

AWS-12

Deploy a project from GitHub to EC2 without using port.

1. Sign-in to AWS console. Then Go to the EC2 dashboard. Now go to the instances page. Click on the create new instance button. Now create an EC2 server using the Security Group created earlier and enter the user data (Refer to Ass10).

Select

Metadata response hop limit [Info](#)

Select

Allow tags in metadata [Info](#)

Select

User data - optional [Info](#)
Enter user data in the field.

```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
git clone https://github.com/kumari-suchi/MyRep2.git
cd MyRep2
cd New-Repo1
npm install
node index.js
```

☐ User data has already been base64 encoded

Summary

Number of instances [Info](#)

1

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...[read more](#)
ami-0d979355d03fa2522

Virtual server type (instance type)

t2.micro

Firewall (security group)

mysec2

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage per AWS Region.

Cancel **Launch instance** [Review commands](#)

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

2. Now successfully instances created.

New EC2 Experience [Tell us what you think](#)

EC2 Dashboard
EC2 Global View
Events
Limits
▼ Instances

Instances (1) [Info](#)

[Refresh](#) [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input type="checkbox"/>	Suchi12	i-03134cf800ee9b98c	Running	t2.micro	Initializing	No alarms	ap-northeast-1c	ec2-43-206-153-...

3. Then click on the instance after creation. Copy the public IPv4 address and paste it in another browser. The nginx homepage will show up.

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

4. Then Copy the Public IPv4 address of the server instance and use this to connect it to the server using Bitwise SSH client. (Refer Ass7)
5. Then open the terminal in Bitwise.
6. Then Enter the following commands in it.

➤ **pwd (To check current directory)**

```
ubuntu@ip-172-31-15-208:~$ pwd
/home/ubuntu
```

➤ **cd /**

```
ubuntu@ip-172-31-15-208:~$ cd /
```

(To go to root folder)

➤ **pwd**

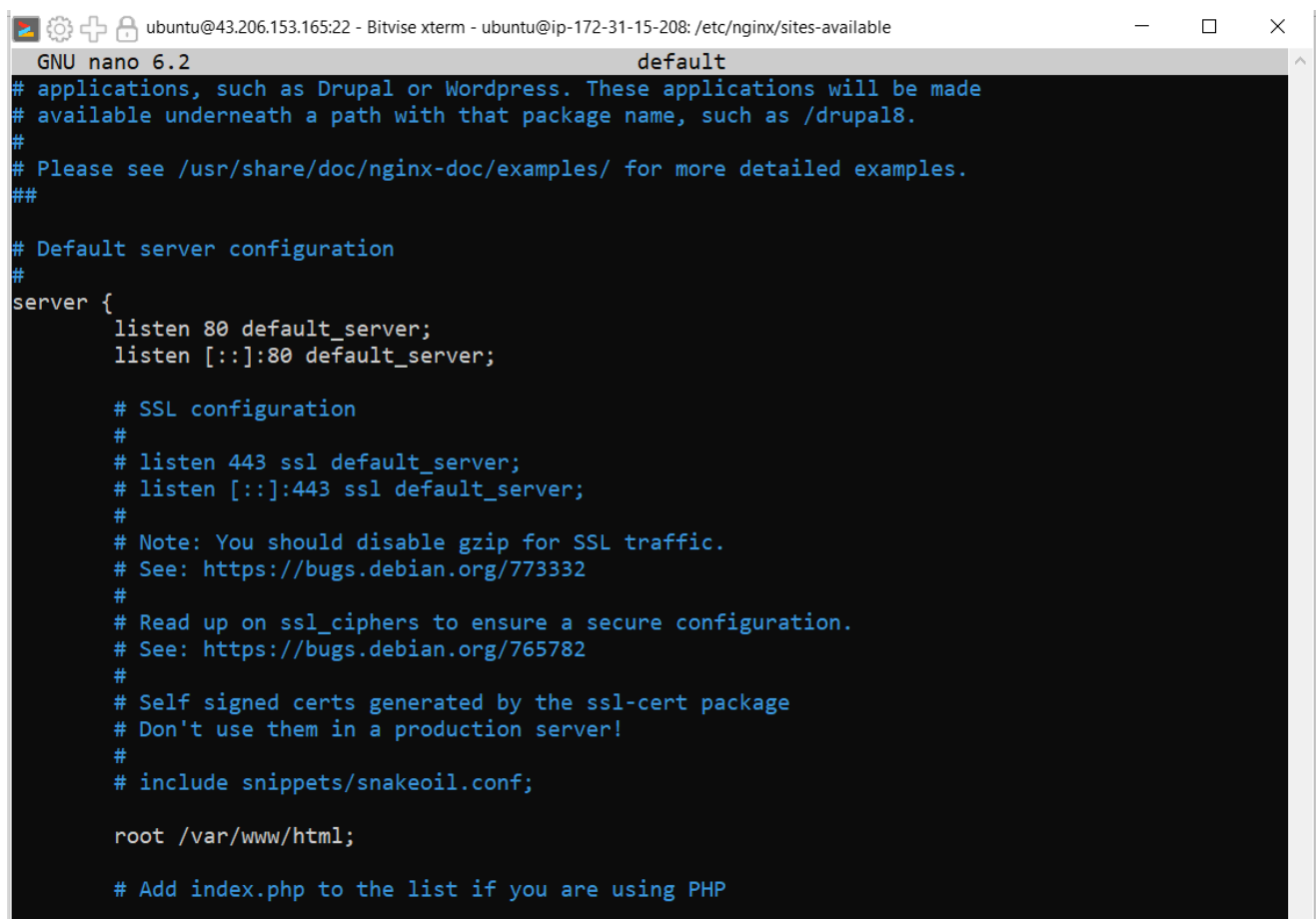
```
ubuntu@ip-172-31-15-208:/$ pwd
/
```

➤ **cd /etc/nginx/sites-available/**

```
ubuntu@ip-172-31-15-208:/$ cd /etc/nginx/sites-available/
ubuntu@ip-172-31-15-208:/etc/nginx/sites-available$
```

(To open the sites-available directory under nginx)

➤ **sudo nano default**



The screenshot shows a terminal window titled 'ubuntu@43.206.153.165:22 - Bitwise xterm - ubuntu@ip-172-31-15-208: /etc/nginx/sites-available'. Inside the terminal, the 'nano' editor is open, editing the 'default' file. The editor's title bar says 'GNU nano 6.2 default'. The content of the file is as follows:

```
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as /drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;

    root /var/www/html;

    # Add index.php to the list if you are using PHP
```

7. After opening the default file in the nano editor, search for the location / {}. It should be after server_name _;. Then comment out the location block and each and every line inside the block.

```

server_name _;

#location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    #try_files $uri $uri/ =404;
#}

```

8. Then paste the following code just under the closing curly bracket.

```

location / {
    proxy_pass http://localhost:4000;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'Upgrade';
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
}

```

```

server_name _;

#location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    #try_files $uri $uri/ =404;
#}
location / {
    proxy_pass http://localhost:4000;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'Upgrade';
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
}

```

9. Now save it by Ctrl+X and exit nano editor. You will be reverted back to the terminal. Type the following command –

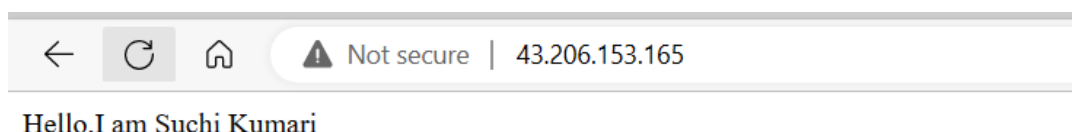
➤ **sudo systemctl restart nginx**

```

ubuntu@ip-172-31-14-117:/etc/nginx/sites-available$ sudo systemctl restart nginx
ubuntu@ip-172-31-14-117:/etc/nginx/sites-available$

```

10. Now paste the public IPv4 address in your browser. Our project page will show up without entering our port no.



We have successfully deployed a project from GitHub to EC2 without using port.