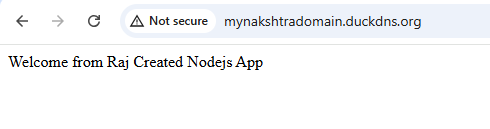
**1.Create Duck DNS Account**

Login > reCAPTCHA > add domain

Update IP



**2.Live Server Setup (Ubuntu EC2)**

sudo apt update

node --version

sudo apt install nodejs -y

npm --version

sudo apt install npm -y

1. **DuckDNS Setup**

mkdir duckdns

cd duckdns

nano duck.sh

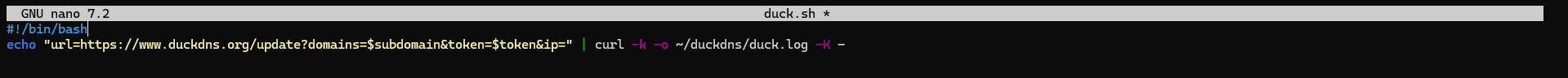
#!/bin/bash

echo "url=https://www.duckdns.org/update?domains=$subdomain&token=$token&ip=" | curl -k -o ~/duckdns/duck.log -K -

chmod 700 duck.sh

crontab -e

\*/5 \* \* \* \* ~/duckdns/duck.sh >/dev/null 2>&1



1. **Create Nginx config**

Sudo apt install nginx -y

sudo nano /etc/nginx/sites-available/mynakshtra

server {

listen 80;

server\_name mynakshtradomain.duckdns.org;

location / {

proxy\_pass http://localhost:3000;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

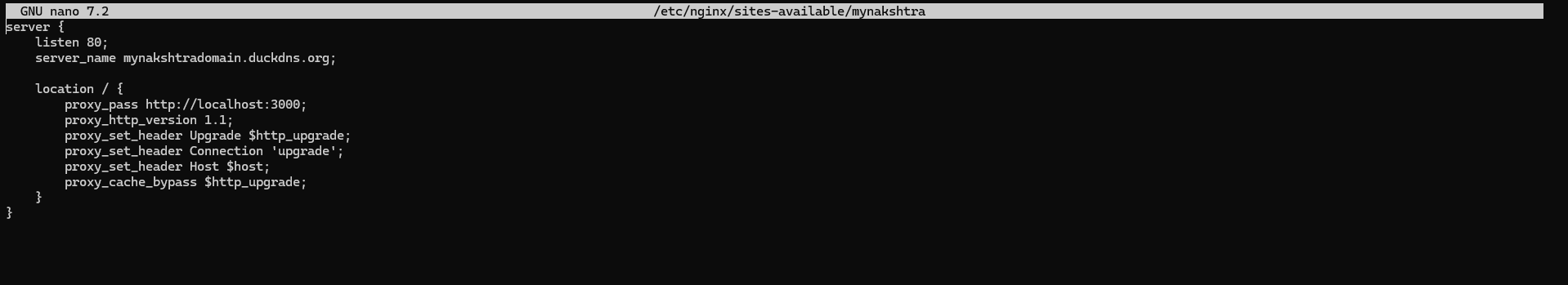
proxy\_set\_header Connection 'upgrade';

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

}

}



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

sudo nginx -t

sudo systemctl restart nginx

1. **SSL Setup**

sudo apt install certbot python3-certbot-nginx -y

sudo certbot --nginx -d mynakshtradomain.duckdns.org

sudo certbot renew --dry-run

1. **Install CloudWatch Agent**

sudo apt update

wget https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb

sudo dpkg -i amazon-cloudwatch-agent.deb

1. **Configure Agent**

sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard

sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl \

-a fetch-config \

-m ec2 \

-c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json \

-s

**3. GitHub Repository Setup**

**A. Create Repository**

Go to GitHub > Create New Repository > Enter Name > Create.

1. **Add Webhook**

Profile > Settings > Webhooks > Add Webhook

Payload URL: <your Jenkins server URL>/github-webhook/

1. **Create GitHub Access Token**

Profile > Settings > Developer Settings > Personal Access Token > Classic

Generate New Token (classic) > Select Scopes (repo, workflow) > Generate

**4. Development Server Setup (Ubuntu EC2)**

mkdir Node\_project

cd Node\_project

nano package.json

nano index.js

mkdir test

cd test/

nano test.js

sudo apt update

node --version

sudo apt install nodejs -y

npm --version

sudo apt install npm -y

Install java and jenkins and Login

git init

git remote add origin <repo\_link>

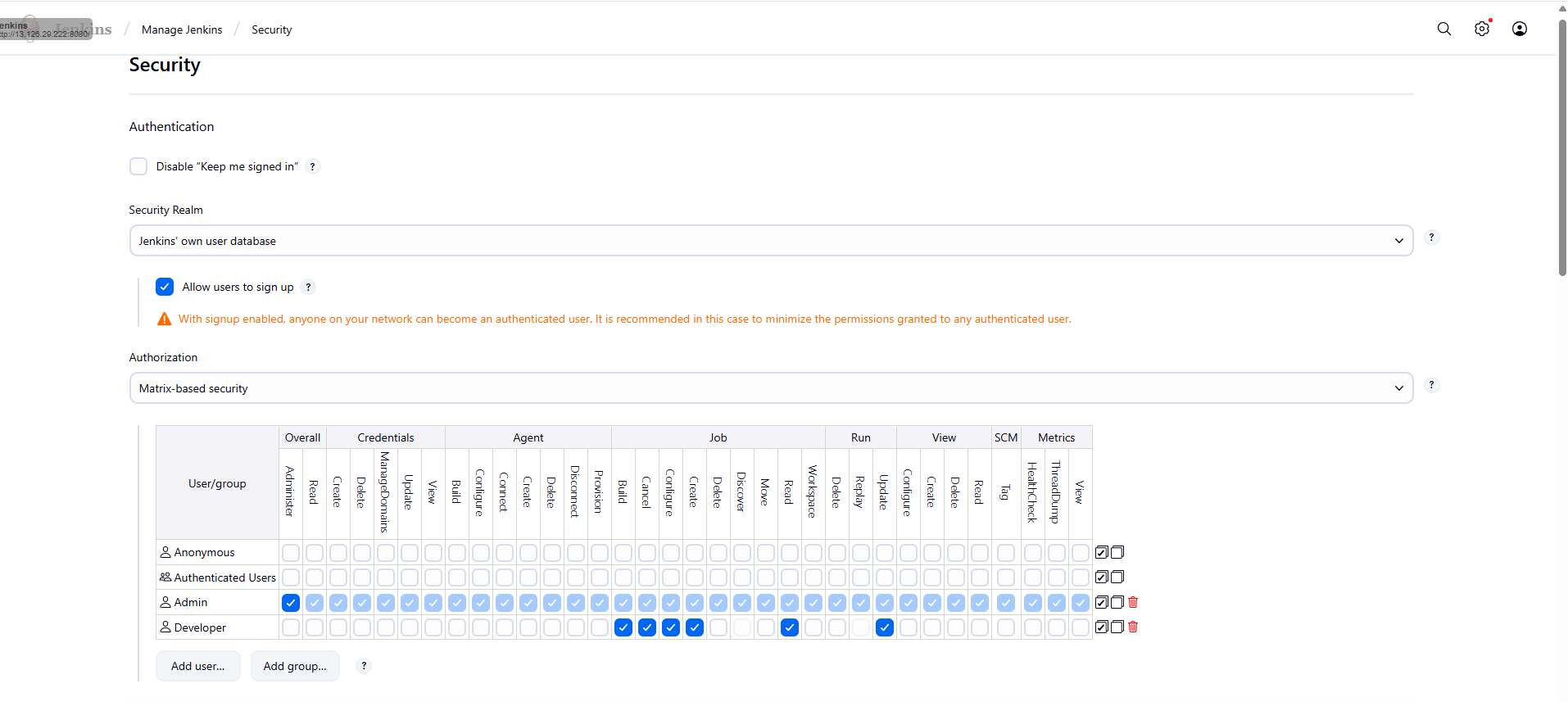
git add .

git commit -m "Initial commit"

git push origin master

**5. Jenkins Configuration**

1. **Install Jenkins NodeJs Plugin**
2. Setting > Manage Jenkins > plugin > Available Plugin > Search(nodejs) > Install
3. Setting > Manage Jenkins > Tool > NodeJS installations > Add NodeJs > Name > Version(Make Sure Match to Live server Version) > Save
4. **Install SSH Plugin**
5. Setting > Manage Jenkins > plugin > Available Plugin > Search(ssh) > Install
6. Setting > Manage Jenkins > Credentials > global > Add Credential > Kind(SSH Username with privatekey) > Username(ubuntu) > Private Key(Enter directly) > add > Create
7. Setting > Manage Jenkins > System > SSH remote hosts > Add > Hostname(Live server Public IP) > Port(22) > Credential(ubuntu) > Save
8. **Set Up Matrix-Based Security in Jenkins**
9. Setting > Manage Jenkins > User/Group > Create User/Group
10. Setting > Manage Jenkins > Security > Authorization(Allow users to sign up) > Add User/Group > Give permissions > Save



**6. Create Jenkins Freestyle Jobs**

**1.Build**

New Item > Name > Freestyle Project > Ok

Description > SCM > Trigger (GitHub hook trigger for GITScm polling) > Environment (Provide Node & npm bin/ folder to PATH) > Build Steps(Execute shell) (npm install ) > Save

**2.Test**

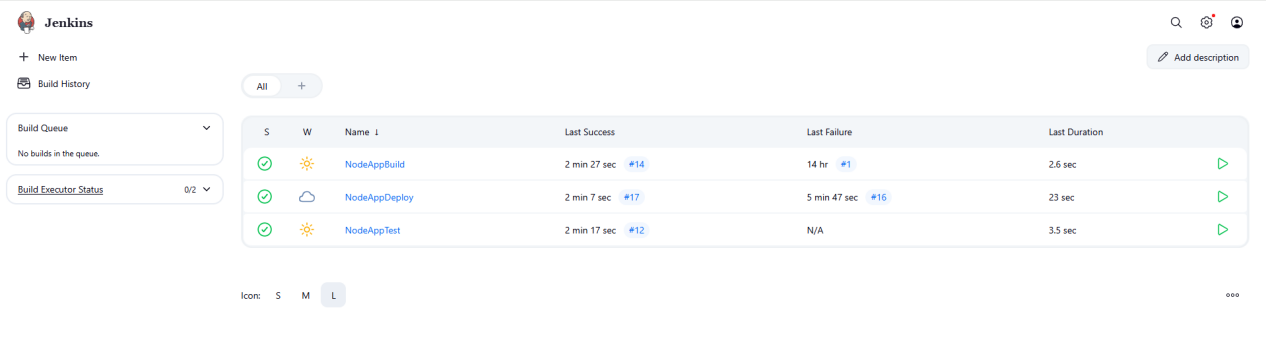
New Item > Name > Freestyle Project > Ok

Description > SCM > Trigger (Build after other projects are built) > Environment (Provide Node & npm bin/ folder to PATH) > Build Steps(Execute shell) (npm install, ./node\_modules/mocha/bin/\_mocha --exit ./test/test.js ) > Save

**3.Deploy**

New Item > Name > Freestyle Project > Ok

Description > SCM > Trigger (Build after other projects are built) > Environment (Provide Node & npm bin/ folder to PATH) > Build Steps(Execute shell script on remote host using ssh) ( git clone https://github.com/Bclouds670/nodejs-app-demo.git sudo npm install sudo npm install -g pm2 pm2 start index.js pm2 start -f index.js) > Save



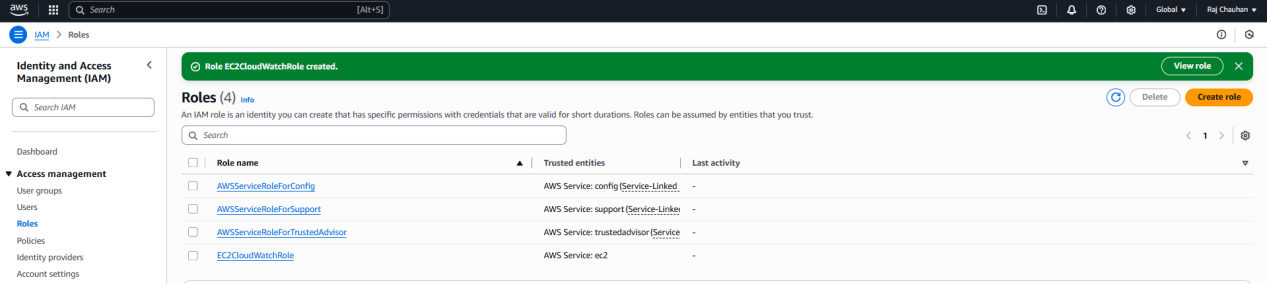
**7. CloudWatch Monitoring Setup**

1. **Create Role**

IAM ROle > Create Role > Select AWS Service(EC2) > Create > Attach policy (CloudWatchAgentServerPolicy) > Next > Name > Create Role

1. **Attach Role on you Live Ec2 Server**

Ec2 > Instances > Select your instance > Actions > Security > Modify IAM role



1. **Create CloudWatch Metrics/Dashboard**

Cloud Watch > Metric > Select CPU, memory, disk > Action > Add Dashboard

