

AWSTemplateFormatVersion: "2010-09-09"
Description: WordPress Prod and Dev/Test Stack on Amazon Linux 2023
with Monitoring

Parameters:

KeyPairName:
 Type: AWS::EC2::KeyPair::KeyName
 Description: Select an existing KeyPair

VpcId:
 Type: AWS::EC2::VPC::Id
 Description: Select your VPC (Default VPC recommended)

SubnetId:
 Type: AWS::EC2::Subnet::Id
 Description: Select a PUBLIC subnet (Auto-assign Public IP
enabled)

LatestAmiId:
 Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
 Default: /aws/service/ami-amazon-linux-latest/al2023-ami-kernel-
default-x86_64
 Description: Latest Amazon Linux 2023 AMI

Resources:

Security Group
#####

WordPressSecurityGroup:
 Type: AWS::EC2::SecurityGroup
 Properties:
 GroupDescription: Allow HTTP and SSH access
 VpcId: !Ref VpcId
 SecurityGroupIngress:
 - IpProtocol: tcp
 FromPort: 22
 ToPort: 22
 CidrIp: 0.0.0.0/0 # Restrict to your IP for security
 - IpProtocol: tcp
 FromPort: 80
 ToPort: 80
 CidrIp: 0.0.0.0/0
 SecurityGroupEgress:
 - IpProtocol: -1
 CidrIp: 0.0.0.0/0

#####

```
# IAM Role for CloudWatch Monitoring
#####
```

```
CloudWatchRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: "2012-10-17"
      Statement:
        - Effect: Allow
          Principal:
            Service: ec2.amazonaws.com
          Action: sts:AssumeRole
    ManagedPolicyArns:
      - arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
```

```
CloudWatchInstanceProfile:
  Type: AWS::IAM::InstanceProfile
  Properties:
    Roles:
      - !Ref CloudWatchRole
```

```
#####
# Production WordPress Instance
#####
```

```
ProdWordPressInstance:
  Type: AWS::EC2::Instance
  Properties:
    InstanceType: t2.micro
    KeyName: !Ref KeyPairName
    ImageId: !Ref LatestAmiId
    IamInstanceProfile: !Ref CloudWatchInstanceProfile
    NetworkInterfaces:
      - AssociatePublicIpAddress: true
        DeviceIndex: 0
        SubnetId: !Ref SubnetId
        GroupSet:
          - !Ref WordPressSecurityGroup
```

```
UserData:
  Fn::Base64: !Sub |
    #!/bin/bash
    dnf update -y
    dnf install -y httpd wget php-fpm php-mysql php-json php
php-devel php-mysqlnd mariadb105-server mariadb105

    systemctl enable httpd
    systemctl start httpd
```

```

systemctl enable mariadb
systemctl start mariadb

mysql -e "CREATE DATABASE wordpress;"
mysql -e "CREATE USER 'wpuser'@'localhost' IDENTIFIED BY
'wppassword';"
mysql -e "GRANT ALL PRIVILEGES ON wordpress.* TO
'wpuser'@'localhost'; FLUSH PRIVILEGES;"

```

```

cd /var/www/html
wget https://wordpress.org/latest.tar.gz
tar -xzf latest.tar.gz
chown -R apache:apache wordpress
chmod -R 755 wordpress

```

```

#####
# Dev/Test WordPress Instance
#####

```

DevWordPressInstance:

Type: AWS::EC2::Instance

Properties:

InstanceType: t2.micro

KeyName: !Ref KeyPairName

ImageId: !Ref LatestAmiId

IamInstanceProfile: !Ref CloudWatchInstanceProfile

NetworkInterfaces:

- AssociatePublicIpAddress: true

DeviceIndex: 0

SubnetId: !Ref SubnetId

GroupSet:

- !Ref WordPressSecurityGroup

UserData:

Fn::Base64: !Sub |

#!/bin/bash

dnf update -y

dnf install -y httpd wget php-fpm php-mysqli php-json php

php-devel php-mysqld mariadb105-server mariadb105

systemctl enable httpd

systemctl start httpd

systemctl enable mariadb

systemctl start mariadb

mysql -e "CREATE DATABASE wordpress_dev;"

```

mysql -e "CREATE USER 'wpuser'@'localhost' IDENTIFIED BY
'wppassword';"

```

mysql -e "GRANT ALL PRIVILEGES ON wordpress_dev.* TO

```
'wpuser'@'localhost'; FLUSH PRIVILEGES;"
```

```
cd /var/www/html
wget https://wordpress.org/latest.tar.gz
tar -xzf latest.tar.gz
chown -R apache:apache wordpress
chmod -R 755 wordpress
```

```
#####
# Outputs
#####
```

Outputs:

```
ProductionURL:
  Description: Production WordPress URL
  Value: !Sub "http://${ProdWordPressInstance.PublicDnsName}/
wordpress"
```

```
DevURL:
  Description: Dev/Test WordPress URL
  Value: !Sub "http://${DevWordPressInstance.PublicDnsName}/
wordpress"
```