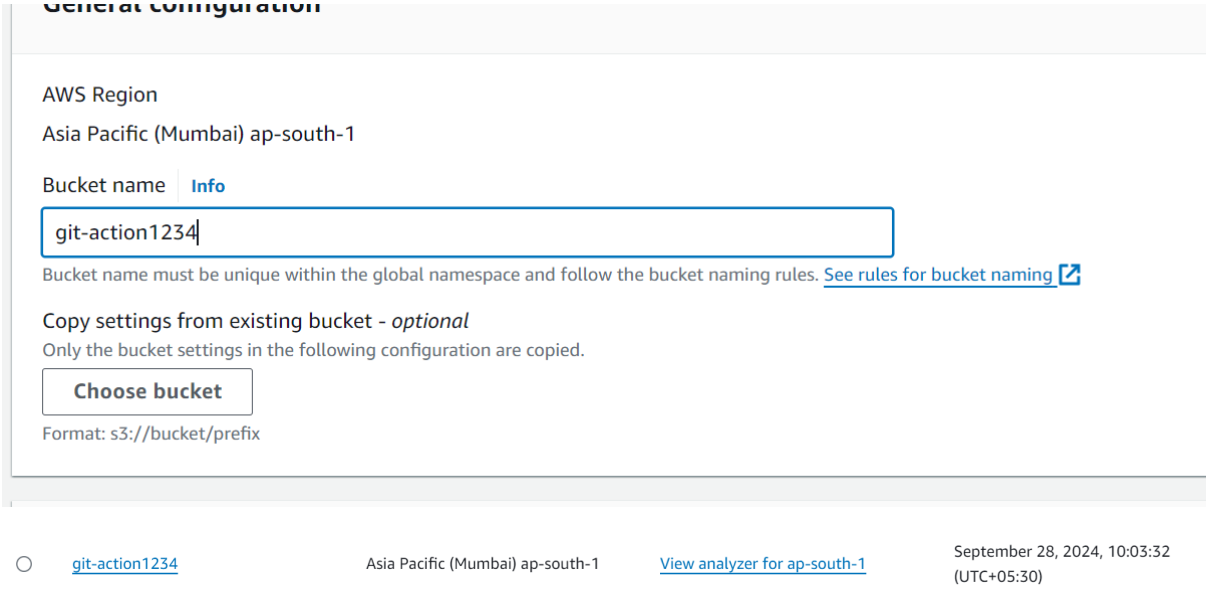
**1. Deploy a static website on AWS S3 bucket using GitHub action. when developer is making some changes in website code. New changes should be reflected on my website**

Solution:

1.Create an S3 bucket



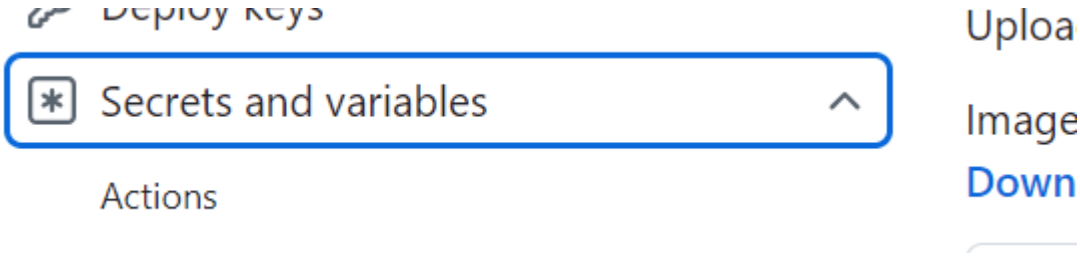

| | | | | |
|---|---|---|---|---|
| ○ | git-action1234 | Asia Pacific (Mumbai) ap-south-1 | View analyzer for ap-south-1 | September 28, 2024, 10:03:32 (UTC+05:30) |

2. Create an Github reposirty and push a normal index.html page with some content in it

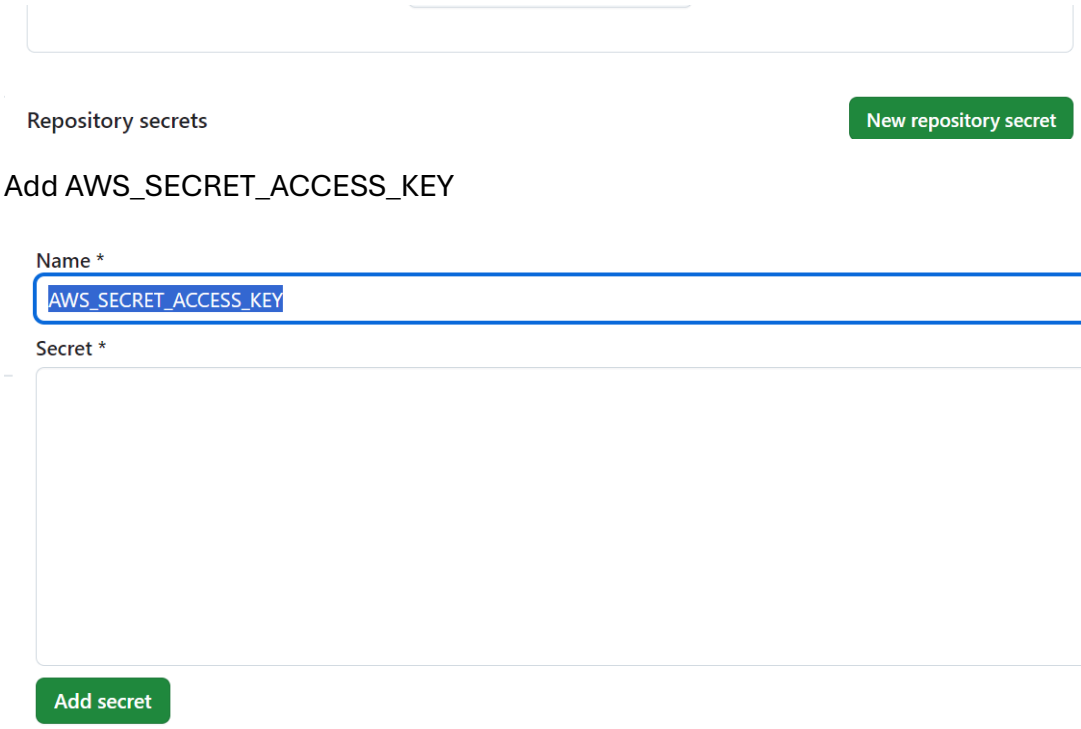
```
1    Hello this is statihhc  ss    hello
2    hii
3    hello
4    ss
5
```

a.)Go to settings of repository

Settings->Secrets&Variable->Actions




b. ) Go to new repository secrets

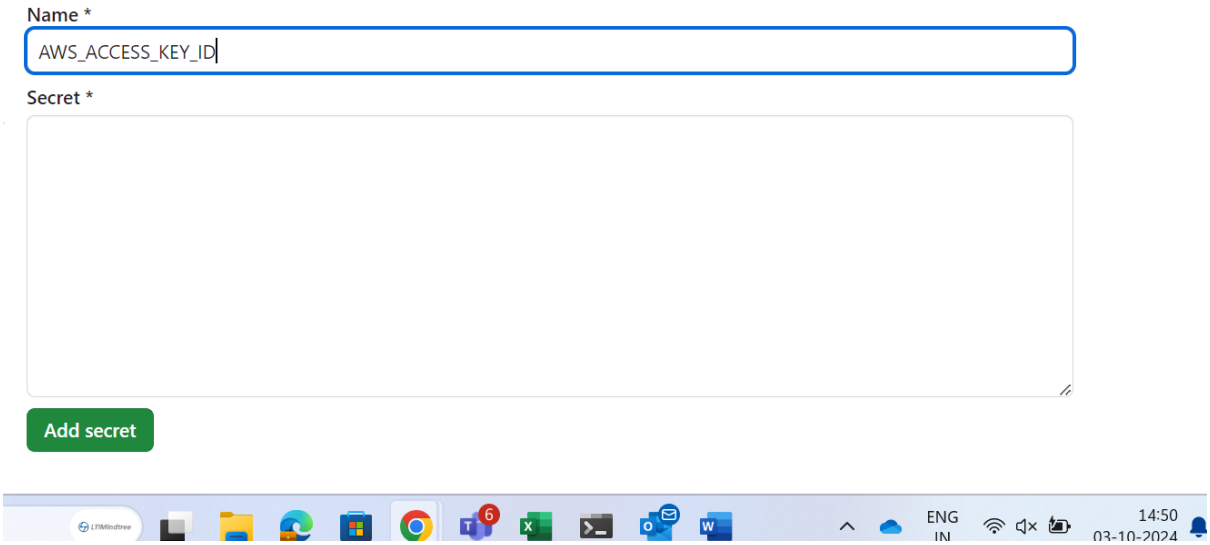



Add AWS_SECRET_ACCESS_KEY



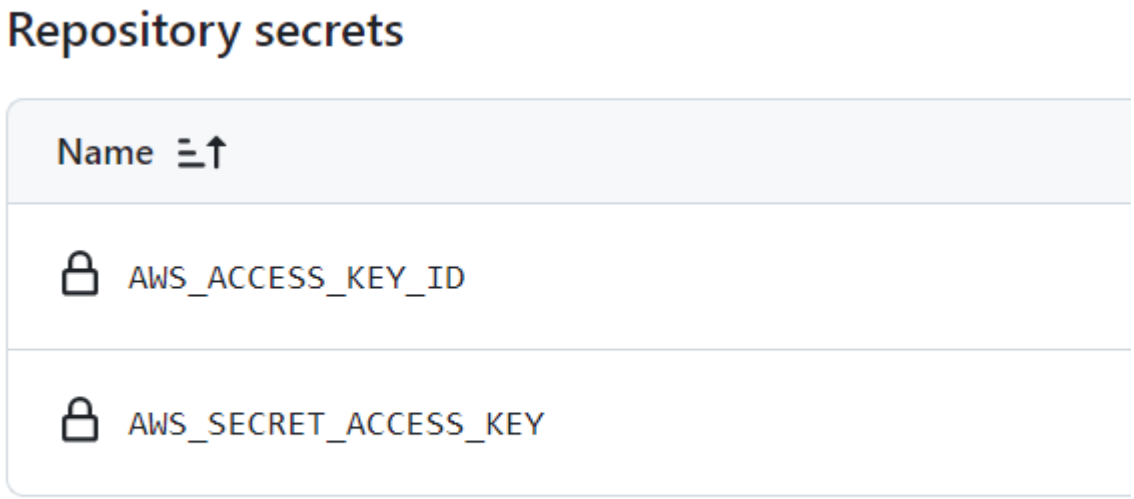

Create one more new secret variable

AWS_ACCESS_KEY_ID



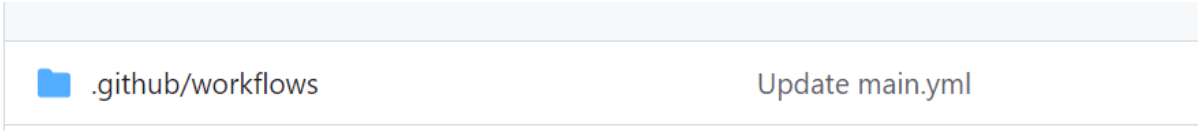

Not:Add your respective key id and secret that you get from creating an IAM user

Now you will be able to see secrets and id in you repository settings

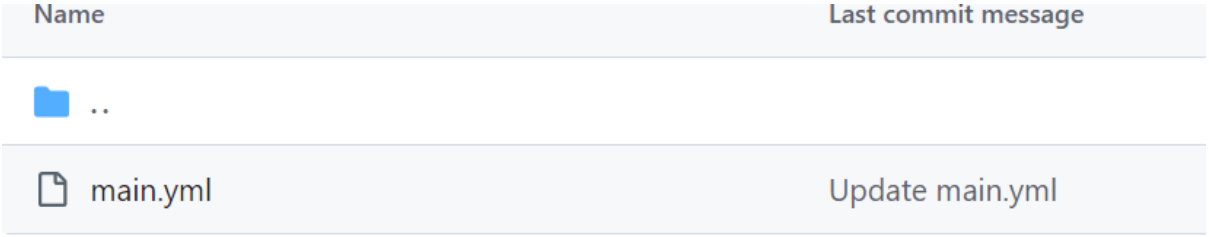



3. Now create a folder named .github and inside it workflows

This folder will consist of your workflow file which will trigger the action




Inside workflow create main.yml




Inside main.yml write  the following code

```
name: Upload Website
on:
 push:
  branches:
  - main
jobs:
 deploy:
  runs-on: ubuntu-latest
  steps:
  - name: Checkout
   uses: actions/checkout@v1
  - name: Configure AWS Credentials
   uses: aws-actions/configure-aws-credentials@v1
   with:
    aws-access-key-id: ${{ secrets.AWS_ACCESS_KEY_ID }}
    aws-secret-access-key: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
    aws-region: us-east-1
```

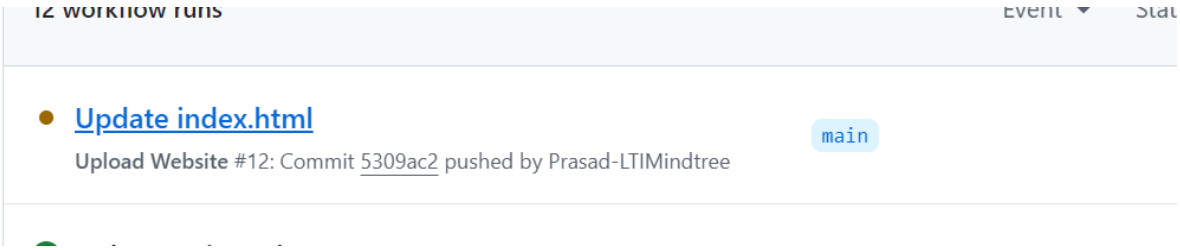
- name: Deploy static site to S3 bucket

run: aws s3 sync . s3://git-action1234 –delete

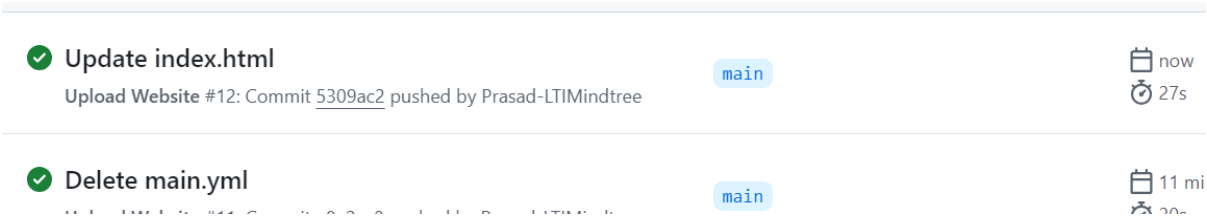
What this code will do is host your index.html on your aws s3 bucket

```yaml
name: Upload Website
on:
  push:
    branches:
    - main
jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
    - name: Checkout
      uses: actions/checkout@v1
    - name: Configure AWS Credentials
      uses: aws-actions/configure-aws-credentials@v1
      with:
        aws-access-key-id: ${{ secrets.AWS_ACCESS_KEY_ID }}
        aws-secret-access-key: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
        aws-region: us-east-1
    - name: Deploy static site to S3 bucket
      run: aws s3 sync . s3://git-action1234 --delete
```

Now commit the changes

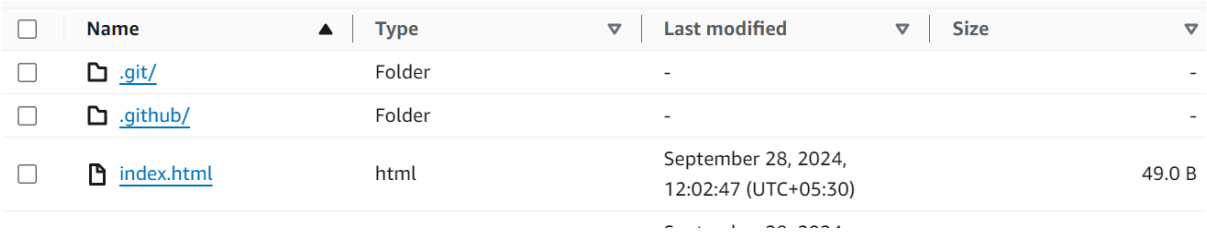


You can see the job running




The green tick has occurred that means build is usccessfull

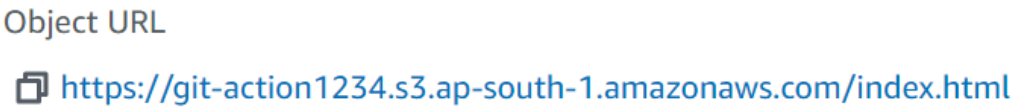
4. Go to s3 bucket



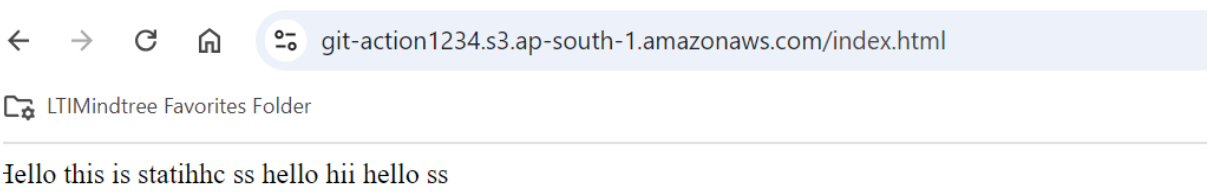
| Name | Type | Last modified | Size |
|---|---|---|---|
| .git/ | Folder | - | - |
| .github/ | Folder | - | - |
| index.html | html | September 28, 2024, 12:02:47 (UTC+05:30) | 49.0 B |

You can see the code is here

Now click on index.html




Click on its object url

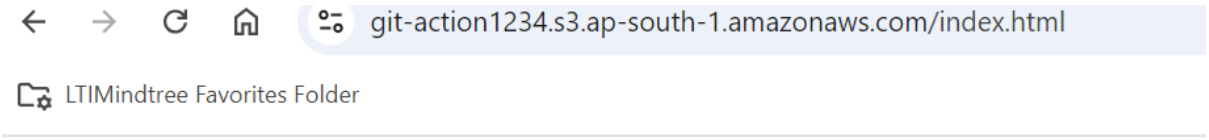



Site is visible in browser

Now lets make changes to index.html

```
5      Milestone 3
```

Added this line ->Commit the changes ->reload the url



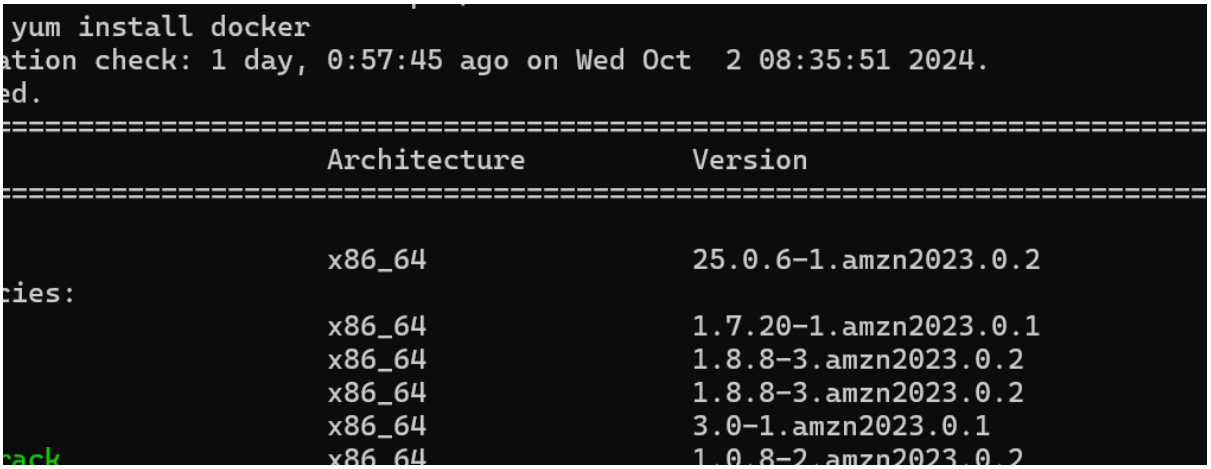

Hello this is statihhc ss hello hii hello ss Milestone 3

You can see chages are visible in the browser

**2. Pull the Ubuntu image from Docker hub and launch a web application in the container on port no. 8080 and this application should be reachable globally.**

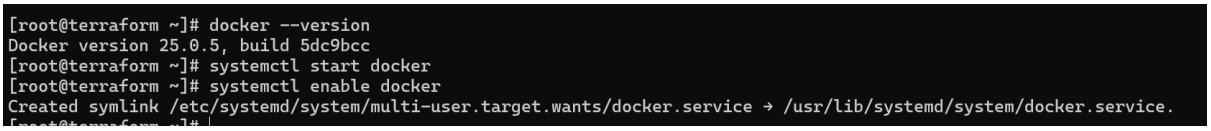
**Solution:**

1. Installing docker




We are using amazon linux here

Verify installation & start and enable docker



Docker has now started

2. Pull an ubuntu image

```
[root@docker-server ~]# docker pull ubuntu:latest
latest: Pulling from library/ubuntu
Digest: sha256:b359f1067efa76f37863778f7b6d0e8d911e3ee8efa807ad01fbf5dc1ef9006b
Status: Image is up to date for ubuntu:latest
docker.io/library/ubuntu:latest
```

3. run the image on 8080

```
[root@docker-server ~]# docker run -it -p 8080:8080 ubuntu
root@4864bd16392a:/#
```

Now we are inside the ubunut image

4. now first update the apt and install apache (for simple web )
   apt-get update
   apt-get install apache2

```
root@c2e5460f312c:/# apt-get update -y
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:3 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [446
Get:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [477 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [367 kB
Get:9 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [19.3 MB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [13.
Get:11 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages [117 kB]
Get:12 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages [1808 kB]
Get:13 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [507 kB]
Get:14 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [678 kB]
Get:15 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [446 k
Get:16 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [17.8
Get:17 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [11.8
Fetched 25.2 MB in 4s (5658 kB/s)
Reading package lists... Done
root@c2e5460f312c:/# apt-get install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  adduser apache2-bin apache2-data apache2-utils ca-certificates krb5-locales libapr1t
  libaprutil1t64 libbrotli1 libcurl4t64 libexpat1 libgdbm-compat4t64 libgdbm6t64 libgs
  libkeyutils1 libkrb5-3 libkrb5support0 libldap-common libldap2 liblua5.4-0 libnghttp
  libsasl2-2 libsasl2-modules libsasl2-modules-db libsqlite3-0 libssh-4 libxml2 media-
```

Now you need to install vim as it is not built in

apt-get update -y

apt-get install vim

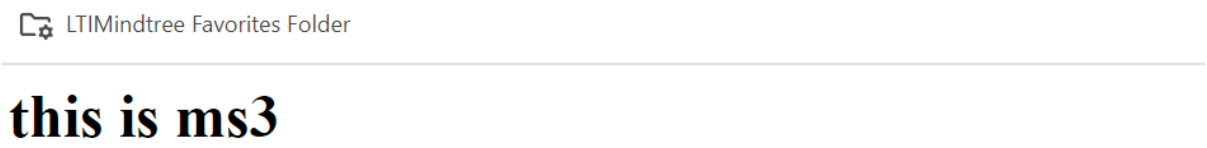
5. go to cd /var/www/html

```
root@51564c9d554d:/# cd /var/www/html
root@51564c9d554d:/var/www/html# vim index.html
```

```
<h1> this is ms3 </h1>
~
~
~
```

Save index.html
Service apache2 start
Service apache2 enable

Go to you instane public ip and paste it in browser




Successfully configured the website