



# **LOMBA KOMPETENSI SISWA (LKS) SEKOLAH MENENGAH KEJURUAN TINGKAT PROVINSI BANTEN TAHUN 2025**

## **NASKAH SOAL**

**Bidang Lomba  
Cloud Computing**

**Modul A – INFRASTRUCTURE SERVICES**



**MERDEKA  
BELAJAR**

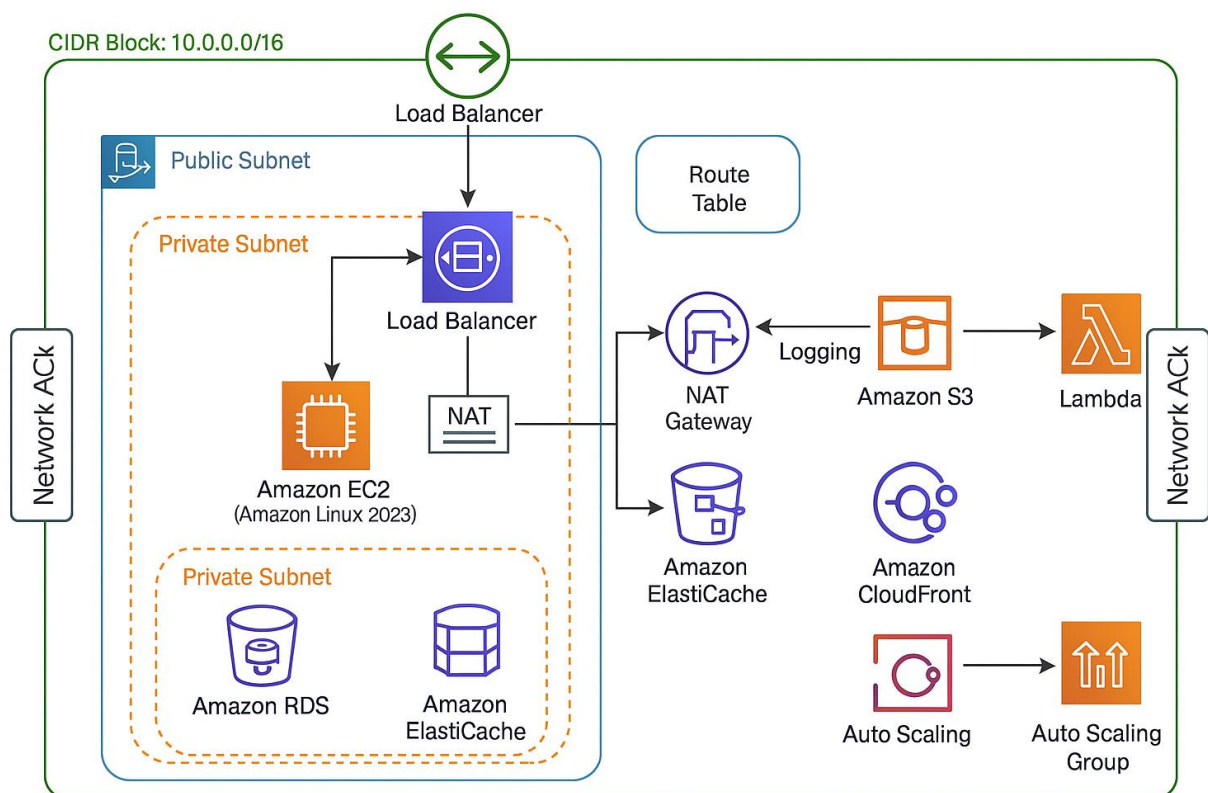
## 1. Deskripsi Umum

PT. Banten Maju Digital adalah perusahaan yang ingin memigrasi layanan e-commerce mereka ke cloud AWS agar dapat menangani lonjakan trafik dengan aman dan efisien. Anda sebagai peserta diminta membuat infrastruktur layanan secara lengkap, aman, dan siap produksi.

## 2. Aturan Umum

- 2.1. Ketidakpatuhan terhadap aturan akan menyebabkan diskualifikasi langsung.
- 2.2. Anda memiliki waktu 4 jam untuk menyelesaikan tugas.
- 2.3. Anda dapat menggunakan AWS Console dan AWS CLI. Tidak diperbolehkan menggunakan SAM, CloudFormation, CDK, atau Terraform.
- 2.4. Setelah acara dimulai dan selesai, Anda tidak diperbolehkan mengakses akun AWS Anda.
- 2.5. Login ganda selama acara tidak diperbolehkan.

## 3. Architecture



- 3.1. Aplikasi akan disebar ke infrastruktur cloud AWS dengan VPC sebagai tumpukan jaringan.
- 3.2. Anda perlu mengatur NAT Gateway untuk akses internet dari instance privat.
- 3.3. Gunakan Amazon EC2 untuk menyebarkan aplikasi Node.js.
- 3.4. Gunakan region ap-southeast-1 (Singapore) dalam seluruh pengerjaan.
- 3.5. Untuk Nama-nama resource gunakan format unik misalnya LKS-CC-25-(Nama\_Peserta)
- 3.6. Gunakan Elastic Load Balancer untuk distribusi lalu lintas.
- 3.7. Gunakan Auto Scaling Group dengan kebijakan skalabilitas berbasis CPU usage.
- 3.8. Install dependensi aplikasi menggunakan user-data.
- 3.9. Gunakan Amazon SNS untuk notifikasi via email.

#### **4. Tugas**

- 4.1. Buat 1 VPC dengan CIDR 10.0.0.0/16, serta 2 subnet (publik dan privat). Sertakan konfigurasi NACL.
- 4.2. Tambahkan Internet Gateway dan NAT Gateway. Atur routing untuk subnet.
- 4.3. Deploy EC2 (Amazon Linux 2023) di subnet private, dan akses aplikasinya melalui Load Balancer yang berada di subnet public, install Node.js via user-data. Buka port 22 dan 80.
- 4.4. Jalankan aplikasi Node.js sederhana (app.js) dan akses via browser.
- 4.5. Buat Load Balancer dan Auto Scaling Group (min 1, max 3) berdasarkan CPU usage > 60%.
- 4.6. Buat bucket S3 berisi index.html dan tampilkan via CloudFront. Aktifkan logging.
- 4.7. Buat Lambda Function untuk mengirim notifikasi SNS saat Auto Scaling menambah instance.
- 4.8. Buat SQS dan Lambda untuk membaca pesan dan log ke CloudWatch.
- 4.9. Gunakan IAM Role dengan custom policy (least privilege) untuk EC2 dan Lambda.
- 4.10. Enkripsi bucket S3 dengan KMS key buatan sendiri.
- 4.11. Buat custom CloudWatch Dashboard untuk monitoring.
- 4.12. Deploy aplikasi alternatif via Elastic Beanstalk dengan konfigurasi custom.

## Contoh Script Program:

- **Node.js App (app.js):**

```
const http = require('http');
const PORT = 80;
http.createServer((req, res) => {
  res.writeHead(200, { 'Content-Type': 'text/plain' });
  res.end('Halo dari Web Server EC2!');
}).listen(PORT, () => console.log(`Server running on port ${PORT}`));
```

- **User Data EC2:**

```
#!/bin/bash
yum update -y
curl -sL https://rpm.nodesource.com/setup_18.x | bash -
yum install -y nodejs
cat <<EOF > app.js
const http = require('http');
const PORT = 80;
http.createServer((req, res) => {
  res.writeHead(200, { 'Content-Type': 'text/plain' });
  res.end('Halo dari EC2 instance!');
}).listen(PORT);
EOF
node app.js &
```

- **Lambda Function (SNS Notify):**

```
import json
import boto3
sns = boto3.client('sns')

def lambda_handler(event, context):
    sns.publish(
        TopicArn='arn:aws:sns:region:account-id:NotifyAutoScale',
        Message='Instance baru telah ditambahkan oleh Auto Scaling Group.',
        Subject='Pemberitahuan Auto Scaling'
    )
    return {'statusCode': 200, 'body': json.dumps('Notifikasi terkirim')}
```

- **Lambda + SQS Reader:**

```
import json

def lambda_handler(event, context):
    for record in event['Records']:
        print(f"Pesan diterima: {record['body']}")
```

## **5. Contoh Output Simulasi**

- 5.1. EC2 menampilkan teks 'Halo dari Web Server EC2!' di browser.
- 5.2. Bucket S3 menampilkan index.html via CloudFront.
- 5.3. Auto Scaling menambah instance saat CPU > 60% dan notifikasi dikirim via SNS.
- 5.4. Lambda mencetak pesan dari SQS ke CloudWatch Logs.
- 5.5. Aplikasi via Elastic Beanstalk dapat diakses publik.