Lomba Kompetensi Siswa
Sekolah Menengah Kejuruan
Tingkat Provinsi Jawa Barat
Tahun 2021

Modul 2 – Portal - SMK Cloud Provider

August 4, 2021

Bidang Lomba Cloud Computing

1. Overview

This project aims to deploy and scale a Portal - SMK Cloud Provider application with Laravel as our technology suite. As we know, many requests come to our website and the traffic becomes very high! At first, everything was normal and our infrastructure had no problems with this. Until we realized our Portal - SMK Cloud Provider application became popular and we were unable to handle traffic requests from multiple clients. Our resources are high on both CPU and memory, so the servers we use crash. So, we needed a scalable cloud infrastructure that would take care of this problem and no longer worry about incoming requests or traffic. Amazon Web Services comes to the rescue. You must build and deploy our web applications into the AWS infrastructure. Before performing tasks in this project, please carefully read the technical details below.

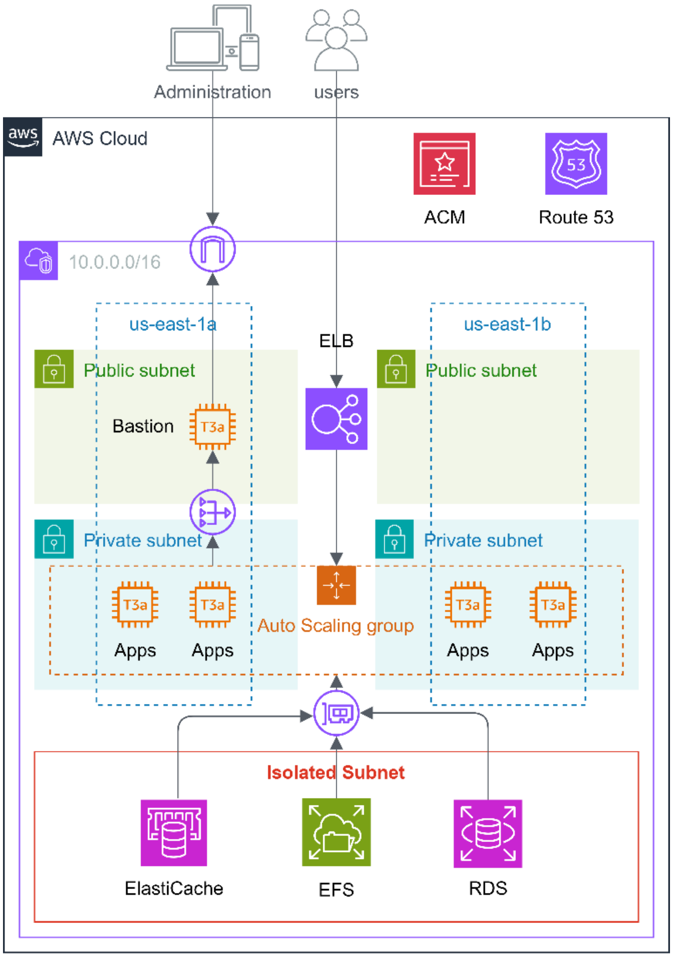
2. General Rules

1. Failure to comply with the rules will result in immediate disqualification.
2. You have 6 hours to finish the tasks.
3. This is an open book test.
4. You may use AWS Console and AWS CLI to deploy the solutions. You may not use CloudFormation or CDK.
5. Between and after the event, you may not access your account. Any activity on AWS during this period is not allowed.
6. During the event, multiple login is not permitted.
7. If you have any question, do not hesitate to ask..

3. External Resource

1. The repository URL for the project is https://github.com/lksjabar2021/modul-2
2. This solution must be deployed in **us-east-1 (N. Virginia)** region. Deploying in another region will result in a major point reduction.

4 Architecture



5 Task

1. Create VPC with the following specifications:
   - IPv4 CIDRs: 10.0.0.0/16
   - Number of NAT Gateways: 1

     a) Name: modul2, Subnet: public1
   - Subnets:

     a) - Subnet Name: public1
     - IPv4 CIDR block: 10.0.0.0/24
     - 0.0.0.0/0 is routed to: Internet Gateway (modul2)

     b) - Subnet Name: public2
     - IPv4 CIDR block: 10.0.1.0/24
     - 0.0.0.0/0 is routed to: Internet Gateway (modul2)

     c) - Subnet Name: private1
     - IPv4 CIDR block: 10.0.2.0/24
     - 0.0.0.0/0 is routed to: NAT Gateway (modul2)

     d) - Subnet Name: private2
     - IPv4 CIDR block: 10.0.3.0/24
     - 0.0.0.0/0 is routed to: NAT Gateway (modul2)

    e) - Subnet Name: private3
       - IPv4 CIDR block: 10.0.4.0/24
    f) - Subnet Name: private4
       - IPv4 CIDR block: 10.0.5.0/24

2. Create an EC2 key pair.
   - Name: modul2
   - Key pair type: RSA
   - Private key file format: .pem
3. Create a bastion host (EC2 instance) to access resources in private subnet remotely.
   - Name: modul2-Bastion
   - Subnet: public1
   - Instance Type: **t3.small**
   - Public IP Address: Enabled
   - Security Group Rule: Allow SSH traffic from anywhere
   - Key pair: modul2
4. Create a database using Amazon MariaDB RDS with the following specification:
   - Select MariaDB Compatibility
   - Select MariaDB version 10.4
   - Specify the database cluster name: **modul2**
   - Specify the database name: **modul2**
   - For the master username: **modul2**
   - Specify your master password
   - DB instance class: **db.t3.small**
   - Network Subnet: private3 and private4
5. Create ElastiCache with the following specifications:
   - Redis Cluster
   - Enable cluster mode
   - Enable multi availability zone
   - Instance Type: **cache.t3.small**
   - Port: 6379
   - Number of shards: 1
   - Number of replicas: 1
   - Network Subnet: private3 and private4
6. Create an EFS file system with the following specifications:
   - Storage class: Standard
   - Transition into IA: 7 days since last access
   - Transition out of IA: On first access
   - Throughput mode: Bursting
   - Encryption: Enable encryption of data at rest
   - Mount targets:
     a) AZ: us-east-1a, Subnet: private3
     b) AZ: us-east-1b, Subnet: private4

7. Create an EC2 instance
    - Name: modul2
    - Key pair: modul2
    - Instance Type: t3.small
    - Storage: 8 GIB of GP2
    - Connect to the instance and finish the following instructions inside the instance:
        a) Install Apache2 for create Web Server Laravel from EC2
        a) PHP version 8.0
        b) And all PHP modules that are required by Laravel
    - You must configure the EFS and clone the source code repository from section 3:
        a) Mount EFS on your EC2 instances in a private subnet, the directory for mounting this EFS is /var/www/efs/modul2
        b) Clone the Laravel source code on mounted EFS storage (/var/www/efs/modul2)
        c) Install Composer
        d) Generate a new key: **php artisan key:generate**
        e) Database migrate: **php artisan migrate --force**
8. Create an EC2 Launch Template from Instance modul1 with the following specifications:
    - Launch template name: modul2
    - Key pair: modul2
    - Instance Type: t3.small
9. Create Auto Scaling Group (ASG) with the following specifications:
    - Network Subnet: private1 and private2
    - Load balancing: Attach to a new load balancer
        a) Load balancer name: modul2
        b) Load balancer type: Application Load Balancer
        c) Load balancer scheme: Internet-facing
    - Minimum Capacity: 2
    - Desired Capacity: 2
    - Max Capacity: 4
    - Scaling policies: Target tracking scaling policy
        a) Metric type: Average CPU utilization
        b) Target value: 75
10. Configure modul2 to redirect HTTP request to HTTPS and forward HTTPS request to the ASG’s target group.
11. Create a certificate in ACM
    - Domain Name: modul2.[YOUR DOMAIN]
    - Validation Method: DNS validation
12. Add custom domain to the Application Load Balancer
13. Open http://modul2.[YOUR DOMAIN] on your browser to check if HTTP request is redirected to HTTPS and make sure the web works correctly.

6 References

- Application Load Balancer documentation
- Certificate Manager documentation
- EC2 documentation
- EC2 Auto Scaling documentation
- Route 53 documentation
- VPC documentation
- Apache documentation
- Amazon Virtual Private Cloud
- Amazon Elastic Compute Cloud
- Amazon Elastic File System
- Laravel
- Composer
- PHP

Good luck!