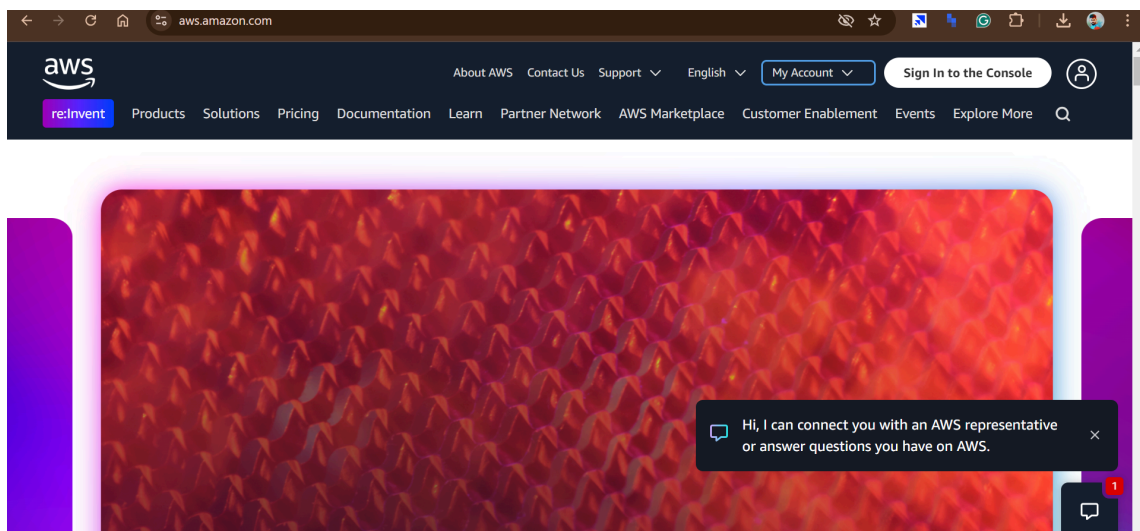


## Steps to Deploy an Angular Application in an ec2 instance with Dockeraize and without Dockeraize

### Method 1 :

#### Without Dockeraize :

**Step 1: Log in to your AWS console using your credentials.**



**Step 2: Search for Ec2 in the AWS Console. Click on Launch Instance. (Ubuntu / Linux)**

Dashboard

EC2 Global View

Events

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

AMI Catalog

Resources

EC2 Global View

You are using the following Amazon EC2 resources in the Asia Pacific (Hyderabad) Region:

Instances (running)	1	Auto Scaling Groups	0
Capacity Reservations	0	Dedicated Hosts	0
Elastic IPs	1	Instances	2
Key pairs	2	Load balancers	0
Placement groups	0	Security groups	4
Snapshots	0	Volumes	3

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the

Service health

AWS Health Dashboard

EC2 Free Tier

Offers for all AWS Regions.

2 EC2 free tier offers in use

End of month forecast

1 offers forecasted to exceed free tier limit.

Exceeds free tier

1 offers exceeded and is now pay-as-you-go pricing.

View Global EC2 resources

Offer usage (monthly)

Linux EC2 Instances

537.371389 hours remaining

28%

Storage space on EBS

## Step 3: Connect to your Ec2 instance

Dashboard

EC2 Global View

Events

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

AMI Catalog

Instances (2)

Info

Last updated less than a minute ago

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

All states

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
	i-08a3488a93a46fdf0	Terminated	c5.large	-	View alarms +
Practice	i-00d75cd2c301598e5	Running	t3.micro	3/3 checks passed	View alarms +

Select an instance

EC2

Instances

i-00d75cd2c301598e5

Connect to instance

Connection type

Connect using EC2 Instance Connect

Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address.

Connect using EC2 Instance Connect Endpoint

Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IPv4 address

18.61.57.3

IPv6 address

Username

Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.

ubuntu

Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel

Connect

**Step 4: Download and install Nodejs and npm. Use the following commands.**

```
root@ip-172-31-37-185:/home/ubuntu# sudo apt install nodejs -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nodejs is already the newest version (18.19.1+dfsg-6ubuntu5).
nodejs set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

- `Npm install -g @angular/cli` ( Install Angular CLI globally).

[illegible]

- *Ng new my-angular-app, cd my-angular-app (Create a new Angular app and go inside the created director).*

```
root@ip-172-31-37-185:/home/ubuntu# cd my-angular-app/
root@ip-172-31-37-185:/home/ubuntu/my-angular-app#
```

- `Ng serve`(Test the Angular app locally).

```

root@ip-172-31-37-185:/home/ubuntu# cd my-angular-app/
root@ip-172-31-37-185:/home/ubuntu/my-angular-app# ng serve
Browser bundles
Initial chunk files | Names          | Raw size
polyfills.js        | polyfills      | 90.20 kB |
main.js             | main           | 18.31 kB |
styles.css          | styles         | 95 bytes |
                    | Initial total  | 108.61 kB

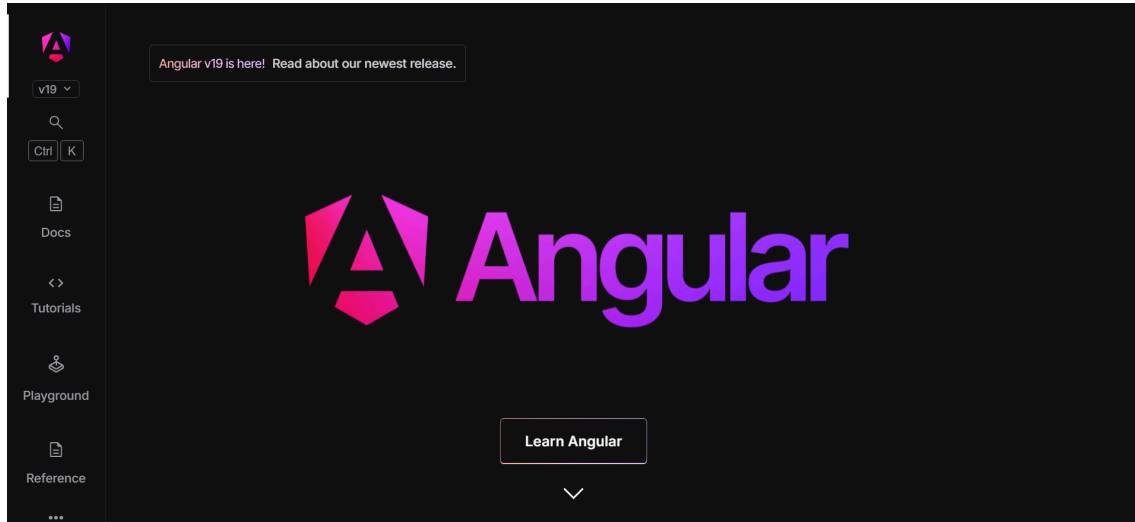
Server bundles
Initial chunk files | Names          | Raw size
polyfills.server.mjs | polyfills.server | 572.91 kB |
main.server.mjs     | main.server     | 19.55 kB |
server.mjs          | server          | 1.86 kB |

Application bundle generation complete. [16.838 seconds]

Watch mode enabled. Watching for file changes...
NOTE: Raw file sizes do not reflect development server per-request transformations.
  → Local:   http://localhost:4200/
  → press h + enter to show help

```

- *Ng build --prod / ng build --configuration production ( Build the production-ready app . The build files will be in the dist/my-angular-app folder).*
- *In the network settings of your ec2 instance, add all SSH, HTTP, and a custom TCP rule for your application port (e.g., 4200).*
- *Copy your ec2 instance public IP address or the private IP address along with your application port id and paste it into the web browser and your application is good to go.*



## Method 2 : With Dockerize

*Step 1: Create a Dockerfile in the root of your Angular project.*

*# Use a lightweight web server*  
*FROM nginx:alpine*

*# Copy built Angular files to the NGINX HTML folder*  
*COPY dist/<project-name> /usr/share/nginx/html*

*# Expose the default NGINX port*  
*EXPOSE 80*

*# Start NGINX*  
*CMD ["nginx", "-g", "daemon off;"]*

Replace <project-name> with the actual name of your Angular project.

Step 2 : Build the Docker Image :

```
docker build -t angular-app .
```

Step 3: Push the image (Optional )

Push the image to docker Hub or any other container registry .

```
docker tag angular-app  
<your-dockerhub-username>/angular-app  
docker push <your-dockerhub-username>/angular-app
```

Step 4 : Connect to your Ec2 instance .

Step 5 : Pull your docker image from your docker hub or any other registry .

```
docker pull <your-dockerhub-username>/angular-app
```

Step 6: Run the container

```
docker run -d -p 80:80 angular-app
```

Your Angular app will be now available on <http://<EC2-Instance-IP>>

