

## Deploying a awesomewebsite

.....

To deploy a website on localhost using Apache2 or Nginx and create a DNS name 'awesomeweb', follow these steps:

Install Nginx on Debian/Ubuntu:

```
""bash
sudo apt update
sudo apt install nginx
""
```

Step 2: Start the web server\*\*

we need to Start Nginx:

```
""bash
sudo service nginx start
""
```

Step 3: Configure the DNS name

1. Open the hosts file:

- vi /etc/hosts

2. Add an entry to the hosts file:

```
""
127.0.0.1 awesomeweb
""
```

Step 4: Create the website content

1. Create a directory to store the website files:

```
""bash
sudo mkdir /var/www/awesomeweb
""
```

2. Navigate to the website directory:

```
""bash
cd /var/www/awesomeweb
""
```

3. Create an HTML file with your website content. For example, you can create an 'index.html' file with the following content:

```
""
```

```
html
<!DOCTYPE html>
<html>
<head>
  <title>Awesome Website</title>
</head>
<body>
  <h1>Welcome to My Awesome Website!</h1>
```

```
<p>This is a simple HTML page hosted on localhost.</p>
</body>
</html>
""
```

## Step 5: Configure Nginx

### 1. Create a virtual host configuration file:

```
""bash
sudo nano /etc/nginx/sites-available/awesomeweb
""
```

### 2. Add the following content to the virtual host configuration:

```
""
server {
    listen 80;
    server_name awesomeweb;

    location / {
        root /var/www/awesomeweb;
        index index.html;
    }
}
""
```

### 3. Enable the virtual host by creating a symbolic link to the 'sites-enabled' directory:

```
""bash
sudo ln -s /etc/nginx/sites-available/awesomeweb /etc/nginx/sites-enabled/
""
```

### 4. Test the Nginx configuration for any syntax errors:

```
""bash
sudo nginx -t
""
```

### 5. If the test is successful, reload Nginx for the changes to take effect:

```
""bash
sudo service nginx reload
""
```

## Step 6: Access the website

\* Open your web browser and navigate to 'http://awesomeweb'. You should see your website content displayed.