

# Deployment of Nodejs Application

## Contents to check:

There are two types of nodejs applications.

- Frontend nodejs Application.
- Backend nodejs Application.

### Frontend nodejs Application:

- We have to check package.json file to know the properties of the application

Note: Frontend application default port no. is **3000**.

### Backend nodejs Application:

- We have to check package.json file to know the properties of the application.
- Check whether it has the file with name 'server.js'.
- Check the port No. in 'server.js' file

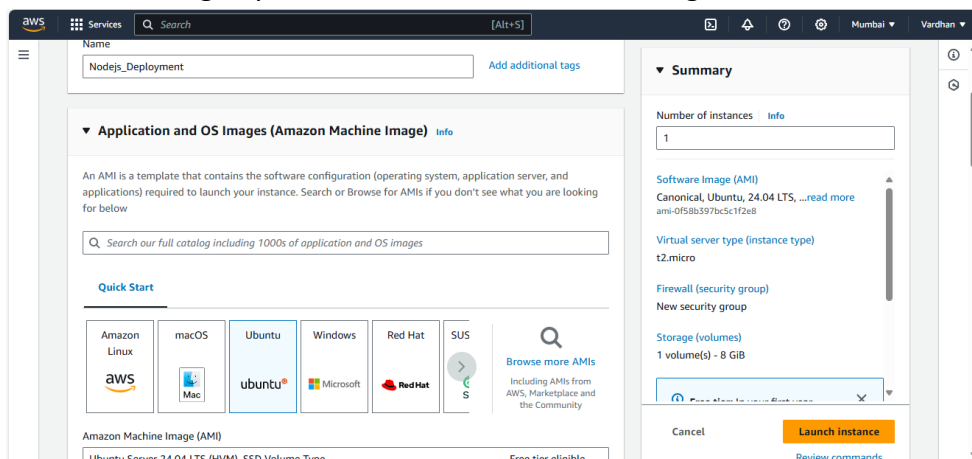
Note: Backend application has its port no. in 'server.js' file

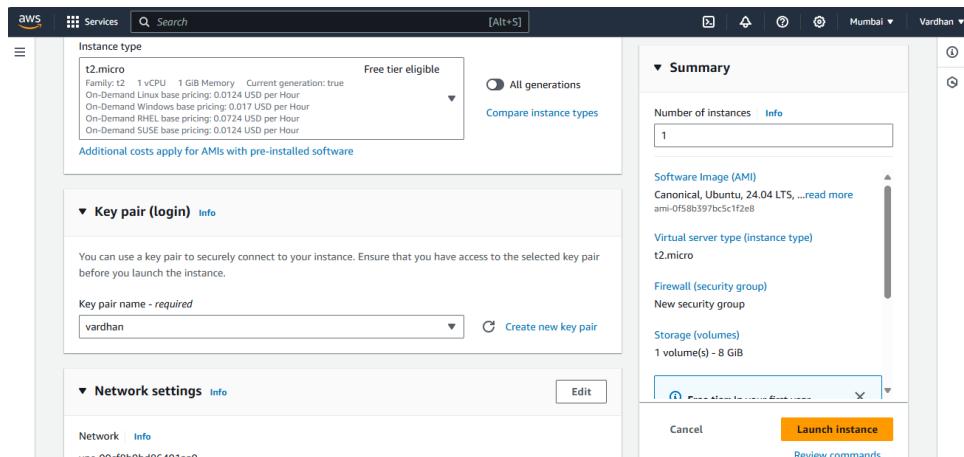
## Requirements to Deploy a Nodejs Application:

- OS.
- Server.
- Region to reduce Latency/Response Time.
- We have to install nodejs, npm.
- After installing npm we have to install pm2, serve and build using npm.

## Steps to deploy a Nodejs Application:

1. Create a Server using any Cloud Service. Here we are using AWS to create a server.





2. Login to your server using any SSH. Here we are using **Git Bash**.

```
ubuntu@ip-172-31-46-47: ~
HP@DESKTOP-4E1U0U6 MINGW64 ~/Desktop
$ ssh -i "vardhan.pem" ubuntu@ec2-3-108-44-27.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-3-108-44-27.ap-south-1.compute.amazonaws.com (3.10
8.44.27)' can't be established.
ED25519 key fingerprint is SHA256:J2VomX8j6JAn25pRU9ns5/Nb6RFnZwIVvRHoyP0gkVw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-108-44-27.ap-south-1.compute.amazonaws.com' (E
D25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1008-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Tue May 28 02:08:48 UTC 2024

System load: 0.72          Processes:            107
Usage of /:  23.2% of 6.71GB Users logged in:          0
Memory usage: 20%          IPv4 address for enx0: 172.31.46.47
Swap usage:  0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-46-47:~$ |
```

### 3. Update your server.

In Ubuntu Linux       =>     **apt update**  
In Amazon Linux       =>     **yum update**

```
ubuntu@ip-172-31-46-47: ~  
HP@DESKTOP-4E1U0U6 MINGW64 ~/Desktop  
$ ssh -i "vardhan.pem" ubuntu@ec2-3-108-44-27.ap-south-1.compute.amazonaws.com  
The authenticity of host 'ec2-3-108-44-27.ap-south-1.compute.amazonaws.com (3.108.44.27)' can't be established.  
ED25519 key fingerprint is SHA256:J2Vomx8j6JAn25pRU9ns5/Nb6RFnZwIVvRHoyP0gkvw.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
warning: Permanently added 'ec2-3-108-44-27.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.  
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1008-aws x86_64)  
  
* documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/pro  
  
System information as of Tue May 28 02:08:48 UTC 2024  
  
System load:  0.72      Processes:           107  
Usage of /:   23.2% of 6.71GB   Users logged in:     0  
Memory usage: 20%          IPv4 address for enx0: 172.31.46.47  
Swap usage:   0%  
  
Expanded Security Maintenance for Applications is not enabled.  
  
0 updates can be applied immediately.  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the
```

### 4. Install 'nodejs' to our server.

In Ubuntu Linux       =>     **apt install nodejs**  
In Amazon Linux       =>     **yum install nodejs**

```
root@ip-172-31-46-47: ~  
root@ip-172-31-46-47:~# apt install nodejs  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  libcares2 libnode109 node-acorn node-busboy node-cjs-module-lexer node-undici node-  
Suggested packages:  
  npm  
The following NEW packages will be installed:  
  libcares2 libnode109 node-acorn node-busboy node-cjs-module-lexer node-undici node-  
0 upgraded, 9 newly installed, 0 to remove and 33 not upgraded.  
Need to get 16.1 MB of archives.  
After this operation, 70.4 MB of additional disk space will be used.  
Do you want to continue? [Y/n] yes  
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libcares2 amd64  
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 node-xten  
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 nodejs am  
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 node-acor  
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 node-cjs-  
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 node-busb  
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 node-undi  
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libnode10  
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 nodejs-do  
Fetched 16.1 MB in 0s (60.5 MB/s)  
Selecting previously unselected package libcares2:amd64.  
(Reading database ... 71839 files and directories currently installed.)  
Preparing to unpack .../0-libcares2_1.27.0-1.0ubuntu1_amd64.deb ...
```

5. Install **npm** (Node Package Manager) to our server.

In Ubuntu Linux       =>     **apt install npm**  
In Amazon Linux       =>     **yum install npm**

```
root@ip-172-31-46-47: ~  
root@ip-172-31-46-47:~# apt install npm  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  binutils binutils-common binutils-x86-64-linux-gnu build-essential bzip2 cpp cpp-13 c  
  g++-13-x86-64-linux-gnu g++-x86-64-linux-gnu gcc gcc-13 gcc-13-base gcc-13-x86-64-lin  
  libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan8 lib
```

npm is used to resolve the modules, libraries and dependencies of the application.

6. Install **pm2** into our server using npm.

Command to install pm2       =>     **npm i -g pm2**

```
root@ip-172-31-46-47: ~  
root@ip-172-31-46-47:~# npm i -g pm2  
  
added 138 packages in 22s  
  
13 packages are looking for funding  
  run `npm fund` for details
```

pm2 helps us to deploy our application permanently. It will stop working until we stop it.

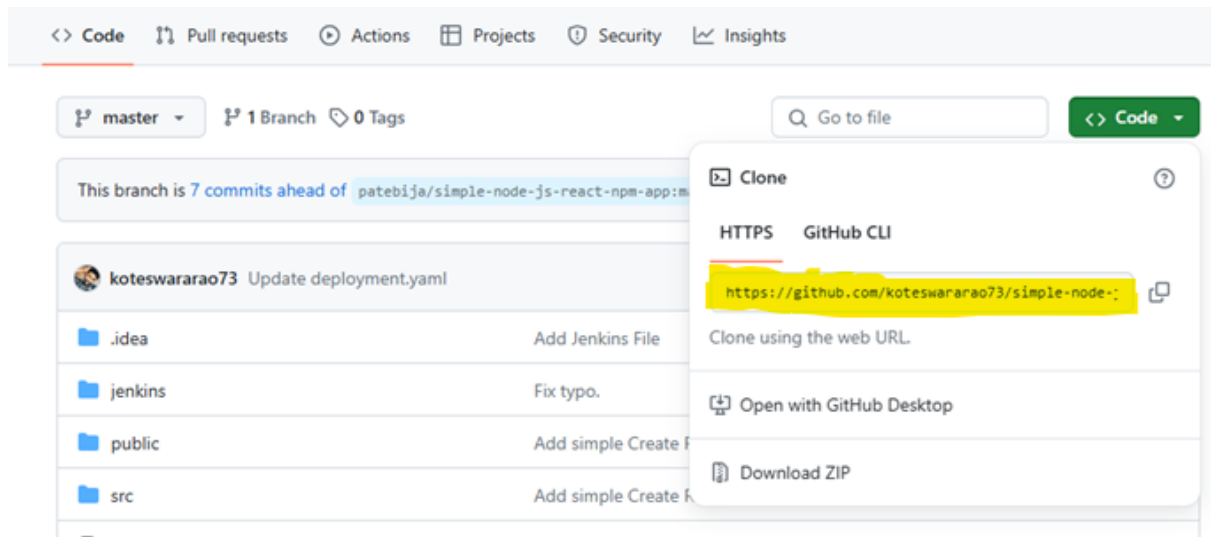
7. Install **serve** into server using npm.

Command to install serve       =>     **npm i -g serve**

```
root@ip-172-31-46-47: ~  
root@ip-172-31-46-47:~# npm i -g serve  
  
added 89 packages in 9s  
  
24 packages are looking for funding  
  run `npm fund` for details  
root@ip-172-31-46-47:~#  
root@ip-172-31-46-47:~# |
```

8. Here we have to import our application to our server. Here we are importing from github.

Command to import form github => **git clone < URL>**



```
root@ip-172-31-46-47: ~
root@ip-172-31-46-47:~# git clone https://github.com/koteswararao73/simple-node-js-react-npm-app.git
Cloning into 'simple-node-js-react-npm-app'...
remote: Enumerating objects: 130, done.
remote: Total 130 (delta 0), reused 0 (delta 0), pack-reused 130
Receiving objects: 100% (130/130), 28.82 KiB | 3.20 MiB/s, done.
Resolving deltas: 100% (53/53), done.
root@ip-172-31-46-47:~#
```

9. Open the directory of the application.

Command to open directory => **cd <directoryName>**

```
root@ip-172-31-46-47: ~/simple-node-js-react-npm-app
root@ip-172-31-46-47:~# ls
simple-node-js-react-npm-app snap
root@ip-172-31-46-47:~# cd simple-node-js-react-npm-app/
root@ip-172-31-46-47:~/simple-node-js-react-npm-app# ls
Dockerfile Jenkinsfile README.md deployment.yaml jenkins package.json public src
root@ip-172-31-46-47:~/simple-node-js-react-npm-app#
```

## 10. Resolve modules, libraries, dependencies using npm.

Command to resolve => **npm i**

```
root@ip-172-31-46-47: ~/simple-node-js-react-npm-app
root@ip-172-31-46-47:~/simple-node-js-react-npm-app# npm i
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated source-map-url@0.4.1: See https://github.com/lydell/source-map-url#deprecated
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated acorn-dynamic-import@2.0.2: This is probably built in to whatever tool you're using. If you still need
nc requests by a key value, which is much more comprehensive and powerful.
npm WARN deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if
npm WARN deprecated rimraf@2.7.1: Rimraf versions prior to v4 are no longer supported
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated abab@1.0.4: Use your platform's native atob() and btoa() methods instead
npm WARN deprecated rimraf@2.6.3: Rimraf versions prior to v4 are no longer supported
npm WARN deprecated eslint-loader@1.9.0: This loader has been deprecated. Please use eslint-webpack-plugin
npm WARN deprecated flatten@1.0.3: flatten is deprecated in favor of utility frameworks such as lodash.
npm WARN deprecated content-type-parser@1.0.2: Use whatwg-mimetype instead
npm WARN deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm WARN deprecated browserslist@1.7.7: Browserslist 2 could fail on reading Browserslist >3.0 config used in other tools
npm WARN deprecated browserslist@1.7.7: Browserslist 2 could fail on reading Browserslist >3.0 config used in other tools
```

Here it creates a directory named 'node\_modules'.

```
root@ip-172-31-46-47: ~/simple-node-js-react-npm-app
root@ip-172-31-46-47:~/simple-node-js-react-npm-app# ls
Dockerfile Jenkinsfile README.md deployment.yaml jenkins node_modules package-lock.json package.json public src
root@ip-172-31-46-47:~/simple-node-js-react-npm-app#
```

## 11. Create a build for an application using npm.

Command to create a build => **npm run build**

```
root@ip-172-31-46-47: ~/simple-node-js-react-npm-app
root@ip-172-31-46-47:~/simple-node-js-react-npm-app# npm run build

> my-app@0.1.0 build
> react-scripts build

Creating an optimized production build...
Compiled successfully.

File sizes after gzip:

  45.03 KB  build/static/js/main.9940296e.js
  299 B     build/static/css/main.c17080f1.css

The project was built assuming it is hosted at the server root.
To override this, specify the homepage in your package.json.
For example, add this to build it for Github Pages:

  "homepage" : "http://myname.github.io/myapp",

The build folder is ready to be deployed.
You may serve it with a static server:

  serve -s build
```

Here we have different ways for deploying Frontend Application and Backend Application

Note: frontend step 13 start from page No.7 and Backend step 13 start from page No.10

### For Frontend Application:

After creating build of an application in Frontend it creates a directory with name 'build'.

```
root@ip-172-31-46-47:~/simple-node-js-react-npm-app# ls
Dockerfile Jenkinsfile README.md build deployment.yaml jenkins
root@ip-172-31-46-47:~/simple-node-js-react-npm-app#
```

12. Now we have to serve the created build to a port no. using pm2.

Command to serve an application => **pm2 serve build <portNo.> --spa**

```
root@ip-172-31-46-47:~/simple-node-js-react-npm-app# pm2 serve build 3000 --spa
[PM2] Starting /usr/local/lib/node_modules/pm2/lib/API/Serve.js in fork_mode (1 instance)
[PM2] Done.
[PM2] Serving /root/simple-node-js-react-npm-app/build on port 3000
```

id	name	namespace	version	mode	pid	uptime	u	status	cpu	mem	use
0	static-page-server-3000	default	5.4.0	fork	15468	0s	0	online	0%	44.6mb	root

```
root@ip-172-31-46-47:~/simple-node-js-react-npm-app#
```

13. Save the configuration of application using pm2.

Command to save configuration of Application => **pm2 startup**

```
root@ip-172-31-46-47:~/simple-node-js-react-npm-app# pm2 startup
[PM2] Init System found: systemd
Platform systemd
Template
[Unit]
Description=PM2 process manager
Documentation=https://pm2.keymetrics.io/
After=network.target

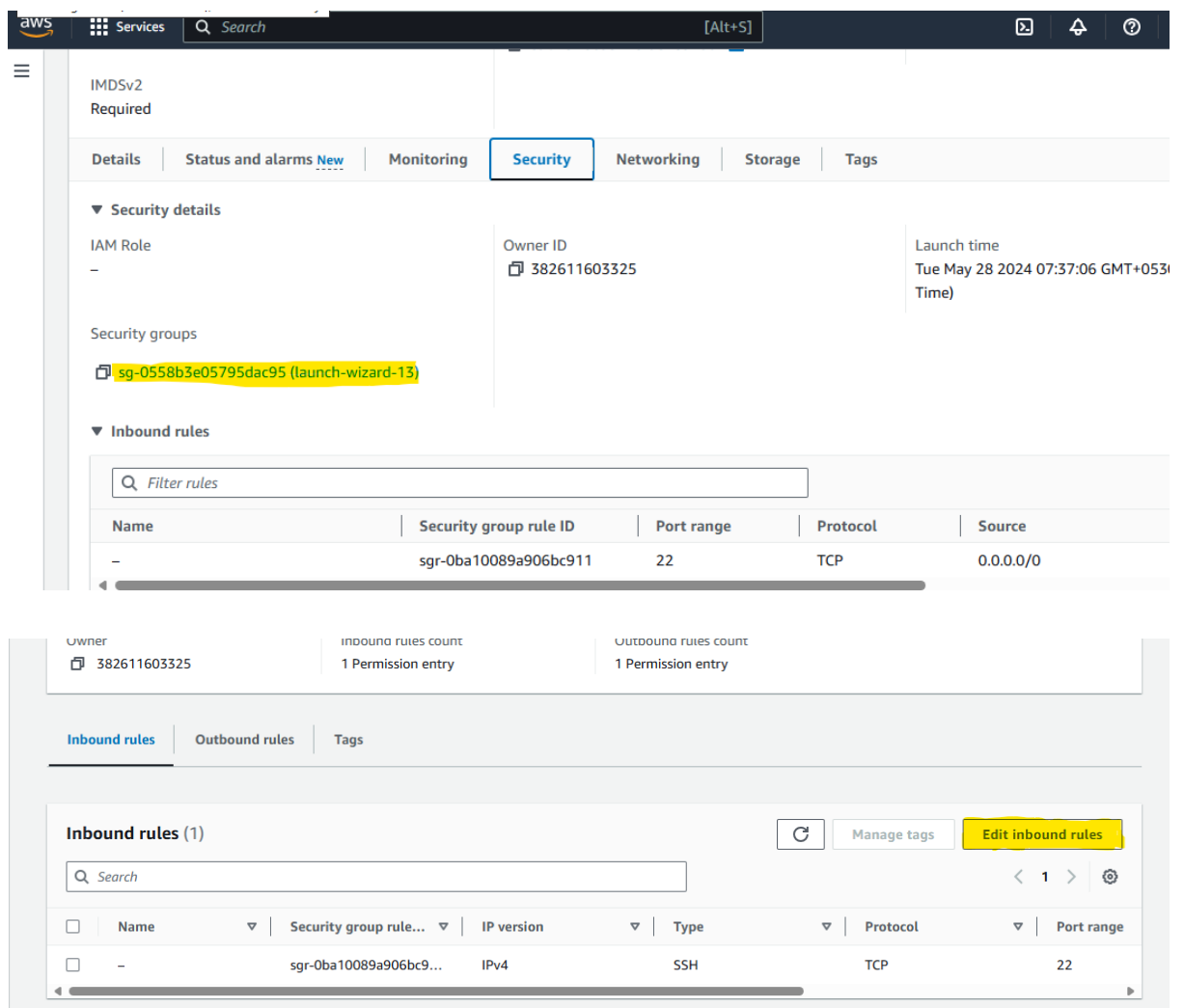
[Service]
Type=forking
User=root
LimitNOFILE=infinity
LimitNPROC=infinity
LimitCORE=infinity
Environment=PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin
Environment=PM2_HOME=/root/.pm2
PIDFile=/root/.pm2/pm2.pid
Restart=on-failure
```

14. Save the configuration path permanently using npm.

Command to save configuration path => **pm2 save -f**

```
root@ip-172-31-46-47: ~/simple-node-js-react-npm-app
root@ip-172-31-46-47:~/simple-node-js-react-npm-app# pm2 save -f
[PM2] Saving current process list...
[PM2] Successfully saved in /root/.pm2/dump.pm2
root@ip-172-31-46-47:~/simple-node-js-react-npm-app#
```

15. Go to Security groups and Allow required port in security groups to deploy Application in.



Port No. for frontend application is by default 3000.



**Inbound rules** [Info](#)

Security group rule ID	Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>	
sgr-0ba10089a906bc911	SSH	TCP	22	Cus...	0.0.0.0/0	Delete
-	Custom TCP	TCP	3000	Any...	0.0.0.0/0	Delete

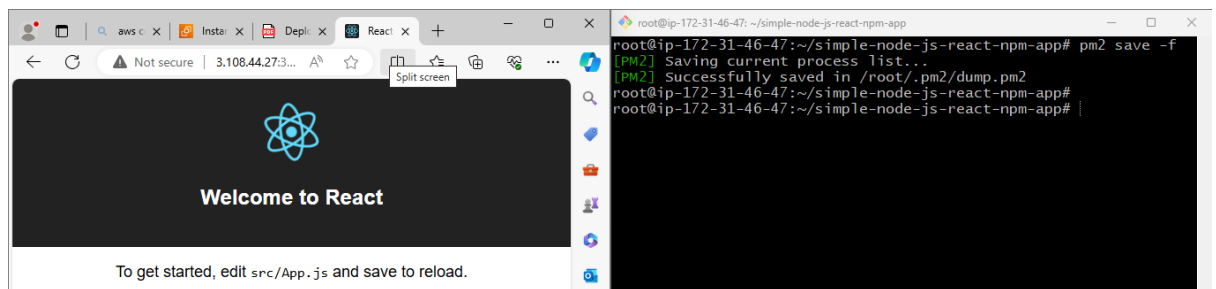
[Add rule](#)

**Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.**

[Cancel](#) [Preview changes](#) [Save rules](#)

16. Here permanent deployment have completed, now you can check it by entering public IP along with port No. in URL of any browser.

==> <publicIP>:<PortNo.>



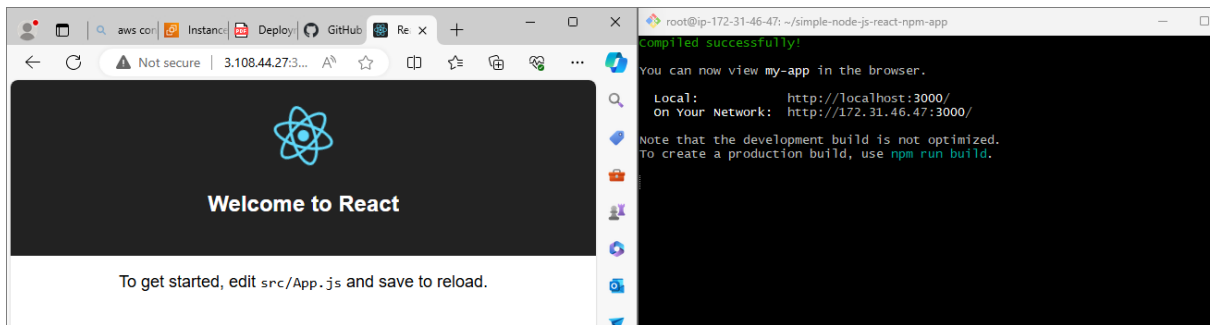
For temporary deployment we use command : **npm start build**.

```

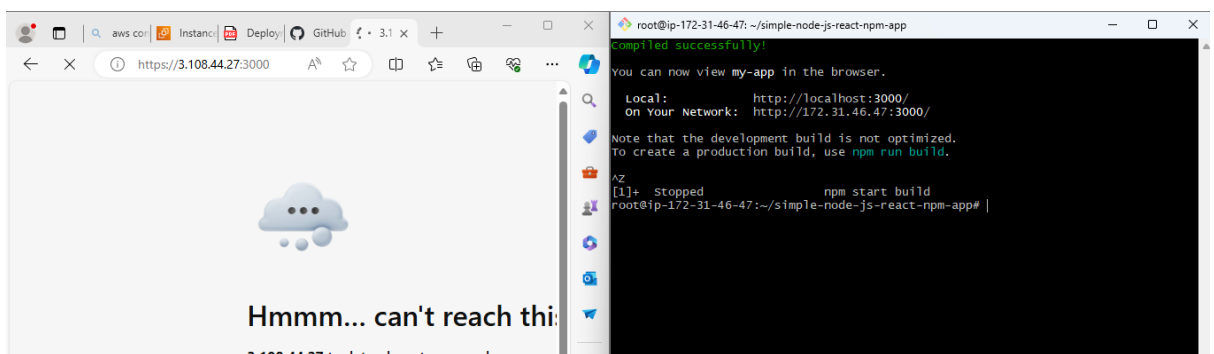
root@ip-172-31-46-47: ~/simple-node-js-react-npm-app
root@ip-172-31-46-47:~/simple-node-js-react-npm-app# npm start build

> my-app@0.1.0 start
> react-scripts start build

```

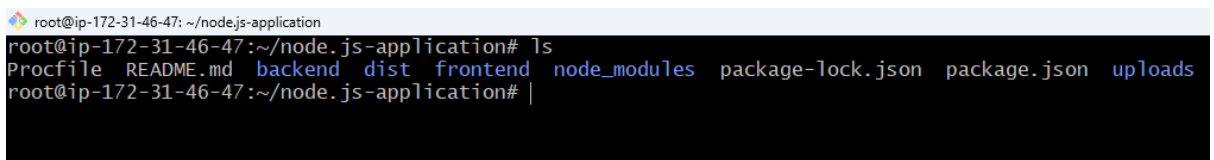


Once we exit that step it stops working.



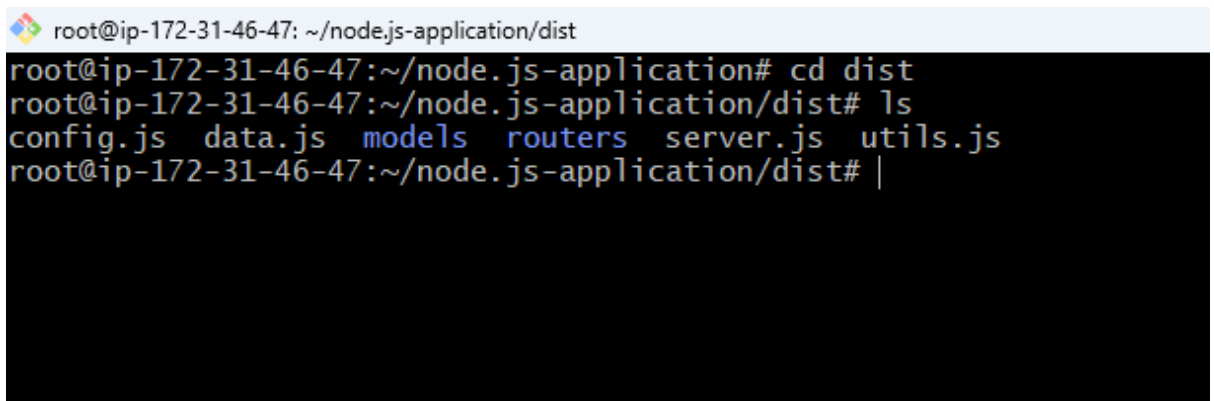
## For Backend Application:

After creating build of an application in Frontend it creates a directory with name 'dist'.



13. Now open the created directory 'dist'

Command to open 'dist' directory => **cd dist**



14. Now start 'server.js' file using pm2.

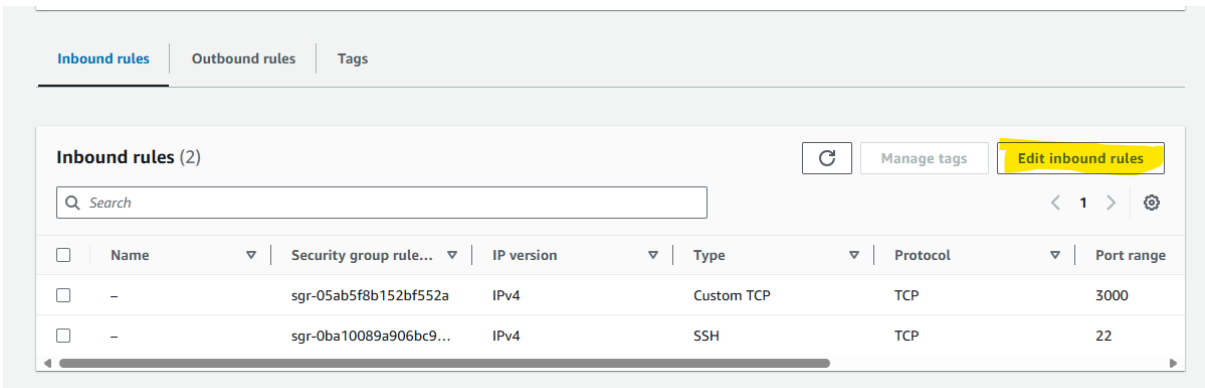
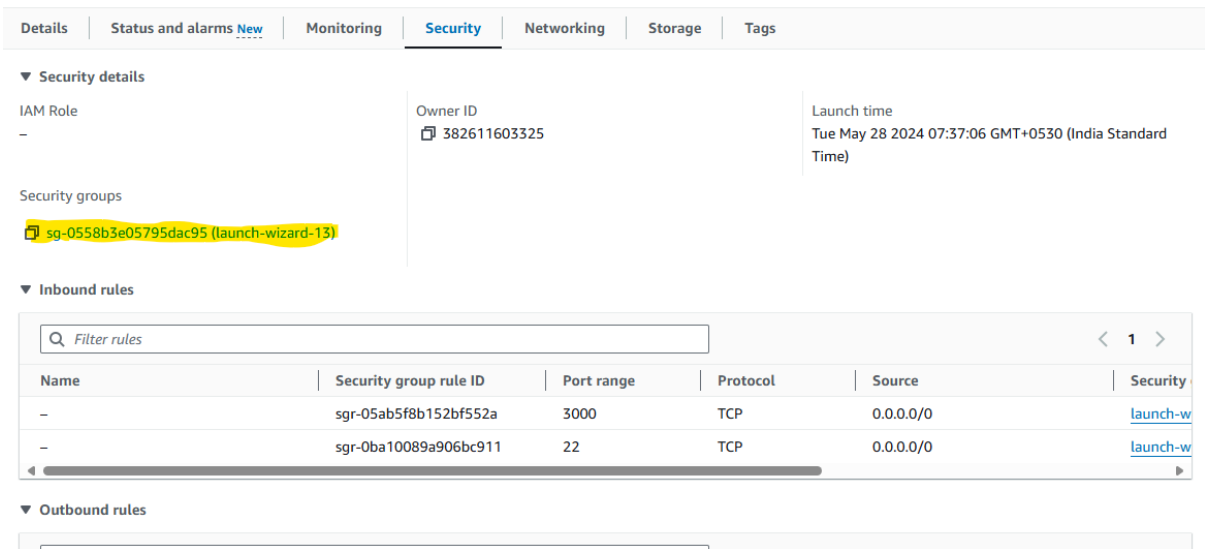
Command to start 'server.js' file      =>      **pm2 start server.js**

```
root@ip-172-31-46-47: ~/node.js-application/dist
root@ip-172-31-46-47:~/node.js-application/dist# pm2 start server.js
[PM2] Starting /root/node.js-application/dist/server.js in fork_mode (1 instance)
[PM2] Done.
```

id	name	namespace	version	mode	pid	uptime	u	status	cpu	mem	user	watching
0	server	default	1.0.0	fork	1117	0s	0	online	0%	11.3mb	root	disabled

```
[PM2][WARN] Current process list is not synchronized with saved list. App static-page-server-3000 differs. Type 'pm2 save' to synchronize.
root@ip-172-31-46-47:~/node.js-application/dist#
```

15. Go to Security groups and Allow required port in security groups to deploy Application in.



Here we have port No. for backend application in server.js file.

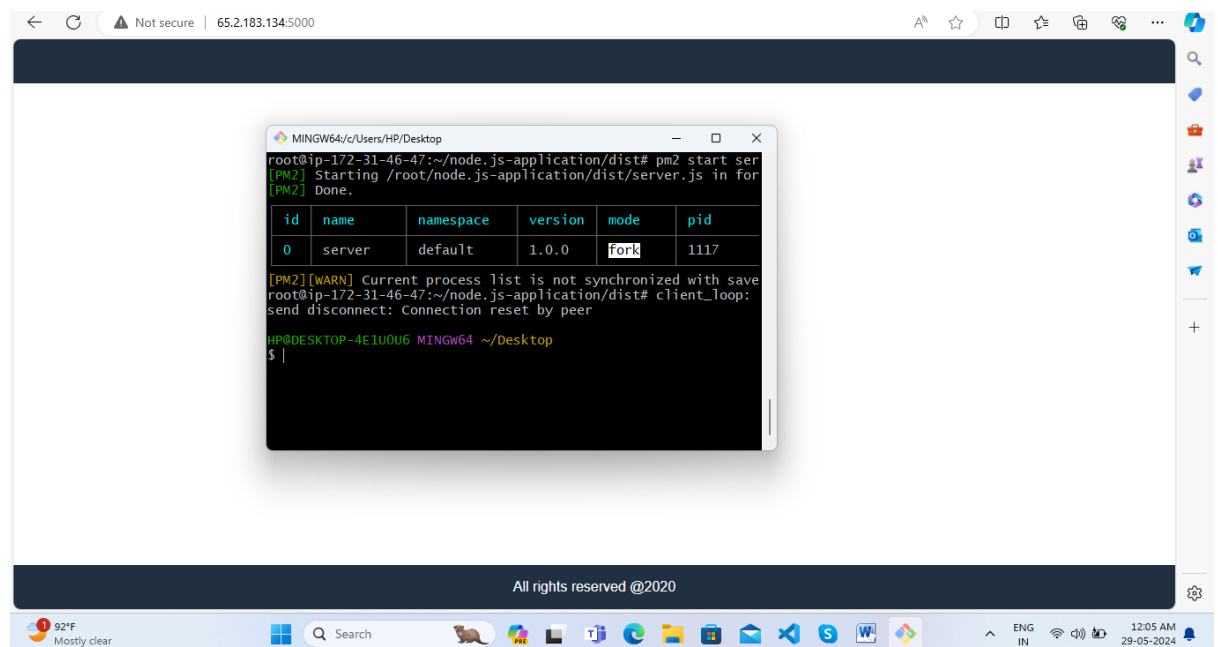
**Inbound rules** [Info](#)

Security group rule ID	Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>	
sgr-05ab5f8b152bf552a	Custom TCP ▼	TCP	3000	Cus... ▼	<input type="text" value="0.0.0.0/0"/>	<input type="text"/> <input type="button" value="Delete"/>
sgr-0ba10089a906bc911	SSH ▼	TCP	22	Cus... ▼	<input type="text" value="0.0.0.0/0"/>	<input type="text"/> <input type="button" value="Delete"/>
-	Custom TCP ▼	TCP	5000	Any... ▼	<input type="text" value="0.0.0.0/0"/>	<input type="text"/> <input type="button" value="Delete"/>

**Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.**

16. Here permanent deployment have completed, now you can check it by entering public IP along with port No. in URL of any browser.

==> <publicIP>:<PortNo.>



## Some pm2 commands:

- **pm2 logs** => It gives the logs of pm2.

```
root@ip-172-31-46-47: ~/simple-node-js-react-npm-app
root@ip-172-31-46-47:~/simple-node-js-react-npm-app# pm2 logs
[TAILING] Tailing last 15 lines for [all] processes (change the value with --lines option)
/root/.pm2/pm2.log last 15 lines:
PM2 | 2024-05-28T18:22:26: PM2 log: App [static-page-server-3000:0] online
PM2 | 2024-05-28T18:22:26: PM2 log: App [server:1] online
PM2 | 2024-05-28T18:23:23: PM2 log: Stopping app:server id:1
PM2 | 2024-05-28T18:23:23: PM2 log: App [server:1] exited with code [0] via signal [SIGINT]
PM2 | 2024-05-28T18:23:23: PM2 log: pid=830 msg=process killed
PM2 | 2024-05-28T18:25:27: PM2 log: App [server:2] starting in -fork mode-
PM2 | 2024-05-28T18:25:27: PM2 log: App [server:2] online
PM2 | 2024-05-28T18:26:11: PM2 log: Stopping app:static-page-server-3000 id:0
PM2 | 2024-05-28T18:26:11: PM2 log: App [static-page-server-3000:0] exited with code [0] via signal [SIGINT]
PM2 | 2024-05-28T18:26:11: PM2 log: pid=829 msg=process killed
PM2 | 2024-05-28T18:26:16: PM2 log: Stopping app:server id:2
PM2 | 2024-05-28T18:26:16: PM2 log: App [server:2] exited with code [0] via signal [SIGINT]
PM2 | 2024-05-28T18:26:16: PM2 log: pid=1060 msg=process killed
PM2 | 2024-05-28T18:26:25: PM2 log: App [server:0] starting in -fork mode-
PM2 | 2024-05-28T18:26:25: PM2 log: App [server:0] online
```

- **pm2 ls** => it gives list of applications that are online.

```
root@ip-172-31-46-47: ~/simple-node-js-react-npm-app
root@ip-172-31-46-47:~/simple-node-js-react-npm-app# pm2 ls
```

id	name	namespace	version	mode	pid	uptime	o	status	cpu	mem	user
0	server	default	1.0.0	fork	1117	18m	0	online	0%	76.8mb	root
1	static-page-server-3000	default	5.4.0	fork	1278	70s	0	online	0%	56.7mb	root

```
root@ip-172-31-46-47:~/simple-node-js-react-npm-app#
```

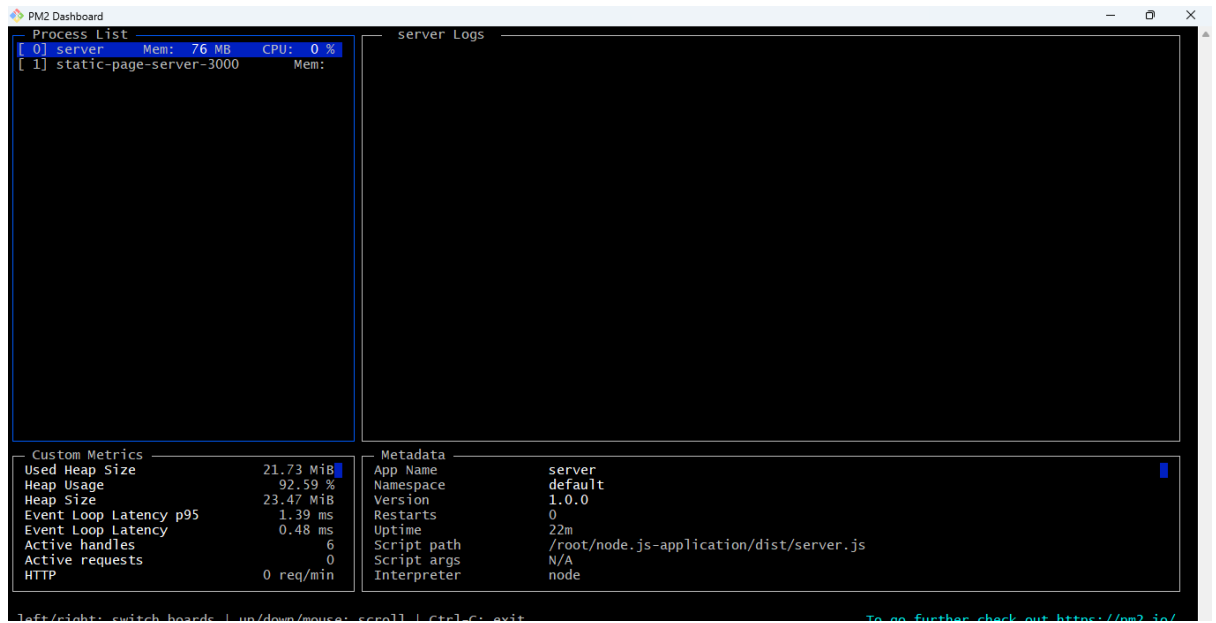
- **pm2 show <id>** => It gives the full information of the application running in given id

```
root@ip-172-31-46-47: ~/simple-node-js-react-npm-app
root@ip-172-31-46-47:~/simple-node-js-react-npm-app# pm2 show 0
Describing process with id 0 - name server
```

status	online
name	server
namespace	default
version	1.0.0
restarts	0
uptime	19m
script path	/root/node.js-application/dist/server.js
script args	N/A
error log path	/root/.pm2/logs/server-error.log
out log path	/root/.pm2/logs/server-out.log
pid path	/root/.pm2/pids/server-0.pid
interpreter	node
interpreter args	N/A
script id	0
exec cwd	/root/node.js-application/dist
exec mode	fork_mode
node.js version	18.19.1
node env	N/A
watch & reload	x
unstable restarts	0
created at	2024-05-28T18:26:25.653Z

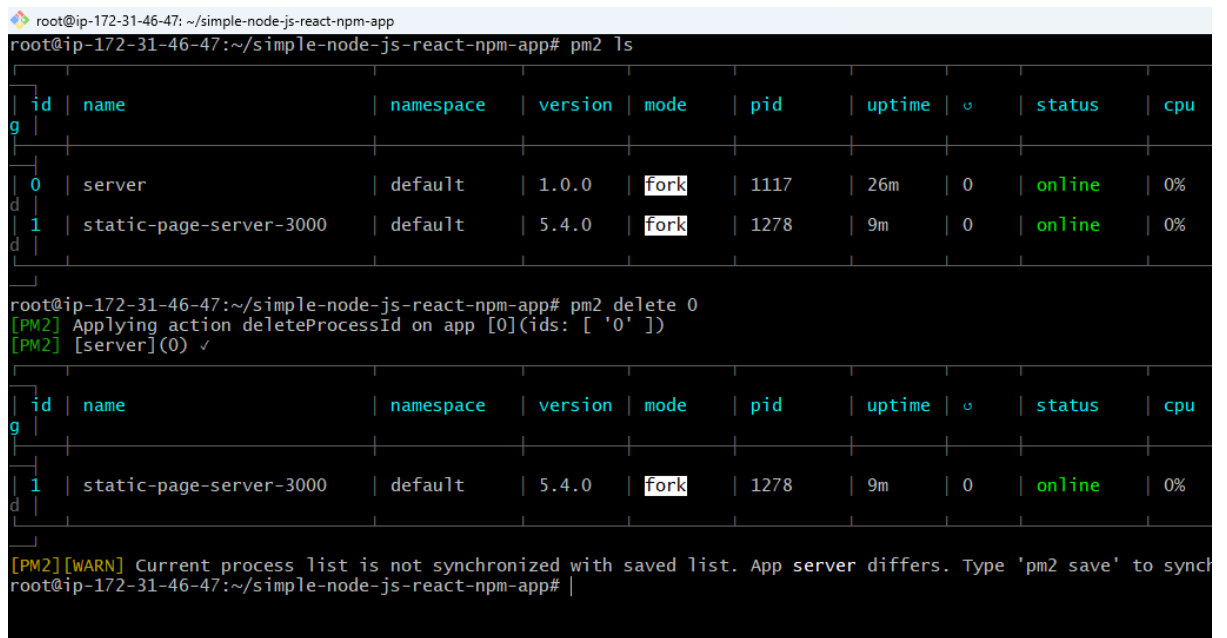
```
Revision control metadata
```

- **pm2 monit** ==> It helps to monitor the servers that are live.



We have to use 'ctrl+c' to exit monit mode.

- **pm2 delete <id>** ==> Delete the running application present in given id.



- **pm2 delete all**      =>      Deletes all running applications from pm2.

```

root@ip-172-31-46-47: ~/node.js-application/dist
root@ip-172-31-46-47:~/node.js-application/dist# pm2 ls

```

id	name	namespace	version	mode	pid	uptime	u	status	cpu	mem	user	watchin
2	server	default	1.0.0	Fork	1380	36s	0	online	0%	74.7mb	root	disable
1	static-page-server-3000	default	5.4.0	Fork	1278	11m	0	online	0%	56.7mb	root	disable

```

root@ip-172-31-46-47:~/node.js-application/dist# pm2 delete all
[PM2] Applying action deleteProcessId on app [all](ids: [ 1, 2 ])
[PM2] [static-page-server-3000](1) ✓
[PM2] [server](2) ✓


```

id	name	namespace	version	mode	pid	uptime	u	status	cpu	mem	user	watching
----	------	-----------	---------	------	-----	--------	---	--------	-----	-----	------	----------

```

[PM2] [WARN] Current process list is not synchronized with saved list. App static-page-server-3000 server differs. Type 'pm2 save' to synchronize.
root@ip-172-31-46-47:~/node.js-application/dist#

```