

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

**NASKAH SOAL**

**Bidang Lomba**

**Cloud Computing**

**Modul A – INFRASTRUCTURE SERVICES**





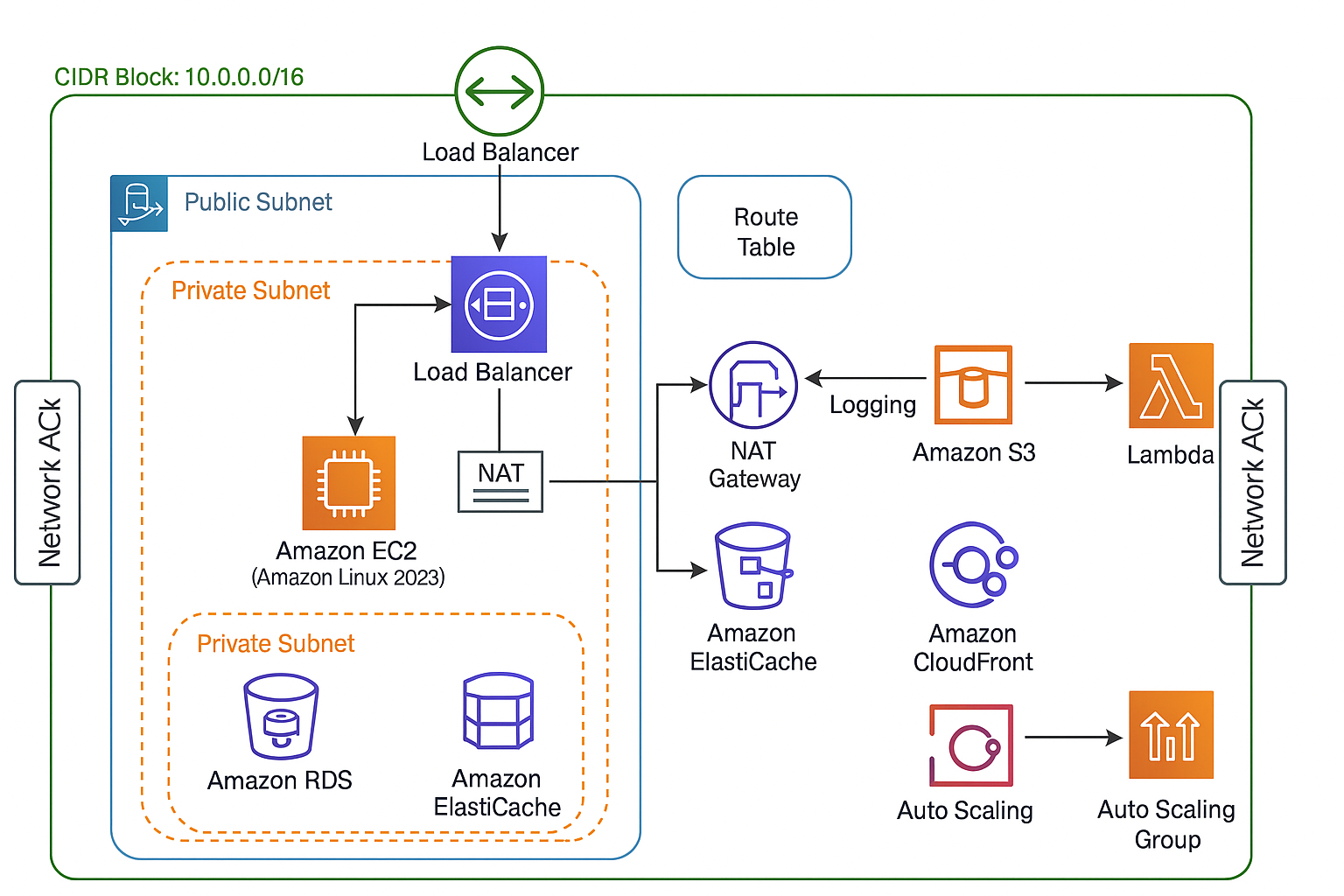
1. **Deskripsi Umum**

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.

### ****Aturan Umum****

* 1. Ketidakpatuhan terhadap aturan akan menyebabkan diskualifikasi langsung.
  2. Anda memiliki waktu 4 jam untuk menyelesaikan tugas.
  3. Anda dapat menggunakan AWS Console dan AWS CLI. Tidak diperbolehkan menggunakan SAM, CloudFormation, CDK, atau Terraform.
  4. Setelah acara dimulai dan selesai, Anda tidak diperbolehkan mengakses akun AWS Anda.
  5. Login ganda selama acara tidak diperbolehkan.

1. **Architecture**



* 1. Aplikasi akan disebarkan ke infrastruktur cloud AWS dengan VPC sebagai tumpukan jaringan.
  2. Anda perlu mengatur NAT Gateway untuk akses internet dari instance privat.
  3. Gunakan Amazon EC2 untuk menyebarkan aplikasi Node.js.
  4. Gunakan Elastic Load Balancer untuk distribusi lalu lintas.
  5. Gunakan Auto Scaling Group dengan kebijakan skalabilitas berbasis CPU usage.
  6. Install dependensi aplikasi menggunakan user-data.
  7. Gunakan Amazon SNS untuk notifikasi via email.

## **Tugas**

* 1. Buat 1 VPC dengan CIDR 10.0.0.0/16, serta 2 subnet (publik dan privat). Sertakan konfigurasi NACL.
  2. Tambahkan Internet Gateway dan NAT Gateway. Atur routing untuk subnet.
  3. Deploy EC2 (Amazon Linux 2023) di subnet publik dan install Node.js via user-data. Buka port 22 dan 80.
  4. Jalankan aplikasi Node.js sederhana (app.js) dan akses via browser.
  5. Buat Load Balancer dan Auto Scaling Group (min 1, max 3) berdasarkan CPU usage > 60%.
  6. Buat bucket S3 berisi index.html dan tampilkan via CloudFront. Aktifkan logging.
  7. Buat Lambda Function untuk mengirim notifikasi SNS saat Auto Scaling menambah instance.
  8. Buat SQS dan Lambda untuk membaca pesan dan log ke CloudWatch.
  9. Gunakan IAM Role dengan custom policy (least privilege) untuk EC2 dan Lambda.
  10. Enkripsi bucket S3 dengan KMS key buatan sendiri.
  11. Buat custom CloudWatch Dashboard untuk monitoring.
  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']}")

# **Contoh Output Simulasi**

* 1. EC2 menampilkan teks 'Halo dari Web Server EC2!' di browser.
  2. Bucket S3 menampilkan index.html via CloudFront.
  3. Auto Scaling menambah instance saat CPU > 60% dan notifikasi dikirim via SNS.
  4. Lambda mencetak pesan dari SQS ke CloudWatch Logs.
  5. Aplikasi via Elastic Beanstalk dapat diakses publik.