WordPress Deployment on AWS EC2 (Monolithic Architecture)

1. Introduction

This project demonstrates how to deploy a WordPress website using a monolithic architecture on AWS. Both the **WordPress** application and **MySQL database** were hosted on a single **EC2 t2.micro instance (Ubuntu 22.04)**. The setup used **CyberPanel** for easy site and SSL management, and **Dynu** for free domain name resolution.

---

2. Project Goals

- Set up a **monolithic server architecture** using a single EC2 instance.
- Install and configure **WordPress** and **MySQL** on the same instance.
- Configure security groups to allow required ports.
- Design and set a welcome page as the homepage.

---

3. Tools and Technologies Used

- AWS EC2 (t2.micro)
- Ubuntu 22.04 LTS
- CyberPanel with OpenLiteSpeed
- WordPress CMS
- MySQL (via CyberPanel)
- Dynu (Free DNS Service)
- Let's Encrypt SSL
- Linux Terminal

---

4. Infrastructure Setup

Step 1: Create EC2 Instance

- **AMI:** Ubuntu Server 22.04 LTS
- **Instance Type:** t2.micro (Free Tier eligible)
- **Storage:** 8 GB (default)
- **Key Pair:** Generated .pem file for SSH access
- **Security Group (Ports Opened):**

- 22: SSH
- 80: HTTP
- 443: HTTPS
- 3389: Reserved (optional)

Step 2: Assign Elastic IP

1. Go to **EC2 Dashboard > Elastic IPs**.
2. Allocate a new IP and associate it with the running instance.

Step 3: Install CyberPanel

SSH into your EC2 instance:

ssh -i "your-key.pem" ubuntu@<your-elastic-ip>

Switch to root:

sudo -i

Download and run the CyberPanel installer:

cd /root

wget -O installer.sh https://cyberpanel.net/install.sh

chmod +x installer.sh

./installer.sh

During installation:

- Select 1 → Install CyberPanel with OpenLiteSpeed
- Choose 1 → Full installation
- Set admin password
- Enable PowerDNS, Postfix, Pure-FTPd as needed (optional)

Step 4: Access CyberPanel

Visit:

https://<your-elastic-ip>:8090

Login:

- **Username:** admin
- **Password:** (as set during installation)

---

5. Website Configuration

Step 5: Create a Website in CyberPanel

1. Go to **Websites > Create Website**
2. Enter:
    - Domain: your Dynu domain (e.g., yourdomain.dynu.com)
    - Email: valid email
    - PHP version: 8.1 or later
3. Click **Create Website**

Step 6: Point Domain to EC2 IP

1. Go to Dynu DNS
2. Create a free account and add your domain.
3. Set **A record** for @ and www to your Elastic IP.

Step 7: Install WordPress

From CyberPanel dashboard:

- Go to **Websites → List Websites → Manage**
- Under **Application Installer**, click **WordPress + LSCache**
- Enter:
    - Site title
    - Admin username/password
    - Email
- Click **Install Now**

Step 8: Install SSL

1. In the same site dashboard, scroll to **SSL > Issue SSL**
2. CyberPanel automatically fetches SSL from Let's Encrypt.

Check:

https://yourdomain.dynu.com

Step 9: Configure Welcome Homepage

1. Login to WordPress Admin:

https://yourdomain.dynu.com/wp-admin

2. Create a new **Page** titled "Welcome."
3. Add your content (text, images, etc.).
4. Go to **Settings > Reading**:
    - Set **Homepage displays** to "A static page"
    - Select the "Welcome" page as the homepage
5. Save Changes

Step 10: Migrate Existing Site (Optional)

If migrating an old site:

1. Use **All-in-One WP Migration** plugin on the original site.
2. Export your site.
3. Install the same plugin on your new WordPress instance and import the site.

---

6. Sample Commands Summary

Task	Command
SSH into EC2	ssh -i "your-key.pem" ubuntu@<ip>
Become root	sudo -i
Update system	apt update && apt upgrade -y
Install CyberPanel	wget -O installer.sh https://cyberpanel.net/install.sh && chmod +x installer.sh && ./installer.sh

---

7. References

- AWS EC2 Docs: https://docs.aws.amazon.com/ec2/
- CyberPanel Docs: https://cyberpanel.net/docs/
- WordPress: https://wordpress.org/
- Dynu DNS Setup: https://www.dynu.com/
- SSL (Let's Encrypt): https://letsencrypt.org/

8. Conclusion

This project successfully demonstrates a simple monolithic WordPress deployment on a single EC2 instance, using CyberPanel to simplify backend tasks. A welcome page was configured as the homepage, and SSL/DNS integration was handled seamlessly with Dynu and Let's Encrypt. This approach is ideal for low to medium traffic websites and development/test environments.