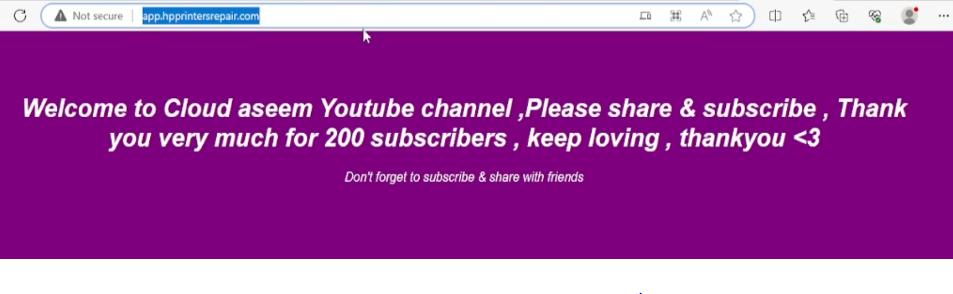
"Welcome to Cloud aseem Youtube channel"\*\*



**aseemnode**

**Node public repo subscribe to my channel , thanks you**

**Simple node.js app that servers "Welcome to Cloud aseem Youtube channel"**

Great for testing simple deployments on Cloud

**Step 1: Install NodeJS and NPM using nvm**

Follow this Medium blog to install Node.js & npm on EC2 server. Link below

<https://medium.com/@mohdaseemakram19/node-js-introduction-installation-of-node-js-on-aws-ec2-2350b7b95581>

sudo apt-get update

sudo apt-get install -y ca-certificates curl gnupg sudo mkdir -p /etc/apt/keyrings curl -fsSL <https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key> | sudo gpg --dearmor -o /etc/apt/keyrings/nodesource.gpg

NODE\_MAJOR=20 echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg] <https://deb.nodesource.com/node_$NODE_MAJOR.x> nodistro main" | sudo tee /etc/apt/sources.list.d/nodesource.list

sudo apt-get update

sudo apt-get install nodejs -y

Check Installation of nodejs packages

node -v v18.18.2

npm -v 9.8.1

**Step 2: Install Git and clone repository from GitHub**

To install git, run below commands in the terminal window:

sudo apt-get update -y

sudo apt-get install git -y

Just to verify if system has git installed or not, please run below command in terminal:

git — version

git clone <https://github.com/Aseemakram19/node-js-docker-cicd.git>

cd node-js-docker-cicd

1. Run the following command to install the required dependencies (Express): npm install express
2. Once the installation is complete, run the following command to start the server: nohup node index.js &
3. You should see the message "Server is running on [http://ip:3000](http://ip:3000/)" in the terminal.
4. Open your web browser and navigate to [http://ip:3000](http://ip:3000/). You should see the welcome page with the specified styling.
5. Kill the existing build on error

sudo lsof -i :3000 list the app running on

port kill -15 pid

Restart the process again

1. npm install express
2. nohup node index.js &

**Docker file Template**

# Use an official Node.js runtime as a base image

FROM node:14

# Copy package.json and package-lock.json to the working directory

COPY package\*.json ./

# Install application dependencies

RUN npm install express

# Copy the application files to the working directory

COPY . .

# Expose the port that the app will run on

EXPOSE 3000

# Define the command to run your application

CMD ["node", "index.js"]

docker build -t your-image-name .

docker run -d -p 3000:3000 your-image-name

docker stop CONTAINER\_ID\_OR\_NAME

# Build docker image of Docker file

docker build . -t nodeaseem1

# Create Docker container of Docker image

docker run -d -p 3000:3000 --name nodecontainer nodeaseem1

**# Add permission for Docker to Jenkins**

sudo usermod -aG docker Jenkins

sudo usermod -aG Jenkins $USER

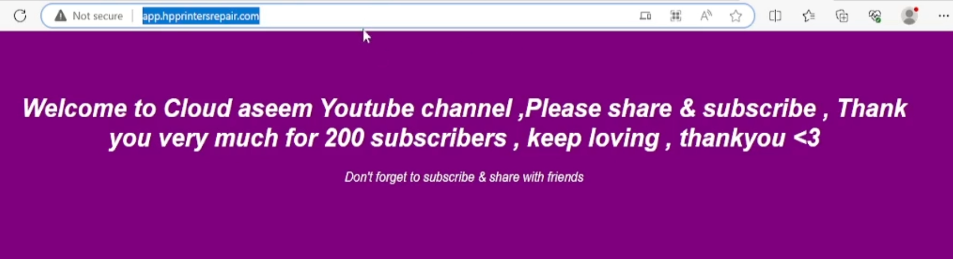
**# Restart the Jenkins after permissions Granted**

sudo systemctl restart Jenkins

**# Restart the Docker after permissions Granted**

sudo systemctl restart docker

Access the Application with ip:3000 in browser



**Reverse Proxy Config File for Subdomain – Nginx web server**

server {

listen 80;

server\_name app.hpprintersrepair.com; # Replace with your domain name or server IP address

location / {

proxy\_pass http://127.0.0.1:3000;

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;

}

# Additional configurations if needed...

}

**# check test syntax of config file**

sudo nginx -t

**# Copy the file by creating symbolic link of sites-availables to site-enable**

sudo ln -s /etc/nginx/sites-available/ appproxy /etc/nginx/sites-enabled

**# Restart Nginx service after creating Config files**

sudo systemctl restart nginx

**# install Cerbot for SSL configuration on subdomain**

sudo apt-get update

sudo apt-get install certbot

**# Install Plugin of Certbot**

sudo apt-get install certbot python3-certbot-nginx

**# Request to issue ssl certificate for our domain with below command**

sudo certbot --nginx -d app.hpprintersrepair.com

**# restart the service after application**

sudo systemctl restart nginx

Open your web browser and navigate to https://app.hpprintersrepair.com   
in my case . You can assign our own Domains

