

## AWS EC2 Setup and Configuration:

1. Login to AWS Console
2. Go to Services → EC2 → Launch Instance
3. Select the following configuration:

**Name and tags** [Info](#)

Name

WebServer

Add additional tags

**▼ Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Q Search our full catalog including 1000s of application and OS images

Recents

Quick Start

Amazon Linux

aws

macOS

Mac

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

Red Hat

SUSE Linux

SUSE

Q

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

**Amazon Machine Image (AMI)**

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type  
ami-0fc5d935ebf8bc3bc (64-bit (x86)) / ami-016485166ec7fa705 (64-bit (Arm))  
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible ▼

**Description**

Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-09-19

Architecture

64-bit (x86) ▼

AMI ID

ami-0fc5d935ebf8bc3bc

Verified provider

▼ Instance type
Info

Instance type

t2.micro

Family: t2
1 vCPU
1 GiB Memory
Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.0716 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

Free tier eligible

☐ All generations
  
[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

▼ Key pair (login)
Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

WebServer

Create new key pair

Use the default VPC and Subnets.

Allow SSH and HTTP/HTTPS in security groups.

- Click on “Launch Instance”
- Once the instance 2 checks click on “connect”.
- Run “*sudo apt-get update && sudo apt-get upgrade -y*” command to update Ubuntu.

aws
Services
Search
[Alt+S]

S3
VPC
EC2
IAM
Amazon Lex
CloudWatch
RDS
Route 53

```

ubuntu@ip-172-31-28-108:~$ sudo apt-get update && sudo apt-get upgrade -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1103 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [239 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [16.0 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1036 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [167 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [536 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [995 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [218 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [22.0 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [41.6 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [9768 B]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [472 B]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [64.2 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [10.5 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [388 B]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [27.8 kB]

```

i-06f98317cc803d79e (WebServer)
  
PublicIPs: 34.229.218.228 PrivateIPs: 172.31.28.108

7. Once the upgrade is done, use command *"sudo apt-get install apache2"*

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-28-108:~$ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

8. Install "npm" and "nodejs" using the command :

*"sudo apt install npm"*

*"sudo apt install nodejs"*

## Application Hosting Files Setup:

Here we are going to create a sample VUE.js application using VUE CLI.

1. Navigate to the html directory using command

*"cd /var/www/html"*.

2. Install the VUE CLI using command:

*"npm install -g @vue/cli"*

```
ubuntu@ip-172-31-28-108:~$ npm install -g @vue/cli
( [REDACTED] ) ⚙️ reify:@types/ejs: timing reify:loadBundles Completed in 0ms
```

3. Create a sample project using the command:

*"vue create hello-world"*

Select Default [Vue 3]

```
Vue CLI v5.0.8
? Please pick a preset: (Use arrow keys)
> Default ([Vue 3] babel, eslint)
  Default ([Vue 2] babel, eslint)
  Manually select features
```

4. Now once the project is created. Navigate to the root directory of the project and use *"npm run build"* to run the application.

```
DONE Compiled successfully in 4025ms
```

```
App running at:
```

```
- Local:   http://localhost:8080/  
- Network: http://172.31.28.108:8080/
```

The web application will not be available to the internet unless we configure the Apache server

## Configuring the Apache server:

To make our WebApp accessible over the internet we need to setup a reverse proxy using the Apache server configurations.

1. We need to enable "proxy" and "proxy\_http" modules using the following commands:

```
"sudo a2enmod proxy && sudo a2enmod proxy_http"
```

2. Restart Apache Server

```
"Sudo systemctl restart apache2"
```

```
ubuntu@ip-172-31-28-108:~$ sudo a2enmod proxy && sudo a2enmod proxy_http  
Module proxy already enabled  
Considering dependency proxy for proxy_http:  
Module proxy already enabled  
Module proxy_http already enabled  
ubuntu@ip-172-31-28-108:~$ sudo systemctl restart apache2
```

3. Navigate to `/etc/apache2/sites-available/000-default.conf`  
Edit the configuration file using any editor to your required settings/configuration

4. Edit the Following:

```
GNU nano 6.2
<VirtualHost *:80>

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    ProxyPreserveHost On
    ProxyPass / http://localhost:8080/
    # Edit the address to your App address and Port
    ProxyPassReverse
    http://localhost:8080/

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

5. Restart Apache.

Now Copy your EC2 Instance Public IP and Past it on the Browser.



## Welcome to Your Vue.js App

For a guide and recipes on how to configure / customize this project,  
check out the [vue-cli documentation](#).

### Installed CLI Plugins

[babel](#) [eslint](#)

### Essential Links

[Core Docs](#) [Forum](#) [Community Chat](#) [Twitter](#) [News](#)

### Ecosystem

[vue-router](#) [vuex](#) [vue-devtools](#) [vue-loader](#) [awesome-vue](#)

