да е публична, като така допринася към всеобщата колекция от снимки. Ако реши, че снимката е много лична, потребителят може да запази само за себе си, като я направи частна. Приложението дава възможност за виртуална разходка из забележителностите на всяка точка от земята. Приложението може да се ползва и от нерегистрирани потребители, като те могат само да разглеждат публичните снимки.

### 3 Теория

За разширяването на приложението се използват три AWS услуги:

**Amazon Elastic Compute Cloud (EC2):** уеб услуга, която осигурява сигурен и преоразмерим изчислителен капацитет в облака чрез използване на виртуални машини. Предоставя лесна интеграция с други AWS услуги, например AWS RDS чрез VPC (частна виртуална мрежа в облака на Amazon). Настоящото приложение върви на EC2 инстанция, като има връзка с друга RDS инстанция.

**Amazon Relational Database Service (RDS):** услуга, която позволява лесно създаване, конфигуриране и мащабиране на релационни бази данни. RDS се грижи за голяма част от административните задачи по поддържането на базата като например създаването на backup, snapshots, security updates и т.н. Приложението съхранява информация за потребителите и параметрите на изображенията в RDS инстанция, която има една база данни pin-your-pics и съответно две таблици pics и users.

**AWS Simple Storage Service (S3):** услуга, която предоставя сигурно и надеждно съхраняване на обекти в облака. Това е интерфейс, който позволява съхраняването и достъпването на голямо количество ресурси. Могат да се съхраняват най-обикновени текстови файлове или по-специални файлове, снимки, видеоматериали. В контекста на приложението е удобно съхраняването на изображенията, които са разпръснати из картата. Тези изображения се достъпват и качват в S3 bucket-а бързо и лесно.

### 4 Използвани технологии

Development:

- **Front-end:** HTML5, CSS3, JavaScript
- **Back-end:** PHP version 'PHP 8.0.18'
- **Database:** MySQL

AWS:

- RDS
- EC2
- S3

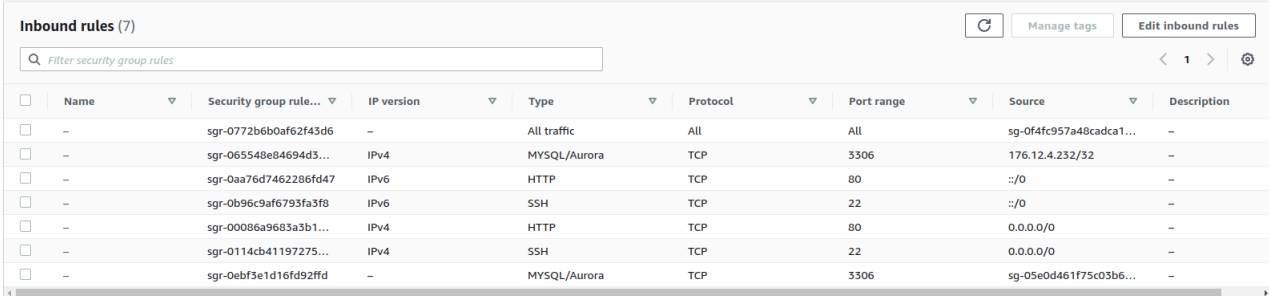
## 5 Инсталация и настройки

### 5.1 IAM

Разбира се всички стъпки изпълняваме с IAM user, на който root user-а закача съответните policies, които да го упълномощават да извършва операциите по инстанциите. Правата са следните: AmazonRDSFullAccess, AmazonEC2FullAccess, AmazonS3FullAccess.

### 5.2 RDS

В AWS console създаваме RDS инстанция с free tier, като указваме master username и password, които ще използваме за връзка с базата. Освен това по време на създаването на инстанцията създаваме security group pin-your-pics-security-group, която в последствие ще дадем и на EC2 инстанцията. След като сме създали инстанцията, в кода connect.inc.php обновяваме данните за достъп до базата, която вече е нашата RDS инстанция.



Inbound rules (7)									
<input type="text" value="Filter security group rules"/>									
<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description	
<input type="checkbox"/>	-	sgr-0772b6b0af62f43d6	-	All traffic	All	All	sg-0f4fc957a48cadca1...	-	
<input type="checkbox"/>	-	sgr-065548e84694d3...	IPv4	MySQL/Aurora	TCP	3306	176.12.4.232/32	-	
<input type="checkbox"/>	-	sgr-0aa76d7462286fd47	IPv6	HTTP	TCP	80	::/0	-	
<input type="checkbox"/>	-	sgr-0b96c9af6793fa3f8	IPv6	SSH	TCP	22	::/0	-	
<input type="checkbox"/>	-	sgr-00086a9683a3b1...	IPv4	HTTP	TCP	80	0.0.0.0/0	-	
<input type="checkbox"/>	-	sgr-0114cb41197275...	IPv4	SSH	TCP	22	0.0.0.0/0	-	
<input type="checkbox"/>	-	sgr-0ebf3e1d16fd92ffd	-	MySQL/Aurora	TCP	3306	sg-05e0d461f75c03b6...	-	

Фигура 1: Настройки на **pin-your-pics-security-group**, достъпни през VPC Security groups

### 5.3 EC2

В AWS console създаваме EC2 инстанция с free tier в pin-your-pics-security-group. Инстанцията се намира в същия VPC като RDS.

Свързваме се с виртуалната машина чрез EC2InstanceConnect или SSH.

Инсталираме php, mysql, apache:

```
> sudo yum install php8.0
```

```
> sudo yum install mysql
```

```
> sudo apt install apache2
```

Стартираме сървъра:

```
> sudo systemctl start httpd
```

Свързваме се с RDS инстанцията чрез следната команда:

```
> mysql -h <endpoint> -P <port> -u <username> -p
```

endpoint и port взимаме от RDS, default port е 3306. Username е master username, а за password въвеждаме master password. След успешно свързване с базата изпълняваме скрипта (dbcreate.sql) за създаване на database и двете таблици в нея.

Клонираме github repository-то:

```
> git clone https://github.com/samoleti/pin-your-pics.git
```

Копираме проекта:

```
> sudo cp -r pin-your-pics /var/www/html
```

Настройваме пътя за приложението за Apache във файла /etc/httpd/conf/httpd.conf, като обновяваме DocumentRoot /var/www/html/pin-your-pics/htdocs.

Необходимо ни е да инсталираме composer:

```
> php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
> php -r "if (hash_file('sha384', 'composer-setup.php') ===
```

```
'55ce33d7678c5a611085589f1f3ddf8b3c52d662cd01d4ba75c0ee0459970c2200a51f492d557530c71c15d8d8b
```

```
echo 'Installer verified'; else echo 'Installer corrupt'; unlink('composer-setup.php');
```

```
echo PHP_EOL;"php composer-setup.php
> php -r "unlink('composer-setup.php');"
```

Инсталираме Amazon SDK for PHP:

```
> composer require aws/aws-sdk-php
```

Приложението можем да достъпим чрез Public IPv4 DNS. Например настоящето приложение е достъпно на адрес:

<http://ec2-3-223-135-185.compute-1.amazonaws.com/>

## 5.4 S3

Създаваме S3 bucket инстанция, която в последствие се ползва от приложението чрез s3 client.

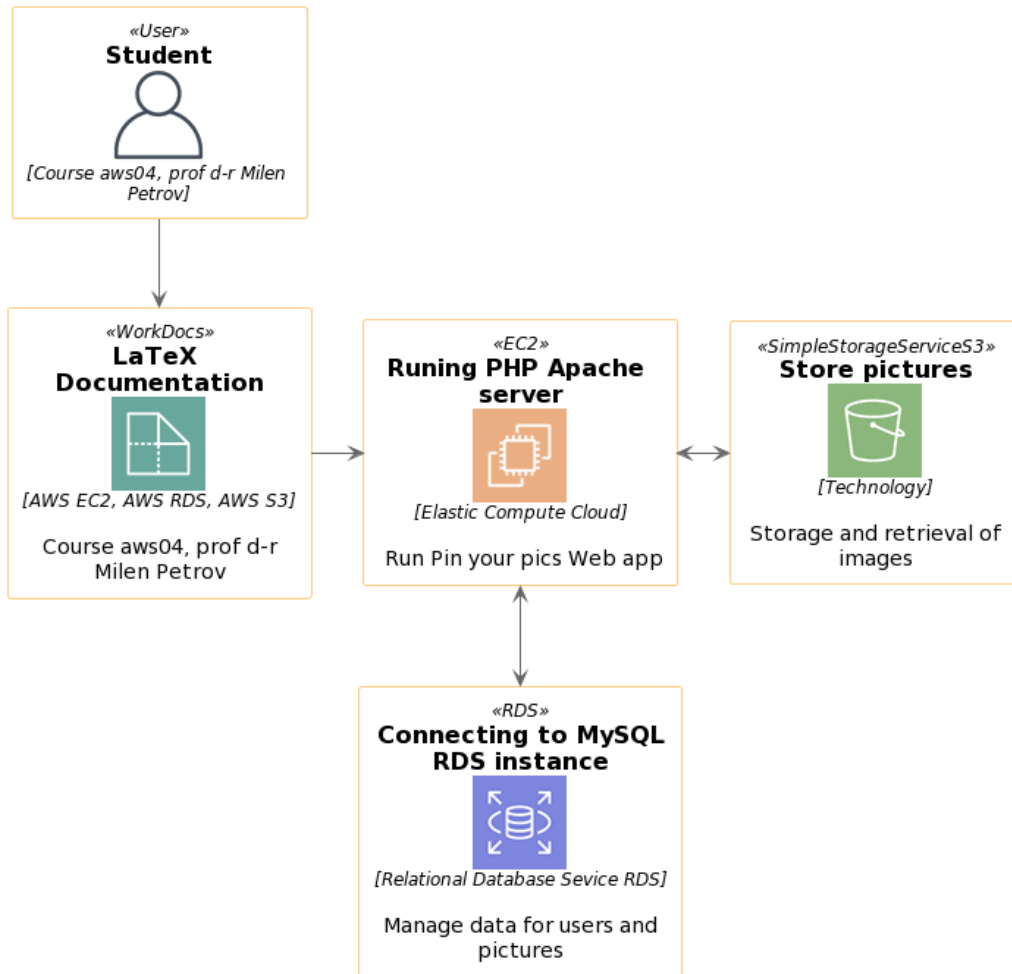
В кода във файловете upload.inc.php, index.php за bucket, IAM\_KEY, IAM\_SECRET на местата на звездичките пишем името на новосъздадения bucket и credentials на IAM user-а, чрез който сме създали bucket-а.

Другият вариант е да даваме достъп на EC2 инстанцията до S3 bucket:

<https://aws.amazon.com/premiumsupport/knowledge-center/ec2-instance-access-s3-bucket/>

По този начин избягваме писането на credentials.

## 6 Кратко ръководство за потребителя



Фигура 2: UML диаграма

## 7 Примерни данни

Потребител:

username: samoleti

email: leti1999@abv.bg

password: 1234

За снимки с GPS локация в EXIF метаданните си използвах примерните изображения от <https://github.com/samoleti/pin-your-pics/tree/master/samples>

## 8 Описание на програмния код

Сървърната част се намира в папката `pin-your-pics/php`. Помощните `php` файлове, които не са отделни страници се намират в `pin-your-pics/php/includes` папката.

Базата от данни се състои от две таблици - `users` и `pics`. Таблицата `users` съдържа информацията за регистрираните потребители. Таблицата `users` съдържа информация за GPS локацията на снимката, кой е авторът, къде на сървъра се пази снимката и дали е частна/публична.

JavaScript кодът се намира в папката `pin-your-pics/js` папката, която се изпълнява на клиента. Той отговаря за извършването на заявки към сървърната част. CSS файловете са в папката `pin-your-pics/css`, с които допълнително се стилизира уеб страницата.

## 9 Приноси на студента, ограничения и възможности за бъдещо развитие

Представената функционалност е разработена от автора на документа.

Възможност за бъдещо подобрене е добавянето на затворени групи за споделяне на снимки. Друга възможност е използването на снимка/карта вместо конкретно географска такава. Откъм услуги в облака е добра стъпка използването на AWS CodeCommit.

## 10 Какво научих

Разширяването на проекта чрез AWS услуги ме научи как да създам уеб приложение, качено в облака. За мен беше интересно да науча как да комбинирам набор от AWS услуги и как те си взаимодействат помежду си.



## 11 Списък с фигури и таблици

### Списък на фигурите

1	Настройки на <b>pin-your-pics-security-group</b> , достъпни през VPC	
	Security groups . . . . .	5
2	UML диаграма . . . . .	7

## 12 Използвани източници

- [1] FMI Moodle course
- [2] AWS Academy Introduction to Cloud: Semester 1
- [3] AWS Academy Introduction to Cloud: Semester 2
- [4] Grant EC2 instance access to S3 bucket/