```
AWSTemplateFormatVersion: '2010-09-09'
Description: Deploy WordPress with Auto Scaling, Scheduled Shutdown, and Notifications in us-east-1

Parameters:
 KeyName:
  Type: AWS::EC2::KeyPair::KeyName
  Description: Name of an existing EC2 KeyPair to enable SSH access

Resources:
 # VPC
 VPC:
  Type: AWS::EC2::VPC
  Properties:
   CidrBlock: 10.0.0.0/16
   EnableDnsSupport: true
   EnableDnsHostnames: true
   Tags:
    - Key: Name
      Value: WordPressVPC

 # Internet Gateway
 InternetGateway:
  Type: AWS::EC2::InternetGateway
  Properties:
   Tags:
    - Key: Name
      Value: WordPressIGW

 AttachGateway:
  Type: AWS::EC2::VPCGatewayAttachment
  Properties:
   VpcId: !Ref VPC
   InternetGatewayId: !Ref InternetGateway

 # Public Subnet
 PublicSubnet:
  Type: AWS::EC2::Subnet
  Properties:
   VpcId: !Ref VPC
   CidrBlock: 10.0.1.0/24
   MapPublicIpOnLaunch: true
   AvailabilityZone: us-east-1a
   Tags:
    - Key: Name
      Value: WordPressPublicSubnet

 # Route Table
 RouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
   VpcId: !Ref VPC
   Tags:
    - Key: Name
      Value: WordPressRouteTable

 Route:
  Type: AWS::EC2::Route
```

```
  DependsOn: AttachGateway
  Properties:
    RouteTableId: !Ref RouteTable
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref InternetGateway

SubnetRouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    SubnetId: !Ref PublicSubnet
    RouteTableId: !Ref RouteTable

# Security Group
InstanceSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Enable SSH and HTTP access
    VpcId: !Ref VPC
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: 0.0.0.0/0
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
    Tags:
      - Key: Name
        Value: WordPressSG

# Launch Template
LaunchTemplate:
  Type: AWS::EC2::LaunchTemplate
  Properties:
    LaunchTemplateName: WordPressLaunchTemplate
    LaunchTemplateData:
      ImageId: ami-0c02fb55956c7d316  # Amazon Linux 2 AMI in us-east-1
      InstanceType: t2.micro
      KeyName: !Ref KeyName
      SecurityGroupIds:
        - !Ref InstanceSecurityGroup
      UserData:
        Fn::Base64: |
          #!/bin/bash
          yum update -y
          amazon-linux-extras install -y php7.4
          yum install -y httpd php php-mysqlnd mysql
          systemctl enable httpd
          systemctl start httpd
          cd /var/www/html
          wget https://wordpress.org/latest.tar.gz
          tar -xzf latest.tar.gz
          cp -r wordpress/* .
          rm -rf wordpress latest.tar.gz
          chown -R apache:apache /var/www/html
          chmod -R 755 /var/www/html
```

```
      TagSpecifications:
       - ResourceType: instance
         Tags:
          - Key: AutoStop
            Value: true

# Auto Scaling Group
AutoScalingGroup:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    VPCZoneIdentifier:
     - !Ref PublicSubnet
    LaunchTemplate:
     LaunchTemplateId: !Ref LaunchTemplate
     Version: !GetAtt LaunchTemplate.LatestVersionNumber
    MinSize: '1'
    MaxSize: '2'
    DesiredCapacity: '1'
    Tags:
     - Key: Name
       Value: WordPressInstance
       PropagateAtLaunch: true

# IAM Role for Lambda
LambdaExecutionRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
     Version: '2012-10-17'
     Statement:
       - Effect: Allow
         Principal:
          Service: lambda.amazonaws.com
         Action: sts:AssumeRole
    Path: /
    Policies:
     - PolicyName: LambdaEC2StopPolicy
       PolicyDocument:
         Version: '2012-10-17'
         Statement:
          - Effect: Allow
            Action:
              - logs:CreateLogGroup
              - logs:CreateLogStream
              - logs:PutLogEvents
            Resource: arn:aws:logs:*:*:*
          - Effect: Allow
            Action:
              - ec2:DescribeInstances
              - ec2:StopInstances
            Resource: "*"

# Lambda Function to Stop Instances
StopInstanceLambda:
  Type: AWS::Lambda::Function
  Properties:
    FunctionName: StopDevInstances
```

```
    Handler: index.handler
    Role: !GetAtt LambdaExecutionRole.Arn
    Runtime: python3.9
    Timeout: 60
    Code:
      ZipFile: |
        import boto3

        def handler(event, context):
            ec2 = boto3.client('ec2')
            filters = [
                {'Name': 'tag:AutoStop', 'Values': ['true']},
                {'Name': 'instance-state-name', 'Values': ['running']}
            ]
            instances = ec2.describe_instances(Filters=filters)
            instance_ids = []
            for reservation in instances['Reservations']:
                for instance in reservation['Instances']:
                    instance_ids.append(instance['InstanceId'])
            if instance_ids:
                ec2.stop_instances(InstanceIds=instance_ids)
                print(f"Stopped instances: {instance_ids}")
            else:
                print("No instances to stop.")

  # CloudWatch Event Rule to Trigger Lambda at 6 PM UTC (1:30 AM IST)
  LambdaScheduleRule:
    Type: AWS::Events::Rule
    Properties:
      ScheduleExpression: cron(0 18 ? * MON-FRI *)  # 6 PM UTC
      State: ENABLED
      Targets:
        - Arn: !GetAtt StopInstanceLambda.Arn
          Id: StopInstanceLambdaTarget

  # Permission for CloudWatch to Invoke Lambda
  PermissionForEventsToInvokeLambda:
    Type: AWS::Lambda::Permission
    Properties:
      FunctionName: !Ref StopInstanceLambda
      Action: lambda:InvokeFunction
      Principal: events.amazonaws.com
      SourceArn: !GetAtt LambdaScheduleRule.Arn

Outputs:
  WebsiteURL:
    Description: WordPress Website URL
    Value: !Sub http://${AutoScalingGroup}
```