**Cloud Deployment Scripts and Configuration**

**1. Launching a Rocky Linux EC2 Instance**

**Step 1: Access AWS and Launch Instance**

aws ec2 run-instances --image-id ami-xxxxxxxxxxxxxxxxx --count 1 --instance-type t3.micro --key-name your-key --security-group-ids sg-xxxxxxxx --subnet-id subnet-xxxxxxxx

**Step 2: Configure Security Groups**

aws ec2 authorize-security-group-ingress --group-id sg-xxxxxxxx --protocol tcp --port 22 --cidr 203.0.113.0/32

aws ec2 authorize-security-group-ingress --group-id sg-xxxxxxxx --protocol tcp --port 80 --cidr 0.0.0.0/0

aws ec2 authorize-security-group-ingress --group-id sg-xxxxxxxx --protocol tcp --port 443 --cidr 0.0.0.0/0

**Step 3: Connect to Instance**

ssh -i your-key.pem ec2-user@<public-ip-address>

**2. Web Server Setup**

**Apache Installation**

sudo dnf install httpd -y

sudo systemctl start httpd

sudo systemctl enable httpd

**Nginx Installation (Alternative)**

sudo dnf install nginx -y

sudo systemctl start nginx

sudo systemctl enable nginx

**Firewall Configuration**

sudo firewall-cmd --permanent --add-service=http

sudo firewall-cmd --permanent --add-service=https

sudo firewall-cmd --reload

**3. Database Setup**

**Install MariaDB**

sudo dnf install mariadb-server -y

sudo systemctl start mariadb

sudo systemctl enable mariadb

**Secure MariaDB**

sudo mysql\_secure\_installation

**Create Database and User**

CREATE DATABASE webapp\_db;

CREATE USER 'webuser'@'localhost' IDENTIFIED BY 'securepassword';

GRANT ALL PRIVILEGES ON webapp\_db.\* TO 'webuser'@'localhost';

FLUSH PRIVILEGES;

EXIT;

**4. Storage Management**

**Verify Attached Disks**

lsblk

**Format New Volume**

sudo mkfs.ext4 /dev/xvdb

**Mount Volume**

sudo mkdir /mnt/data

sudo mount /dev/xvdb /mnt/data

**Persist Mount on Reboot**

sudo blkid /dev/xvdb

sudo echo 'UUID=<your-uuid> /mnt/data ext4 defaults,nofail 0 2' | sudo tee -a /etc/fstab

**5. Infrastructure as Code (Terraform)**

**Terraform Configuration (main.tf)**

provider "aws" {

region = "us-east-1"

}

resource "aws\_instance" "web" {

ami = "ami-0abcdef1234567890"

instance\_type = "t3.micro"

key\_name = "your-key"

tags = {

Name = "WebServerInstance"

}

}

**Apply Terraform Configuration**

terraform init

terraform apply -auto-approve

**6. CI/CD Automation**

**AWS CodeBuild (buildspec.yml)**

version: 0.2

phases:

install:

commands:

- sudo dnf install -y httpd php mysql

pre\_build:

commands:

- echo "Running pre-build tasks..."

build:

commands:

- echo "Building the application..."

post\_build:

commands:

- echo "Build completed."

artifacts:

files:

- '\*\*/\*'

**7. Security Configurations**

**Restrict SSH Access**

aws ec2 authorize-security-group-ingress --group-id sg-xxxxxxxx --protocol tcp --port 22 --cidr 203.0.113.0/32

**IAM Policy for Least Privilege Access**

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": "s3:ListBucket",

"Resource": "arn:aws:s3:::my-secure-bucket"

}

]

}

**Enable CloudTrail Logging**

aws cloudtrail create-trail --name MyTrail --s3-bucket-name my-log-bucket

aws cloudtrail start-logging --name MyTrail

This document consolidates essential scripts and configurations for deploying, securing, and managing Rocky Linux-based cloud applications on AWS.