Mini Project 1

Objectives

- Implement and demonstrate your understanding of the deployment of cloud-native applications

Requirements

Application

Web application will include 3 pages, (pulled from your Github account) (grads need to add some CSS and formatting)

1. index page - A welcome page that has links to the gallery and submit page
2. gallery page (reads your database and retrieves all of the before and after pictures)
    a. You will need to access your database via the AWS language SDK of your choice
    b. http://docs.aws.amazon.com/aws-sdk-php/v3/guide/getting-started/installation.html
    c. http://docs.aws.amazon.com/aws-sdk-php/v3/api/
    d. http://docs.aws.amazon.com/aws-sdk-php/v3/api/class-Aws.Rds.RdsClient.html
3. submit page (form) (upon submission of the form, images are placed in an S3 bucket, and a URL and information relating to that job is entered). The image is run through an image manipulation library and a black and white version or thumbnail is generated (your choice) programming language of your choice. The form contents are listed in the schema at the end of the document.

Infrastructure

1. Load Balancer with sticky-bits
2. AutoScaling group (desired state 3 EC2 instances) w/Launch configuration
    a. Using: `--iam-instance-profile`
    b. http://docs.aws.amazon.com/cli/latest/reference/ec2/run-instances.html
3. 1 RDS instance (your choice) (to store URLs and job status) See schema below:
4. S3 Bucket for storing before (pre-processing)
5. S3 Bucket for storing after (post-processing)

Variables

Variables needed to launch on my own account must be passed by positional parameters and details mentioned in the ReadMe.md

Deliverables

Locate all resources in a folder in your Github repo named: **mp1** (note the spaces and spelling)

1. create-env.sh
2. install-app-env.sh (installs all the needed dependencies per EC2 instance as well as clone your code)
3. destroy-env.sh script to delete all of these items

***Note for this assignment assume that your security group and key-pairs are already pre-existing

Documentation

Rubric

It has to work - 18 points just for trying

Remaining 32 points spread over the 8 items listed above with either 1 point if missing, 2 points if implemented poorly, and 4 points if implemented in a fully functional capacity.

Database Schema and Prepared Statements

```
$sql = "CREATE TABLE records
(
id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
email VARCHAR(32),
phone VARCHAR(32),
s3-raw-url VARCHAR(32),
s3-finished-url VARCHAR(32),
status INT(1),
reciept BIGINT
)";

/* Prepared statement, stage 1: prepare */
if (!($stmt = $link->prepare("INSERT INTO records (id, email, phone,
s3-raw-url, s3-finished-url, status,reciept) VALUES
(NULL,?,?,?,?,?,?)"))) {
    echo "Prepare failed: (" . $mysqli->errno . ") " . $mysqli->error;
}

$stmt->bind_param("ssssii",$email,$phone,$s3-raw-url,$s3-finsihed-
url,$status,$reciept);
```