
Task: Advanced Angular Application Development and Deployment Across Two Virtual Machines (VMs)

Objective:

Students will:

1. **Develop** an Angular Calculator application on one VM.
2. **Deploy** the Angular application to another VM using **Nginx**.
3. **Configure** Nginx to ensure optimal deployment (handling of single-page applications).
4. **Document** every step meticulously, including screenshots, error resolutions, and configuration details.
5. **Troubleshoot common issues** and ensure successful deployment.

Task Overview:

- **VM1 (Development VM)**: This VM will be used for developing the Angular application. The application will then be built and prepared for deployment.
- **VM2 (Deployment VM)**: The build from VM1 will be transferred and deployed on VM2 using Nginx.
- **Objective for Students**: Ensure the Angular application works perfectly when accessed via VM2's public IP. Handle common pitfalls such as routing in Angular and proper Nginx configuration.

Step-by-Step Instructions

1. Setting Up VM1 (Development VM)

1.1: VM1 Configuration

1. Set up a virtual machine (VM1) with a clean installation of **Ubuntu**.

Update system packages to ensure the environment is up to date:

```
sudo apt -y update
sudo apt -y upgrade
```

2.

Optionally, set the hostname for better identification (useful for larger environments):

```
sudo hostnamectl set-hostname vm1-dev
```

3.

1.2: Install Node.js, npm, and Angular CLI
Install **Node.js** and **npm**:

```
sudo apt install nodejs npm -y
```

1.

Ensure that `node` and `npm` versions are compatible by checking them:

```
node -v
npm -v
```

2.

Install **Angular CLI** globally:

```
sudo npm install -g @angular/cli
```

3.

1.3: Clone the Repository
Clone the Angular Calculator repository from GitHub into your project directory:

```
git clone https://github.com/Ai-TechNov/AngularCalculator.git
cd AngularCalculator
```

1.
2. **Screenshot 1**: Capture the terminal output showing the successful clone and the directory structure.

1.4: Install Project Dependencies

Install the required dependencies:

```
npm install
```

1.
2. **Screenshot 2**: Capture the terminal showing the successful installation of dependencies.

1.5: Build the Angular Application

Perform a production build of the Angular application:

```
sudo ng build --prod
```

1.
2. **Screenshot 3**: Capture the terminal output showing the successful completion of the build and the creation of the `dist/` folder.

---

2. Setting Up VM2 (Deployment VM)

2.1: VM2 Configuration

1. Set up a virtual machine (VM2) with a fresh installation of **Ubuntu**.

Update the system packages on VM2:

```
sudo apt -y update
sudo apt -y upgrade
```

2.

Set the hostname for VM2:

```
sudo hostnamectl set-hostname vm2-prod
```

3.

2.2: Install Nginx

Install **Nginx** on VM2:

```
sudo apt install nginx -y
```

1.
2. **Screenshot 4**: Capture the terminal showing the successful installation of Nginx.

Start the Nginx service and verify that it's running:

```
sudo systemctl status nginx
```

3.
4. **Screenshot 5**: Capture the terminal showing that Nginx is active and running.

---

3. Transfer the Angular Application to VM2

3.1: Transfer Build Files

On **VM1**, navigate to the `dist/angularCalc` directory:

```
cd dist/angularCalc
```

1.

Transfer the contents of the `dist/angularCalc/` directory to **VM2** using `scp`:

```
scp -r * username@vm2-ip:/var/www/html/
```

2. Replace `username` with your VM2 username and `vm2-ip` with the IP address of VM2.

3. **Screenshot 6**: Capture the `scp` command output, showing that files are being transferred to VM2.

3.2: Verify the Transfer

Log into VM2 and verify that the files have been successfully copied to `/var/www/html/`:

```
ls /var/www/html/
```

1.

2. **Screenshot 7**: Capture the terminal showing that all files from `dist/angularCalc/` are present in `/var/www/html/`.

---

4. Configure Nginx on VM2

4.1: Nginx Configuration for Angular SPA

Edit the default Nginx server block configuration to ensure that it serves the Angular application correctly. This is especially important for Single Page Applications (SPA) to handle client-side routing properly:

```
sudo nano /etc/nginx/sites-available/default
```

1.

Replace the server block with the following configuration:

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    root /var/www/html;
    index index.html;

    server_name _;

    location / {
        try_files $uri $uri/ /index.html;
    }
}
```

2. This configuration ensures that all routes are handled by Angular's `index.html` file.

3. **Screenshot 8**: Capture the Nginx configuration file (`/etc/nginx/sites-available/default`) showing the updated configuration.

4.2: Restart Nginx

After saving the configuration, restart Nginx to apply the changes:

```
sudo systemctl restart nginx
```

1.
2. **Screenshot 9**: Capture the terminal showing that Nginx has been restarted successfully.

---

5. Verify the Application on VM2

5.1: Testing the Deployment

1. Open a browser and navigate to `http://<vm2-ip>`, where `<vm2-ip>` is the IP address of VM2.

2. The Angular Calculator application should load in the browser.

3. **Screenshot 10**: Capture the web browser showing the Angular Calculator app running on VM2.

---

6. Detailed Documentation Submission

Students should submit a comprehensive document that includes:

1. **Step-by-step instructions** for the entire process, with **proper explanations**.

2. **Screenshots** for each major step in the process, as outlined in the instructions.

3. **Code snippets** showing the exact configuration changes made in the Nginx configuration file.

4. **Error Handling**:

    - Describe any issues encountered during the deployment process and how they were resolved (e.g., issues with routing in Angular, incorrect Nginx configurations, missing dependencies).

5. **Recommendations** for future improvements or optimizations in the deployment process.

Evaluation Criteria

1. **Correctness**:
    - Ensure the Angular application is built and deployed correctly.
    - Verify that routing works properly (using the `try_files` configuration in Nginx).
2. **Documentation**:
    - The report should be thorough, well-structured, and contain detailed explanations of the steps.
    - All screenshots should be clear and in the right context.
3. **Troubleshooting**:
    - Ability to identify and solve issues related to building the Angular app, transferring files, configuring Nginx, and handling Angular routing in production.
4. **Functionality**:
    - The application should be fully functional and accessible via the public IP of VM2.

Additional Notes

- **Networking**: Ensure both VMs are on the same network, and that VM2's firewall allows HTTP traffic (port 80).
- **Security**: Make sure that the Nginx configuration is secure and only serves the required files.