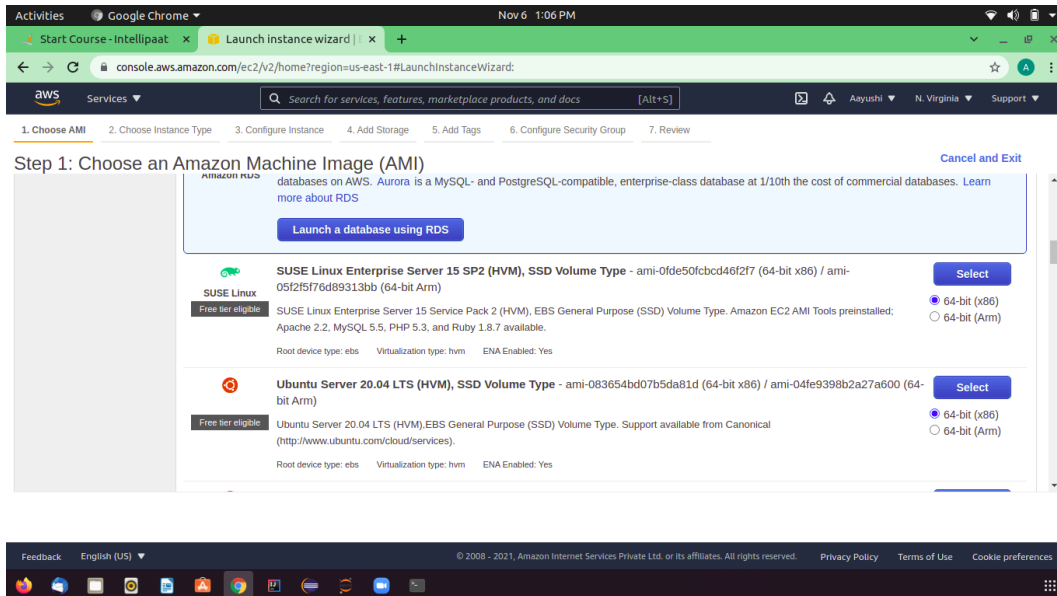


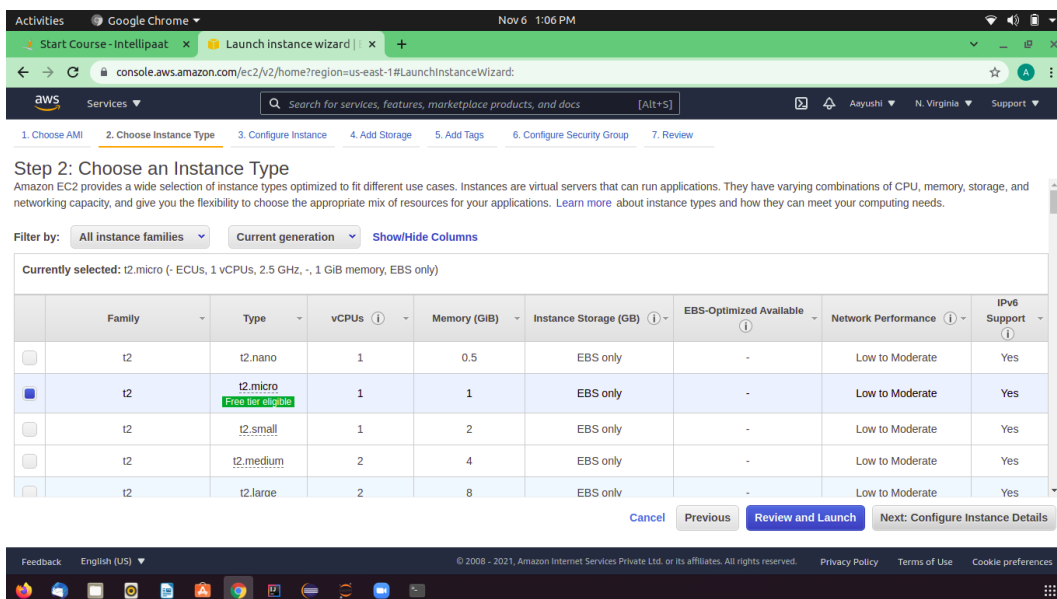
# EC2 Assignment 1:

Create an instance in us-east-1a(N.Virginia) with an Ubuntu OS with website Hello World

Create an instance:



Choose t2.micro as instance type



## Configure the Instance Details

### Select us-east-1a

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances  [Launch into Auto Scaling Group](#)

Purchasing option ☐ Request Spot instances

Network  [Create new VPC](#)

Subnet  [Create new subnet](#)  
4091 IP Addresses available

Auto-assign Public IP

Placement group ☐ Add instance to placement group

Capacity Reservation

Domain join directory  [Create new directory](#)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

## Add security group

Step 6: Configure Security Group

Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group  
☐ Select an existing security group

Security group name:

Description:

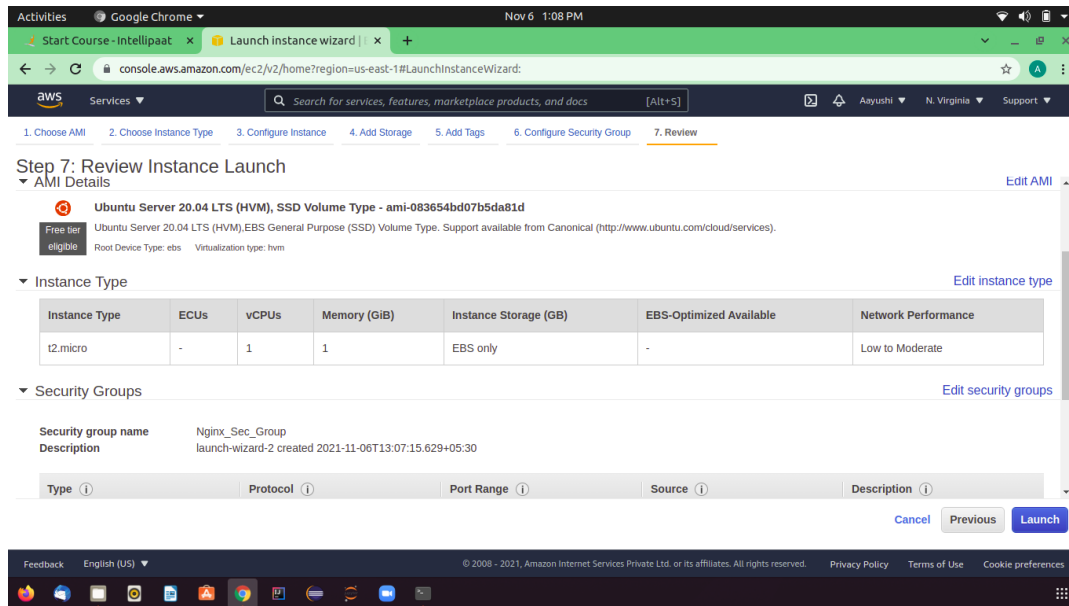
Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Anywhere 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

[Add Rule](#)

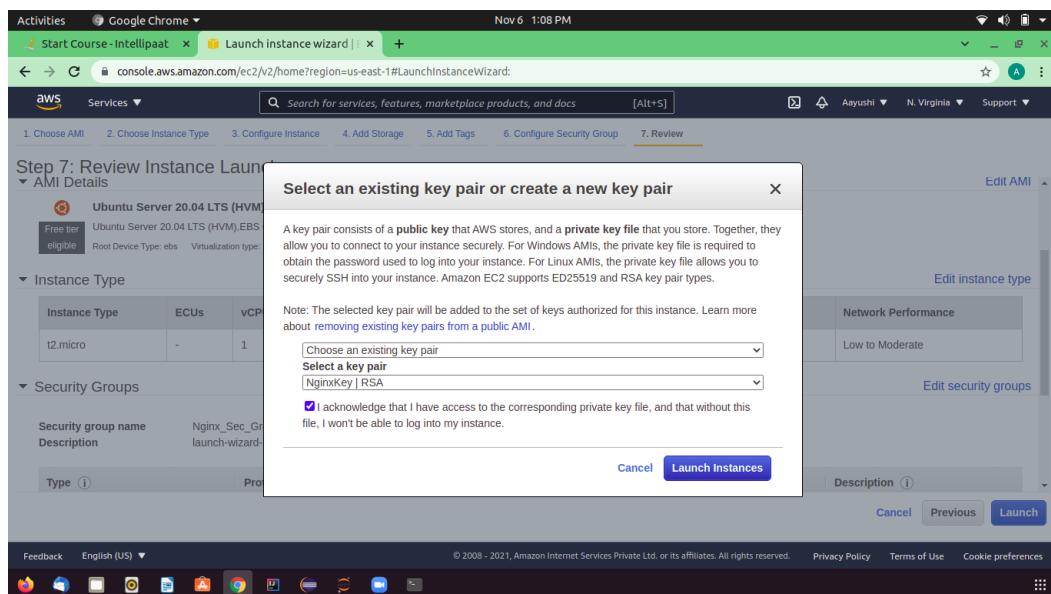
**Warning**  
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Previous](#) [Review and Launch](#)

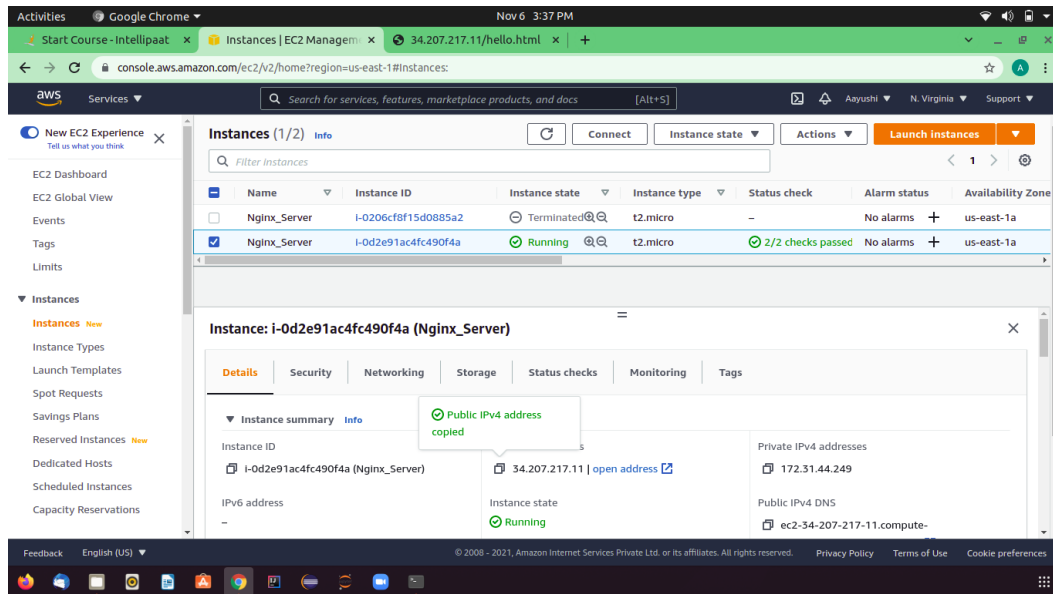
## Now take review



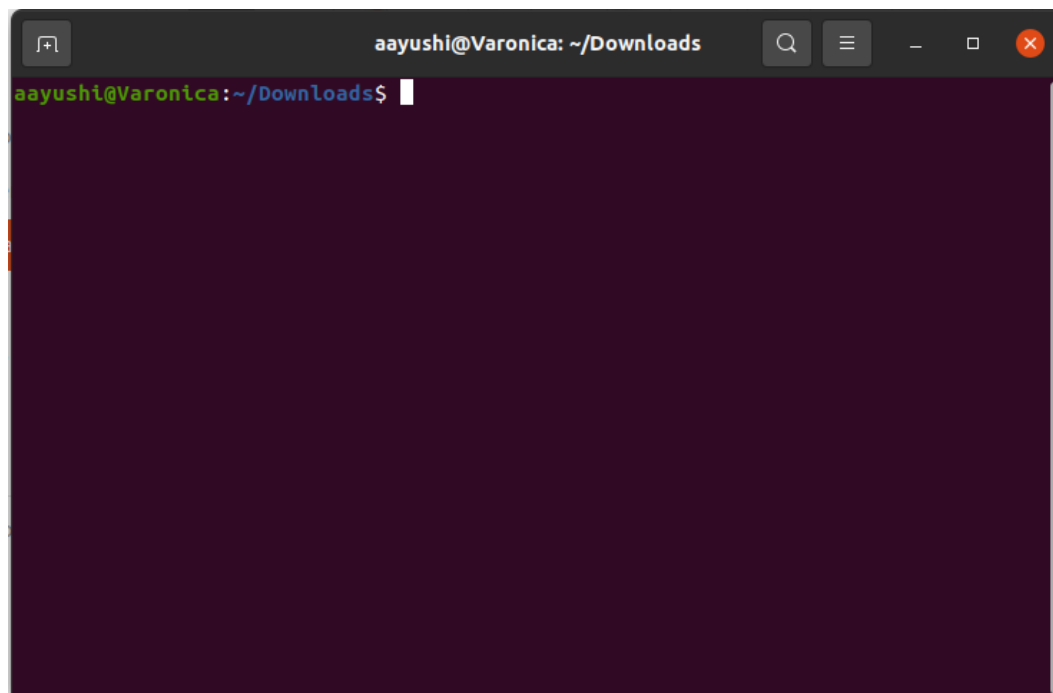
Create key pair and download key



Launch the instance



Go to the command prompt



Now go to the root using sudo su command



Now get the permission of private key  
Using command `chmod 400 Key name.pem`

```
$# chmod 400 NginxKey.pem
```

This is the explanation of the SSH command:

- `ssh`: Command to use SSH protocol
- `-i`: Flag that specifies an alternate identification file to use for public key authentication.
- `username`: Username that uses your instance
- `ip-address`: IP address given to your instance

Using command `ssh -i key name.pem username@Public-ip-address`

```
root@Varonica:/home/aayushi/Downloads# ssh -i "NginxKey.pem" ubuntu
@ec2-34-207-217-11.compute-1.amazonaws.com
The authenticity of host 'ec2-34-207-217-11.compute-1.amazonaws.com
(34.207.217.11)' can't be established.
ECDSA key fingerprint is SHA256:zwCEEKp5DCEEFpJgYgvf8+ffBeUTdnk/B7f
SoCOzMUY.
Are you sure you want to continue connecting (yes/no/[fingerprint])
? yes
Warning: Permanently added 'ec2-34-207-217-11.compute-1.amazonaws.c
om,34.207.217.11' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1020-aws x86_64)

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

System information as of Sat Nov  6 09:55:51 UTC 2021

System load:  0.21                       Processes:           101
Usage of /:   17.7% of 7.69GB             Users logged in:    0
```

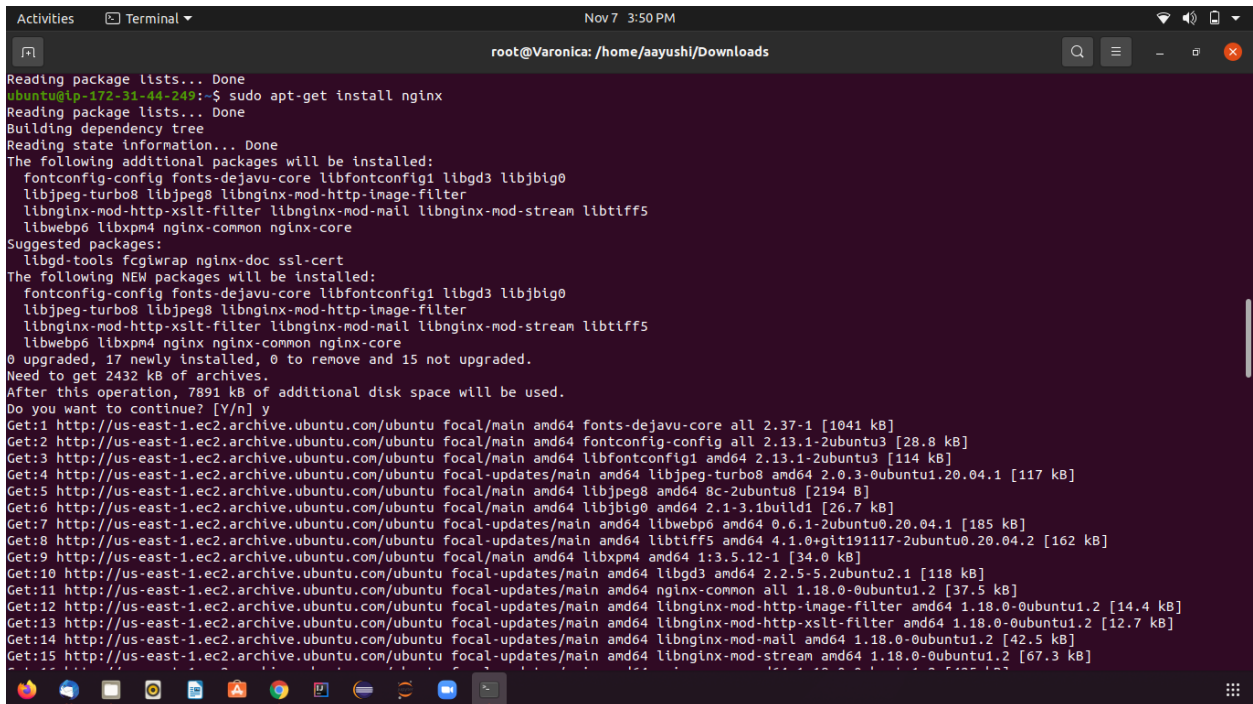
Install the Nginx

Using command: `sudo apt-get update`

`Sudo apt-get install nginx`

```
:~$ sudo apt-get update
```

```
. Done
~$ sudo apt-get install nginx
```



```
Reading package lists... Done
ubuntu@ip-172-31-44-249:~$ sudo apt-get install nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core libfontconfig1 libgd3 libjpeg8
  libjpeg-turbo8 libpng16-0 libnginx-mod-http-image-filter
  libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream libtiff5
  libwebp6 libxpm4 nginx-common nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  fontconfig-config fonts-dejavu-core libfontconfig1 libgd3 libjpeg8
  libjpeg-turbo8 libpng16-0 libnginx-mod-http-image-filter
  libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream libtiff5
  libwebp6 libxpm4 nginx-common nginx-core
0 upgraded, 17 newly installed, 0 to remove and 15 not upgraded.
Need to get 2432 kB of archives.
After this operation, 7891 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 fonts-dejavu-core all 2.37-1 [1041 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 fontconfig-config all 2.13.1-2ubuntu3 [28.8 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libfontconfig1 amd64 2.13.1-2ubuntu3 [114 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libjpeg-turbo8 amd64 2.0.3-0ubuntu1.20.04.1 [117 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libjpeg8 amd64 8c-2ubuntu8 [2194 B]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libgd3 amd64 2.1.3-1build1 [26.7 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libwebp6 amd64 0.6.1-2ubuntu0.20.04.1 [185 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libtiff5 amd64 4.1.0+git191117-2ubuntu0.20.04.2 [162 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libxpm4 amd64 1:3.5.12-1 [34.0 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libgd3 amd64 2.2.5-5.2ubuntu2.1 [118 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 nginx-common all 1.18.0-0ubuntu1.2 [37.5 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libnginx-mod-http-image-filter amd64 1.18.0-0ubuntu1.2 [14.4 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libnginx-mod-http-xslt-filter amd64 1.18.0-0ubuntu1.2 [12.7 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libnginx-mod-mail amd64 1.18.0-0ubuntu1.2 [42.5 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libnginx-mod-stream amd64 1.18.0-0ubuntu1.2 [67.3 kB]
```

Adjust the firewall

By using: `sudo ufw app list`

```
ubuntu@ip-172-31-44-249:~$ sudo ufw app list
Available applications:
  Nginx Full
  Nginx HTTP
  Nginx HTTPS
  OpenSSH
```

`sudo ufw allow 'Nginx HTTP'`

```
ubuntu@ip-172-31-44-249:~$ sudo ufw allow 'Nginx HTTP'
Rules updated
Rules updated (v6)
```

Checking the web server

Using command: `systemctl status nginx`

```
:-$ systemctl status nginx
```

```
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2021-11-06 07:49:27 UTC; 1min 25s ago
     Docs: man:nginx(8)
  Main PID: 1954 (nginx)
    Tasks: 2 (limit: 1154)
   Memory: 4.9M
    CGroup: /system.slice/nginx.service
            1954 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
            -1955 nginx: worker process
```

```
ubuntu@ip-172-31-44-249:~$ curl -4 icanhazip.com
34.207.217.11
```

When you have your server's IP address, enter it into your browser's address bar:

[http://your\\_server\\_ip](http://your_server_ip)

## 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](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

Setting up server blocks

Create the directory for as follows, using the `-p` flag to create any necessary parent directories:

```
sudo mkdir -p /var/www/html
```

```
:-$ sudo mkdir -p /var/www/html
```

Next, assign ownership of the directory with the `$USER` environment variable:

```
sudo chown -R $USER:$USER /var/www/html
```

```
:-$ sudo chown -R $USER:$USER /var/www/html
```

```
sudo chmod -R 755 /var/www/html
```

Next, create a file `hello.html` page using `vi`:

```
vi /var/www/html/hello.html
```

Inside, add the following sample of html:

In order for Nginx to serve this content, it's necessary to create a server block with the correct directives. Instead of modifying the default configuration file directly, let's make a new one at: `sudo nano /etc/nginx/sites-available/html`



```
:~$ sudo vi /etc/nginx/sites-available/html
:~$
```

Paste in the following configuration block, which is similar to the default, but updated for our new directory and domain name:

```
server
{
    listen 80;
    listen [::]:80;

    root /var/www/html;
    hello hello.html hello.htm hello.nginx-debian.html;

    server_name html www.html;

    location /
    {
        try_files $uri $uri/ =404;
    }
}
```

```
sudo ln -s /etc/nginx/sites-available/html /etc/nginx/sites-enabled/
```

```
:~$
:~$ sudo ln -s /etc/nginx/sites-available/html /etc/nginx/sites-enabled/
```

```
sudo vi /etc/nginx/nginx.conf
```

```
:~$
:~$ sudo vi /etc/nginx/nginx.conf
```

Find the `server_names_hash_bucket_size` directive and remove the `#` symbol to uncomment the line. If you are using nano, you can quickly search for words in the file by pressing `CTRL` and `w`.

Next, test to make sure that there are no syntax errors in any of your Nginx files:

```
sudo nginx -t
```

```
ubuntu@ip-172-31-44-249:~$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

If there aren't any problems, restart Nginx to enable your changes:  
`sudo systemctl restart nginx`

```
ubuntu@ip-172-31-44-249:~$  
ubuntu@ip-172-31-44-249:~$ sudo systemctl restart nginx  
ubuntu@ip-172-31-44-249:~$
```

Goto the Web server

<http://34.207.217.11/hello.html>

