Lab 12: Instrumentation, keep your lights on

The purpose of the lab is to demonstrate some of the concepts presented at course in a cloud environment.

A. Setup your environment using a Cloud Formation Script 0.5 points

Go to the AWS Console. Check console in region **Ireland**.

1. Go to **CloudFormation Management Console** > **Create New Stack.**
   - For **Stack template**, choose **Upload a template to Amazon S3** and use the file @ https://goo.gl/92sY1h. Click **Next.**
   - In **Parameters**, set **Stack name**, **DBPassword** (at least 8 *alphanumeric* characters), **DBUser** and **KeyName** (choose the existing "**qwikLABS-<...>**" key pair). Click **Next**.
   - In the **Advanced** section of the page, choose **New Amazon SNS topic**: give a name for **Topic**, and your email as **Email.**
   - Click **Next**. Verify the details you entered and click **Create**.

   The stack creation will take around 12 minutes. The stack consists of an Elastic Load Balancer with two EC2 instances that will be your webservers. The webserver connects to an RDS (Amazon Relational Database Service) MySQL database using PHP. The Cloud Formation Script also creates an EC2 instance that we are going to use to load test the web application. The needed Security Groups are also created.
   You can see the details in the **Events** tab while the Stack is creating. Frequently refresh the table with the stack info and wait until the status becomes: CREATE_COMPLETE. (If you get ROLLBACK_COMPLETE instead, call lab assistant).
2. To access the web page of your application, check the **Outputs** tab. We are going to use this URL to load test the system. The page is simple: it connects to the DB, and makes 10 dummy queries, to create some load on DB. In addition to this, the application is also doing some load on the web server by calculating the first 5000 prime numbers.
   Open the URL in your browser to check everything is working as expected.

B. Setup your initial Dashboard. Add alarms. 1.5 points

Very important: Think about what it makes more sense to monitor: the **Average, Sum, Maximum, or the Minimum** for each metric.

1. Add Web servers monitoring. Go to **CloudWatch Management Console > Metrics > EC2 > By Auto Scaling Group.** (If you don't see the option *By Auto Scaling Group*, wait for another 2-3 minutes and retry).
   Select the checkbox on **CPUUtilization**. Go to **Graphed Metrics** tab and change the Period (column) from 5 Minutes to 1 Minute. From here, you can also change the aggregation type (avg, max, min, etc.).
2. In the **Actions** column, Click **Create Alarm**. Alarm on CPU >= 80 for 5 consecutive periods. **Actions**: when **State is Alarm**, send notification to: select the topic you created when you set up the Stack. Click **Create Alarm**.
3. In the **Actions** drop-down, Click **Add to dashboard**, give a name for your dashboard, click on **Add to Dashboard**. Don't forget to click **Save dashboard**.
4. Go to **Metrics**, in the **All metrics** tab.
5. Add database monitoring to your dashboard. Go to **RDS** (these are the database metrics), select **By Database Engine**. Here are some suggested metrics to add to the dashboard: **CPUUtilization**, **DatabaseConnections**. You can also add more alarms, as you consider fit.
6. Add ELB monitoring. This will be used to see the health of the machines behind it, the status codes of the responses, the latency, request count. Select the ones that make more sense to you, and add them to the dashboard.

C. Start the load test 0.5 points

1. Go to **EC2 Management Console > Instances**. Connect to the **t2.medium** instance.[1] (Check foot-note if you don't remember how). This instance has apache benchmark already installed. You can use this to load test your application.
2. Start your load test. Use the URL of the application you got after you created the Stack. Use **ab -h** to see the options to Apache Behchmark. Here is a simple command that makes 10k requests, with a concurrency of 10.

   $ ab -n 10000 -c 10 <URL_of_the_application>

   (Recommendation: write the command by hand, as copy-pasting it into the terminal might not work due to the '-' character.)

[1] If you don't remember how to SSH on the EC2 hosts, check:
- using the **ssh** client: **steps 2. -> 7.** @ [Connecting to Your Linux Instance Using SSH > To connect to your instance using SSH]
- or using Putty: **steps 2. -> 7.** @ [Connecting to Your Linux Instance from Windows Using PuTTY > To start a PuTTY session]

D. Understand the limits of the system 1 point

1. Go to the dashboard you created. **CloudWatch Management Console > Dashboards > <Your Dashboard>**. Understand the correlation from the number of requests from your load test and the metrics you see in your dashboard.
2. Do you need more metrics to understand what's happening? Go to the **Metrics** section on the left and add them to the dashboard.
3. Increase the number of concurrent requests done by **ab.** You can also run **ab** from 2 or more terminals.

E. Game Day – simulate failures in your system 0.5 points

1. Use the **EC2 Management Console** (Instances, Security Groups, Load Balancers, etc.) to simulate failures in your system.
2. Do you have enough metrics on your dashboard to observe the failure you introduced? Add it if it doesn't exist. Add more alarms. Is your system resilient to failures?