**Building a Fault-Tolerant WordPress Environment on AWS**

**Step 1: Request Public Certification for Your Domain**

1. Provide the domain name (e.g., YourDomain.com) and the subdomain (\*.YourDomain.com, if applicable).

2. The certification request will use DNS validation, where you'll demonstrate control over the domain by adding specific DNS records.

3. Once validation is successful, the certification authority will issue a public certificate for your domain, securing your website with SSL/TLS encryption, enhancing its trustworthiness and security.

**Step 2: Create a Hosted Zone in Route 53 for Your Domain**

1. Sign in to the AWS Management Console and navigate to the Route 53 service.

2. Click on "Create Hosted Zone."

3. Enter your domain name (e.g., example.com) in the "Domain Name" field.

4. Choose the "Public hosted zone" option to manage publicly accessible domain DNS records.

5. Click on "Create" to create the hosted zone.

6. Note down the four nameservers (NS records) provided by Route 53 during hosted zone creation.

7. Access your 3rd-party domain provider's account where your domain is registered.

8. Locate the DNS management settings or domain settings for your domain.

9. Update the nameservers for your domain with the ones provided by Route 53.

10. Save the changes to update the nameservers.

11. Allow time for DNS changes to propagate across the internet (usually a few minutes to a couple of hours).

**Step 3: Create RDS Instances for Production and Recovery Servers**

1. Open the RDS Console: Navigate to the "Services" menu and select "RDS" under the "Database" category.

2. Choose "Create Database": Click on the "Create Database" button to start the RDS creation process.

­­-Server, etc.).

4. Specify Database Settings: Provide necessary details like DB instance identifier, username, and password. Select the instance class and storage type.

5. Configure Additional Settings: Set the VPC, subnet group, public accessibility, and other options based on your requirements.

6. Set Database Security: Define security group rules to control network access to your RDS instance.

7. Add Database Options: Configure additional database options like backup retention, maintenance, or monitoring.

8. Choose Backup and Maintenance Settings: Set the backup window, retention period, and preferred maintenance window.

9. Add Tags (Optional): Optionally, add tags to your RDS instance for better resource organization and management.

10. Review Configuration: Double-check all settings to ensure they are correct.

11. Create the RDS Instance: Click on the "Create Database" button to create the RDS instance with the provided configuration.

12. Wait for Creation: The RDS instance creation process may take a few minutes to complete.

13. Verify RDS Instance Status: Once created, go to the RDS dashboard and check that the instance status is "available."

**Step 4: Create IAM Role with EC2 and S3 Permissions**

To create an IAM role with EC2 and S3 permissions:

1. Sign in to the AWS Management Console and navigate to the IAM service.

2. In the IAM dashboard, click on "Roles" in the left-hand menu.

3. Click on the "Create role" button.

4. Choose the service that will use this role (select "EC2" from the list of services) and click "Next: Permissions."

5. Attach the necessary permissions policies for EC2 and S3 access to the role.

**Step 5: Create EC2 Instances for Production and Recovery Servers**

1. Sign in to the AWS Management Console and access the EC2 service.

2. Launch instances using suitable AMIs and instance types based on your application's requirements.

3. Configure instance details, such as network settings, IAM role, and advanced options.

4. Set up storage volumes, tags, and security group rules for both instances.

5. Review the instance launch settings and proceed to create the instances.

6. Choose or create a key pair for secure access to the instances.

7. After launching, configure your applications and services accordingly.

8. For the recovery server, implement a robust backup strategy and data synchronization for disaster recovery readiness.

**Step 6: Set Up WordPress on AWS EC2 Instance**

1. Attach the Bash Script in Advanced Details and Launch the Server: Include a Bash script in the advanced details section when creating the EC2 instance. This script will install and set up the required software for WordPress.

2. Access WordPress Host Page: After launching the server, open your browser and enter the EC2 instance's public IP address. You should see the WordPress host page.

3. Proceed with WordPress Installation: Click "Let's Go" on the bottom section of the page and provide the necessary database credentials to complete the WordPress installation.

4. Log in via PuTTY and Gain Root Access: Access the EC2 instance using PuTTY, and switch to root access for administrative privileges.

5. Navigate to WordPress Directory: Change the directory to /var/www/html, where WordPress files are stored.

6. Create and Edit Configuration Files: Use the vi editor to create and modify configuration files to fully access and configure WordPress.

7. Login with WordPress Credentials: After making the necessary configurations, log in to WordPress using the provided username and password.

**Follow similar steps for the recovery server.**

**Step 7: Create Classic Load Balancer (CLB) for Production and Recovery Servers and Attach to Route 53**

1. Create Classic Load Balancer: Sign in to the AWS Management Console, navigate to the EC2 service, and click on "Create Load Balancer." Choose "Classic Load Balancer" as the load balancer type. Configure settings for both production and recovery environments, including name, listener, security groups, and subnets. Specify health check settings to ensure traffic routes only to healthy instances.

2. Register Instances: After creating the Classic Load Balancer, register the production server EC2 instance(s) to it. Repeat the process to register the recovery server EC2 instance(s).

3. Create a Hosted Zone in Route 53: If you haven't already created a hosted zone for your domain in Route 53, follow the steps mentioned earlier to create one.

4. Create Alias Records: In the Route 53 dashboard, click on your hosted zone and select "Create Record Set." Choose "Alias" for the record type. Select the Classic Load Balancer as the target for both the production and recovery environments. Save the changes to create the Alias records.

5. Associate Domain with Alias Records: In your domain provider's account, update the DNS settings for your domain. Add two CNAME records, one for the production environment and another for the recovery environment, both pointing to their respective Alias records in Route 53.

6. Wait for DNS Propagation: DNS changes may take some time to propagate across the internet. Be patient and wait for the changes to take effect.

**Step 8: Create a Record in Hosted Zone with Simple Routing for DNS Validation and Attach an Application Load Balancer (ALB)**

1. Create a hosted zone for your domain in Route 53.

2. Create a record set with the subdomain name and select "Simple routing" as the routing policy.

3. Add two records with the same subdomain name: one with the ALB's DNS name for normal traffic routing and another with a random value for DNS validation purposes.

4. Attach the ALB to the record set and specify the AWS region where the ALB is deployed.

5. Save the changes, and the DNS records will be set up for DNS validation, and the ALB will handle traffic for the subdomain.

**Step 9: Add HTTPS to Application Load Balancer (ALB) and Associate ACM Certificate**

1. Go to the ALB settings in the AWS Management Console.

2. Edit the listeners and add a new HTTPS listener.

3. Select an existing ACM certificate or request a new one.

4. Configure security settings and select the target group for HTTPS traffic.

5. Save the changes, and the ALB will handle HTTPS traffic using the ACM certificate.

**Step 10: Create Two Buckets Named "media" and "text" in Amazon S3**

1. Sign in to the AWS Management Console and go to the S3 service.

2. Create the "media" bucket by entering the name and selecting a region.

3. Repeat the process to create the "text" bucket with a different name or region.

**Step 11: Synchronize Data between Production and Recovery Servers**

For both production and recovery servers:

1. Every two minutes, synchronize the `/var/www/html/wp-content/uploads` directory with the `s3://media` S3 bucket.

2. Every two minutes, synchronize the entire `/var/www/html/` directory (WordPress code) with the `s3://text` S3 bucket.

These Cron expressions ensure data synchronization between the production and disaster recovery environments, ensuring consistent backups and data availability for the WordPress setup.

**CONCLUSION** :By following these steps, you can build a fault-tolerant WordPress environment on AWS, ensuring high availability, security, and reliability for your website or blog.