

host your Laravel application on AWS Cloud

1. Set Up an EC2 Instance on AWS

- **Log in to your AWS account.**
- **Navigate to EC2 (Elastic Compute Cloud)** and launch a new instance.
 - **Choose an AMI (Amazon Machine Image):** You can choose an Amazon Linux AMI or Ubuntu server.
 - **Instance Type:** Choose an instance based on your traffic requirements (e.g., t2.micro for a small project).
 - **Configure Instance:** Select the appropriate network settings.
 - **Add Storage:** Choose a storage size (e.g., 8GB for development).
 - **Configure Security Group:** Open ports 22 (SSH), 80 (HTTP), 443 (HTTPS) for web traffic, and port 3306 for MySQL if it's hosted on the same instance.
 - **Launch the Instance** and download the key pair for SSH access.

2. Install Required Software on EC2

- **SSH into your EC2 instance** using your key pair:

```
ssh -i /path/to/your-key.pem ec2-user@your-ec2-public-ip
```

- **Install Apache, PHP, Composer, MySQL, and other necessary packages:**

```
# For Amazon Linux (Use `sudo` for each command)
sudo yum update -y
sudo yum install -y httpd php php-mysqlnd php-fpm php-json php-xml php-mbstring git unzip
curl
sudo yum install -y mariadb-server
```

```
# Install Composer (for Laravel)
curl -sS https://getcomposer.org/installer | php
sudo mv composer.phar /usr/local/bin/composer
```

- **Start Apache and MySQL services:**

```
sudo systemctl start httpd
sudo systemctl enable httpd
sudo systemctl start mariadb
sudo systemctl enable mariadb
```

3. Set Up the Database (MySQL or MariaDB)

- **Secure MySQL installation:**

```
sudo mysql_secure_installation
```

- **Create a new database for your Laravel project:**

```
mysql -u root -p
CREATE DATABASE laravel_db;
CREATE USER 'laravel_user'@'localhost' IDENTIFIED BY 'your_password';
GRANT ALL PRIVILEGES ON laravel_db.* TO 'laravel_user'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

4. Upload Your Laravel Project to EC2

- You can **clone your Laravel project from GitHub** or **upload it directly via SFTP/FTP** or using **SCP**.
For example:

```
git clone https://github.com/yourusername/your-laravel-repo.git /var/www/html/your-project
```

- **Set the correct permissions:**

```
sudo chown -R apache:apache /var/www/html/your-project
sudo chmod -R 775 /var/www/html/your-project/storage
sudo chmod -R 775 /var/www/html/your-project/bootstrap/cache
```

5. Set Up Environment File (Laravel .env)

- Navigate to your Laravel project directory.
- Edit the .env file to configure the database connection:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel_db
DB_USERNAME=laravel_user
DB_PASSWORD=your_password
```

6. Install Laravel Dependencies

- Run **Composer** to install all Laravel dependencies:

```
cd /var/www/html/your-project
composer install --no-dev --optimize-autoloader
```

7. Set Up Apache Virtual Host for Laravel

- Create an Apache configuration for your Laravel site:

```
sudo vi /etc/httpd/conf.d/laravel.conf
```

- Add the following configuration:

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/your-project/public
    ServerName devnahid.com

    <Directory /var/www/html/your-project/public>
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

- Restart Apache to apply the changes:

```
sudo systemctl restart httpd
```

8. Configure Your Domain (devnahid.com)

- **Update DNS settings** for devnahid.com to point to your EC2 instance:
 - Go to your domain registrar (e.g., Namecheap, GoDaddy).
 - Find the DNS settings and add an **A record** pointing to your EC2 instance's public IP address.
 - Example:
 - Type: A
 - Name: @ (for the root domain)
 - Value: <your-ec2-public-ip>
- It may take up to 24-48 hours for DNS to propagate.

9. Set Up SSL (Optional, but recommended for production)

- **Install Certbot** and get a free SSL certificate from Let's Encrypt:

```
sudo yum install -y certbot python2-certbot-apache
sudo certbot --apache -d devnahid.com
```

- Certbot will automatically configure Apache for HTTPS.

10. Test the Laravel Project

- After the DNS has propagated, visit <http://devnahid.com> or <https://devnahid.com> (if you set up SSL).
- Your Laravel application should now be live!

11. Final Setup

- **Set up Laravel environment:**

```
php artisan key:generate  
php artisan migrate  
php artisan config:cache
```

- **Ensure your project is running smoothly** and that everything is configured properly.
-

Exporting a Database (Creating a Backup)

1. Export Using Command Line:

You can export a MySQL database into a .sql file with the mysqldump command. Run this command from your terminal:

```
mysqldump -u username -p database_name > backup_name.sql
```

- Replace username with your MySQL username (e.g., root or laravel_user).
- Replace database_name with the name of the database you want to export (e.g., laravel_db).
- Replace backup_name.sql with the desired name of the backup file.

Example:

```
mysqldump -u laravel_user -p laravel_db > laravel_db_backup.sql
```

- ### 2. Export Specific Tables:
- If you only want to export specific tables from your database, you can list them after the database name:

```
mysqldump -u username -p database_name table1 table2 > backup_name.sql
```

Importing a Database (Restoring a Backup)

To import a .sql file into your MySQL database, follow these steps:

- ### 1. Create a New Database
- (if needed): If the database you want to import into doesn't exist yet, you can create it with the following command:

```
mysql -u username -p -e "CREATE DATABASE database_name;"
```

Example:

```
mysql -u root -p -e "CREATE DATABASE laravel_db;"
```

2. **Import the SQL Backup:** To import a .sql file (backup) into your MySQL database, use the mysql command:

mysql -u username -p database_name < backup_name.sql

- Replace username with your MySQL username (e.g., root or laravel_user).
- Replace database_name with the name of the database into which you want to import the backup (e.g., laravel_db).
- Replace backup_name.sql with the name of the SQL file you want to import.

Example:

mysql -u laravel_user -p laravel_db < laravel_db_backup.sql