

WEB SERVER [NGINX]

Class Notes

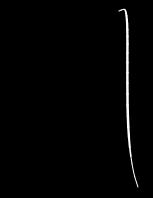
Date: March 5, 2022

By

Amarnath, Ph.D

Internet

www



Fibre optics

Backbone networks

Intranet

within Campus

Small scope

LAN

WAN

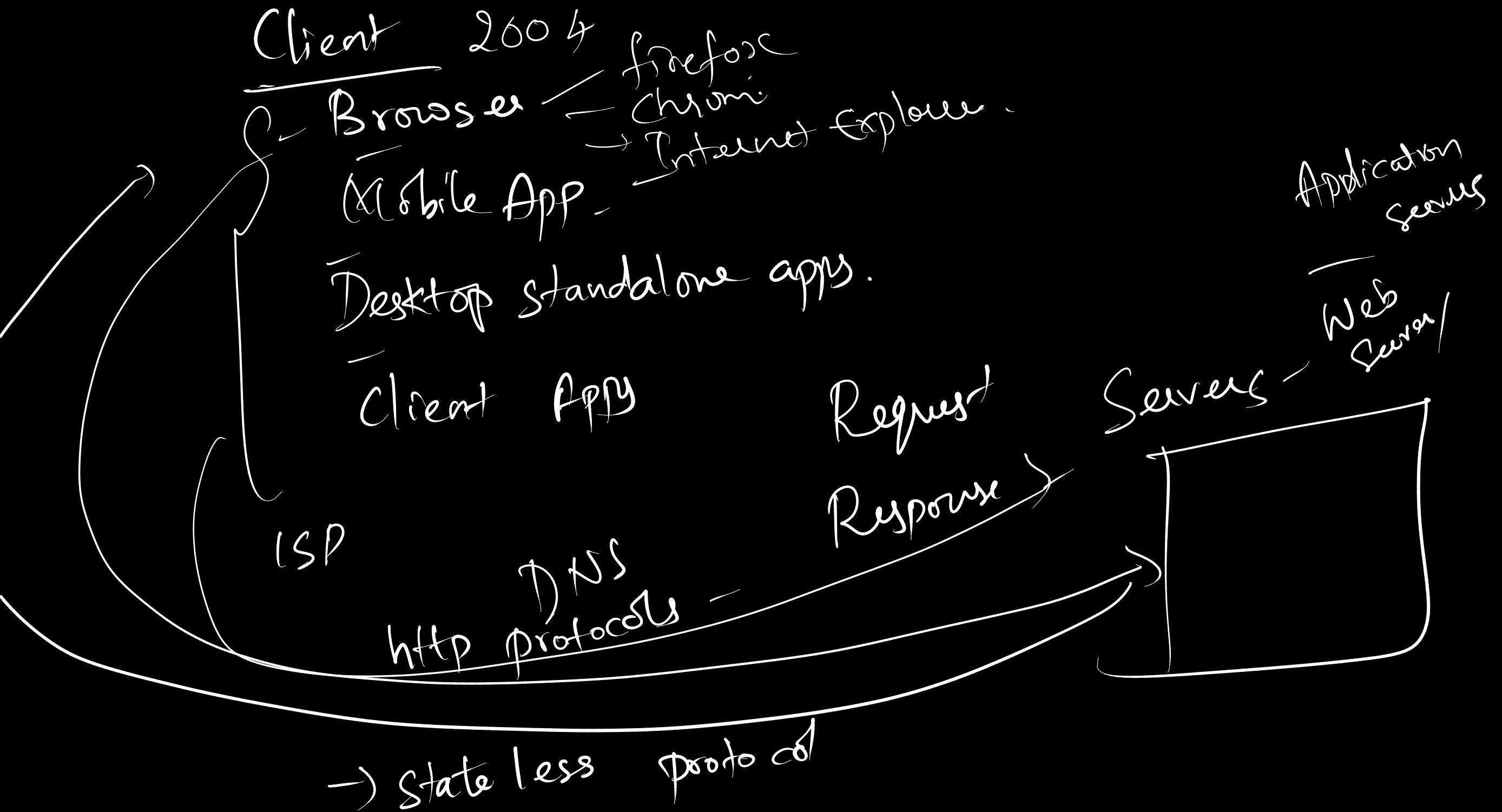


google.com

DNS
www.google.com

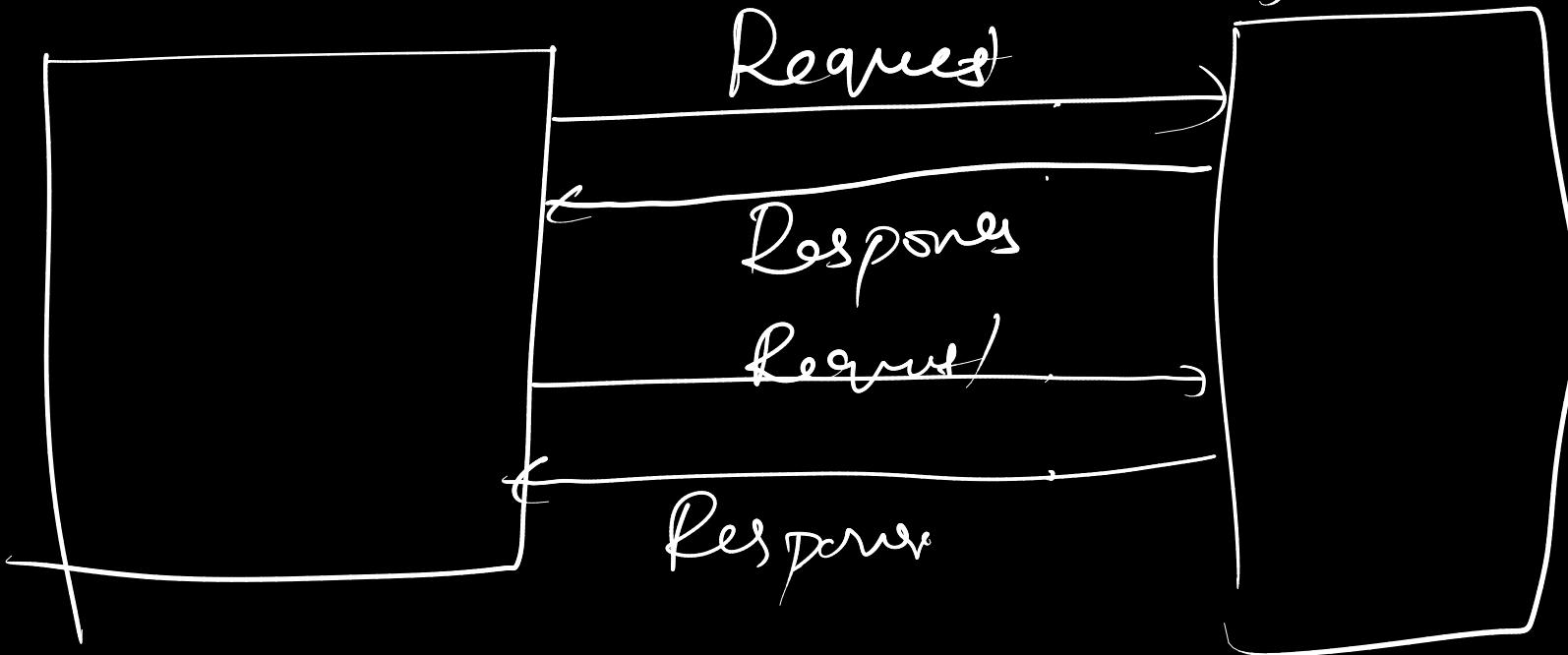
ISP

google.com



chrome browser

google.com



Independent
Request &
Responses.

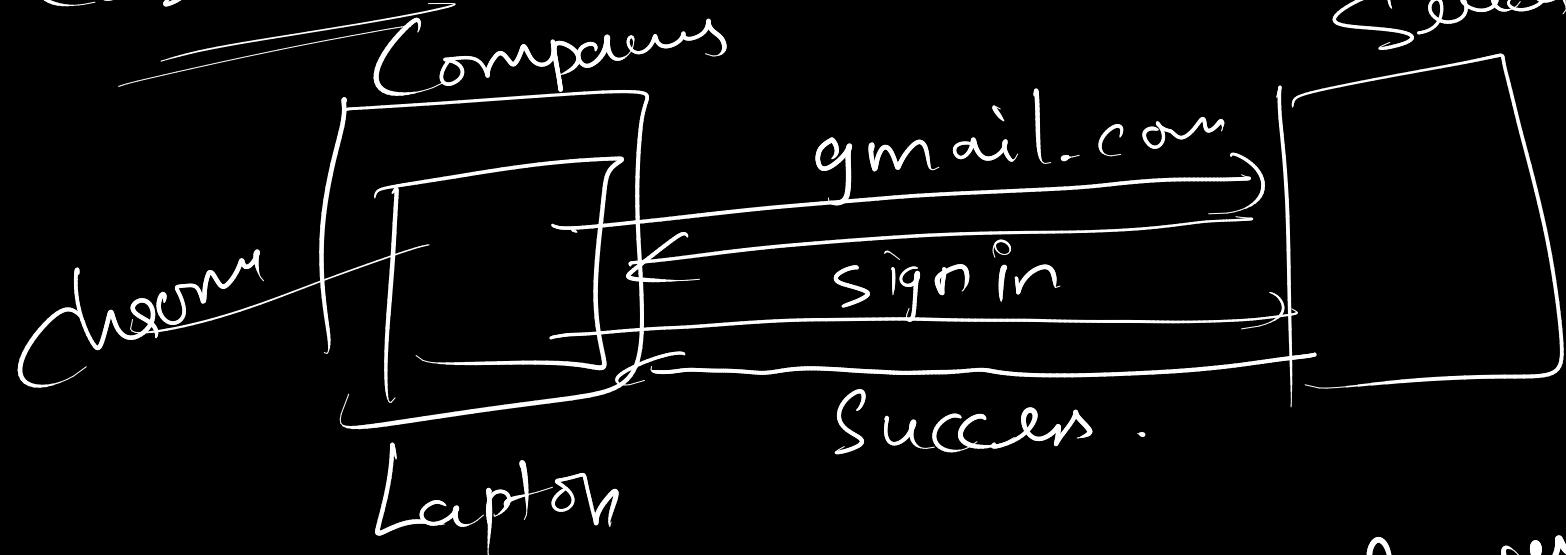
↓ http
apache.org

200 - success

16 request &
response

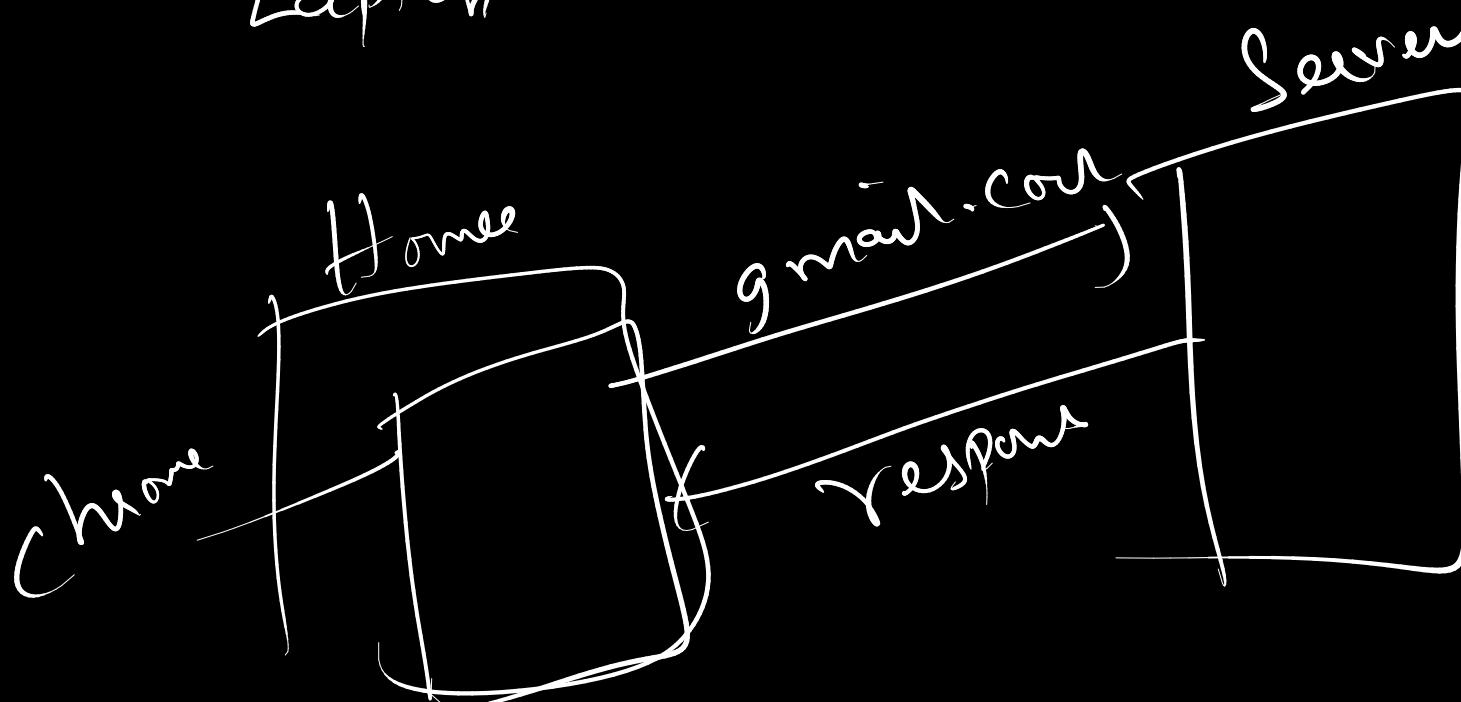
404 - Page Not foun.

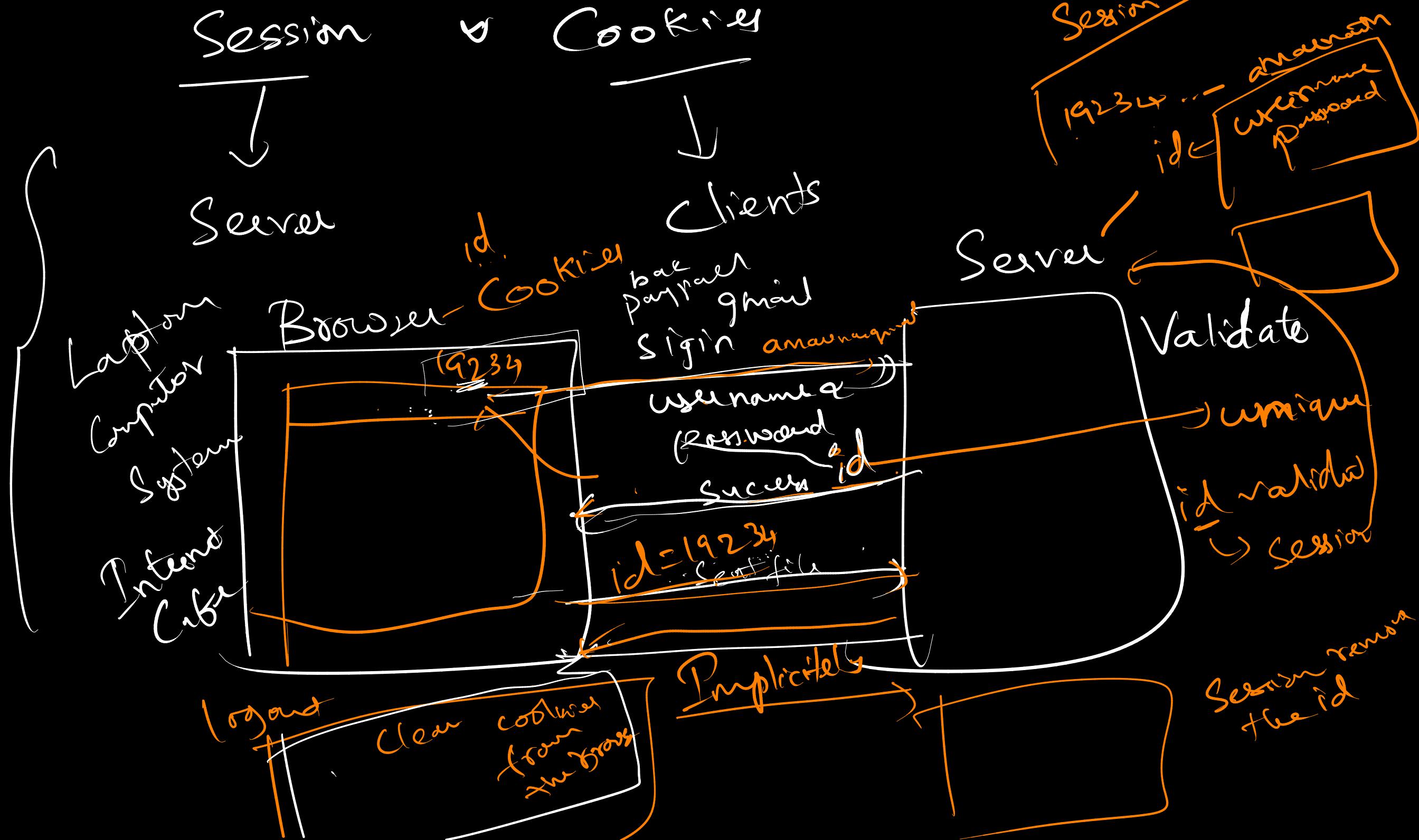
Case Studies

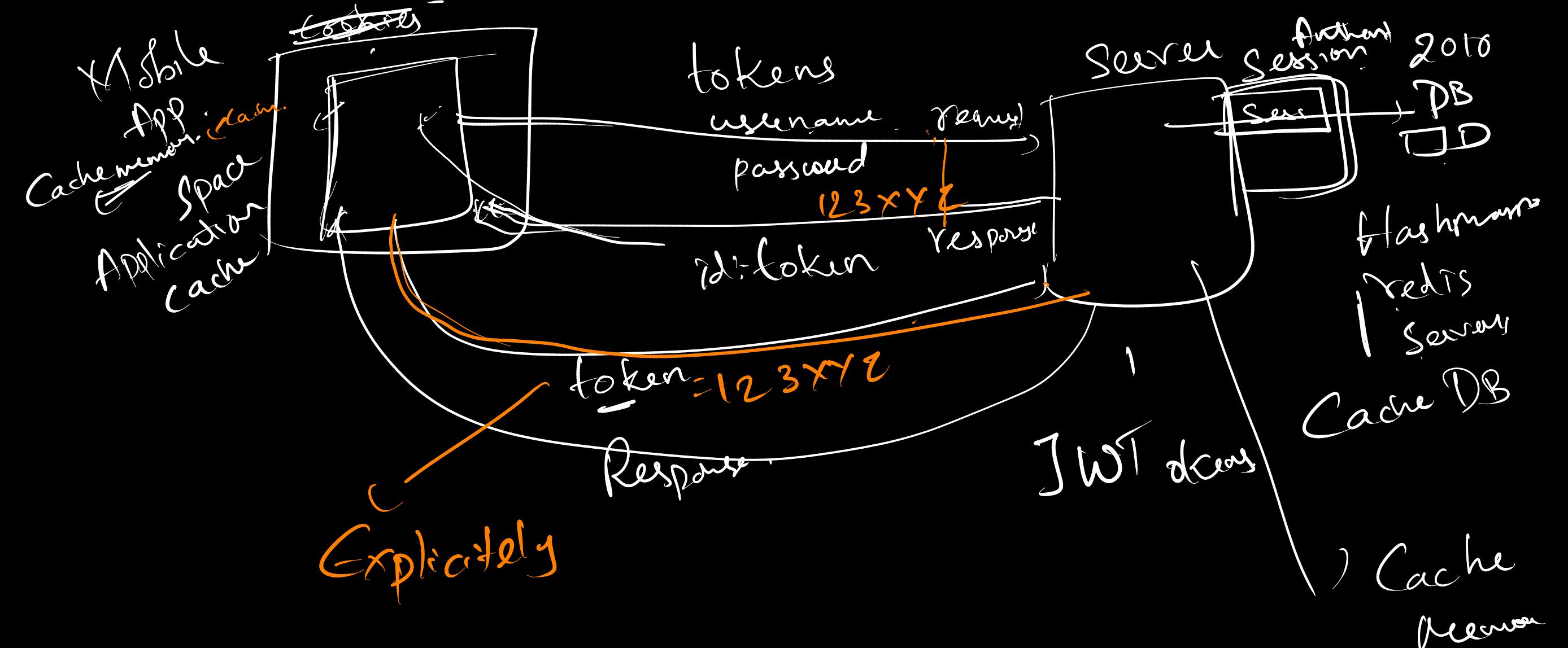


1. Cache files
2. Sessions
3. Cookies

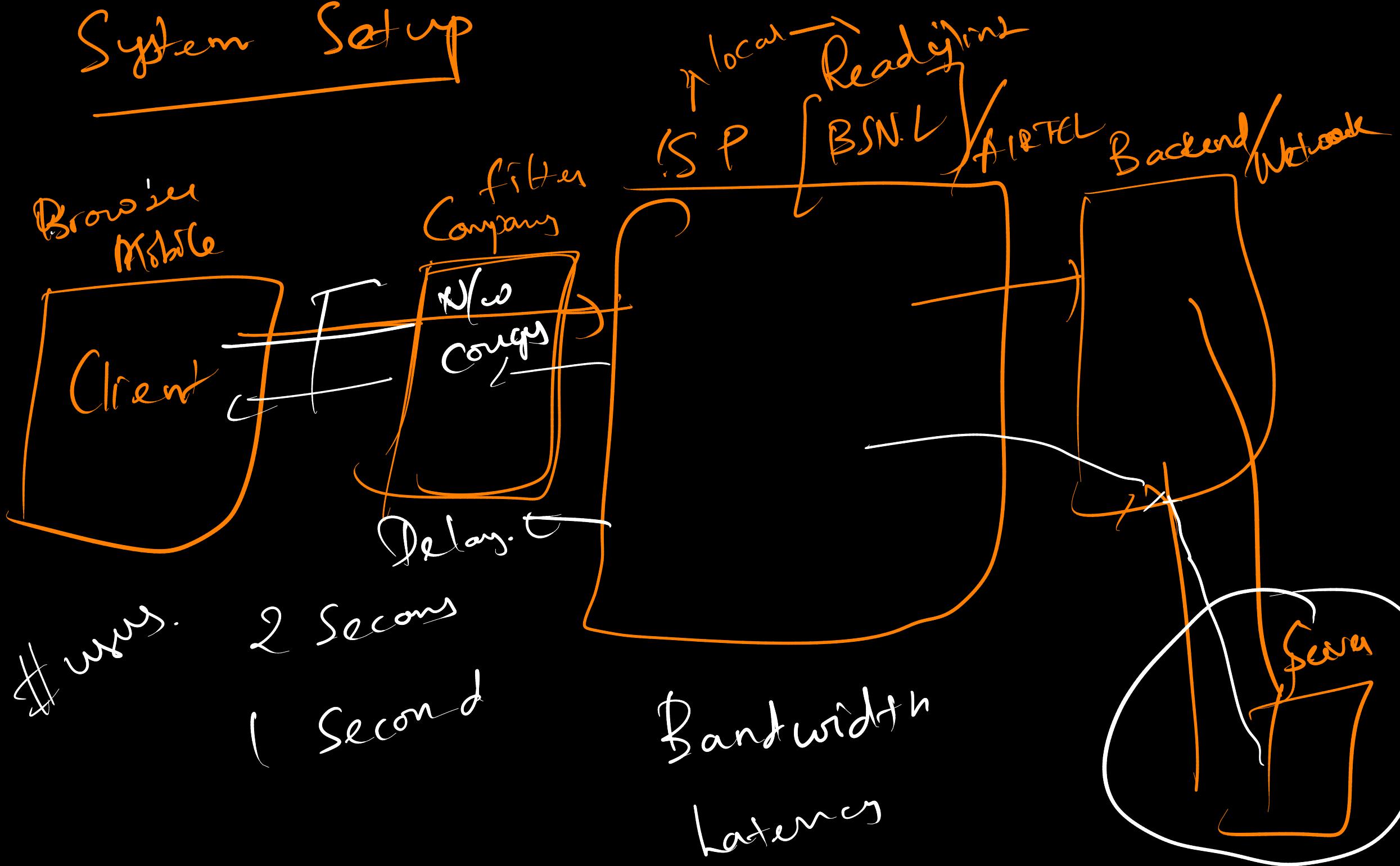
Never ASK for
password.







System Setup



Servers

→ Response to the client

→ HTML

→ image

→ PDF

→ document

→ resp

→ XML

→ JSON

Web Server

↓

Static Data

Apache 2

Nginx

ISS

Application Server

↓

↓

Dynamic Data

Tomcat

IBM server

Amazon AWS

Web Server

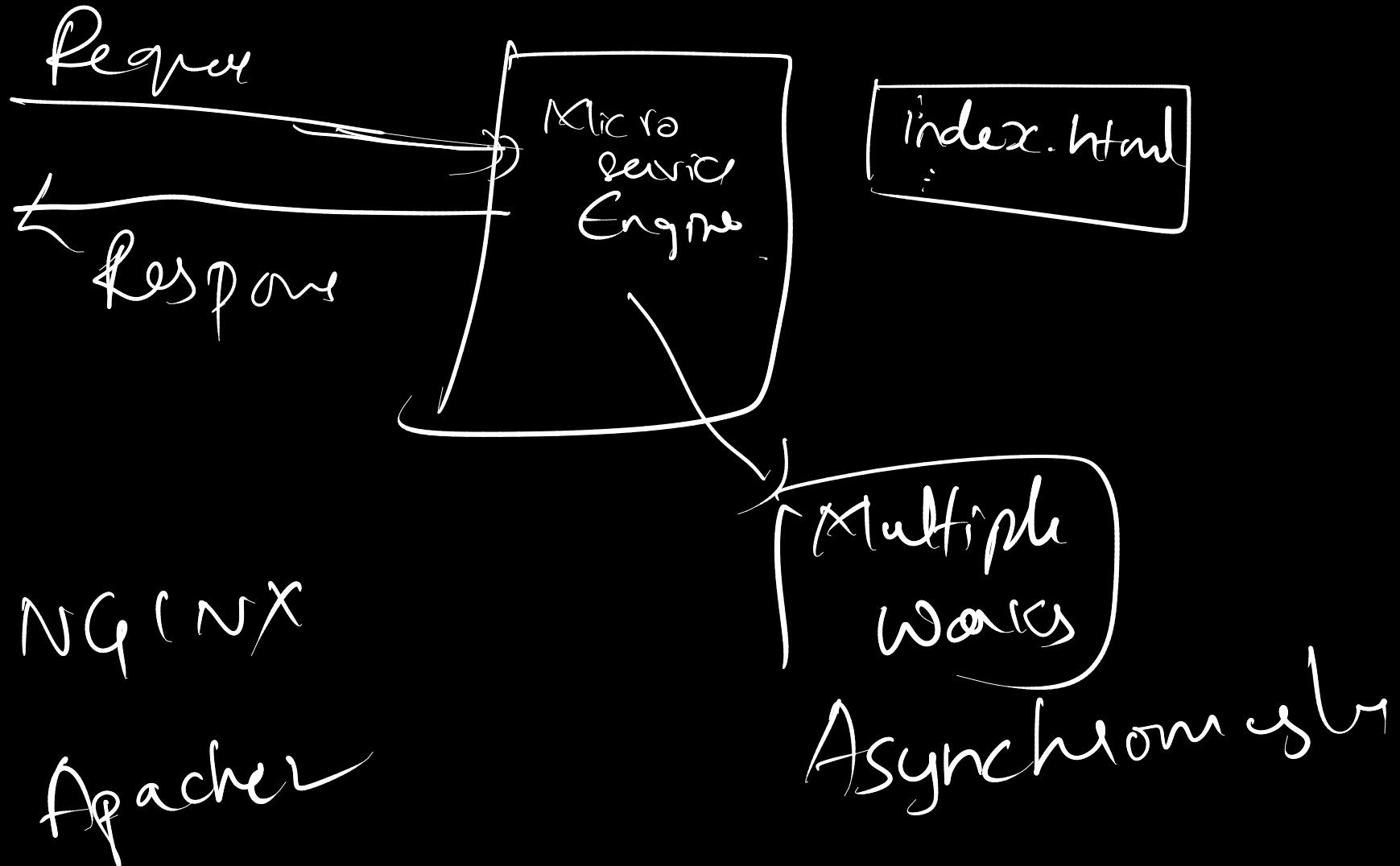
Apache 2 ✓ Containers

NGINX

Application serv.

Dynamic response

Websense



→ VM → Ubuntu

→ Amazon AWS → EC2 → Launch Instance
↓
Ubuntu

→ Pem Key -

→ Permission 400.

→ Open Run Command .

ssh -i epam2.pem ubuntu@
13.233.106.73

→ sudo apt update

sudo apt install nginx

```
# getting inside  
$ cd /etc/nginx/conf.d/  
# create a new configuration file  
$ sudo nano amar.local.conf
```

```
server{  
    listen 8000 default_server;  
    index index.html;  
    server_name amar.local;  
    root /var/www/amar.local;  
}
```

```
# create a folder  
$ cd /var/www
```

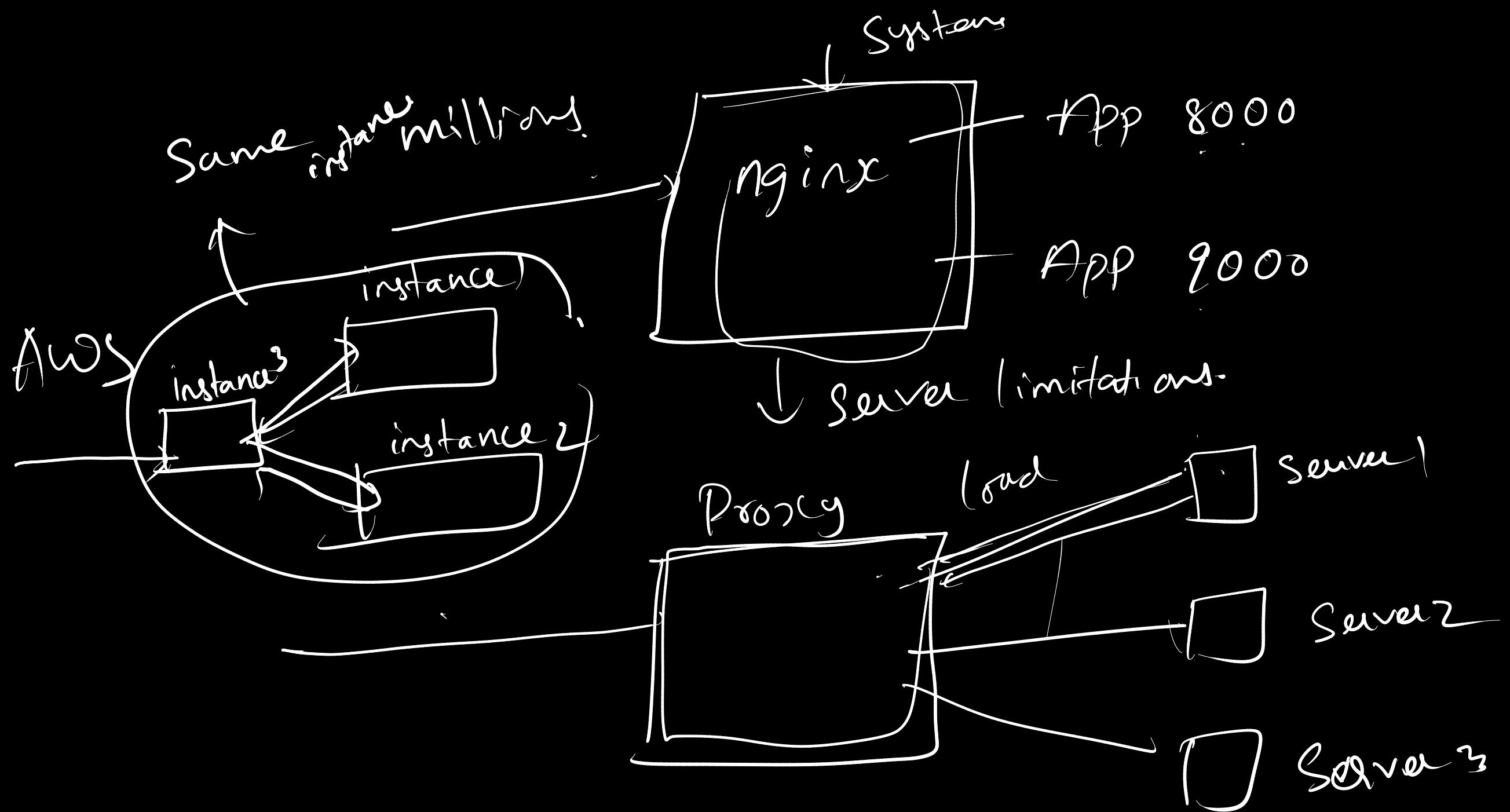
```
#  
$ sudo mkdir amar.local
```

To check the configuration file is correct
\$ sudo nginx -t
navigate inside the folder
\$ sudo nano index.html

\$ sudo systemctl restart nginx

open the browser

http://public_ip



```
sudo apt install nginx
cd /etc/nginx/conf.d/
sudo nano amar.local.conf
```

Past:

```
server{
  listen 8000 default_server;
  index index.html;
  server_name amar.local;
  root /var/www/amar.local;
}
```

ctl+x (save and exit)

```
cd /var/www/
  mkdir amar.local
```

```
cd amar.local
```

```
sudo nano index.html
```

Type hello world

ctl+x (save and exit)

```
sudo nginx -t
```

```
sudo systemctl restart nginx
```

```
cd /etc/nginx/sites-available/
```

```
sudo cp default default.bak
```

sudo nano default

comment all the lines

Paste below:

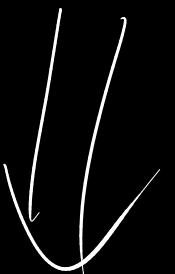
```
upstream backend{
  server ip_address:port;
  server ip_address:port;
}
server{
```

```
  listen:80;
  location / {
    proxy_pass http://backend;
  }
}
```

sudo systemctl restart nginx

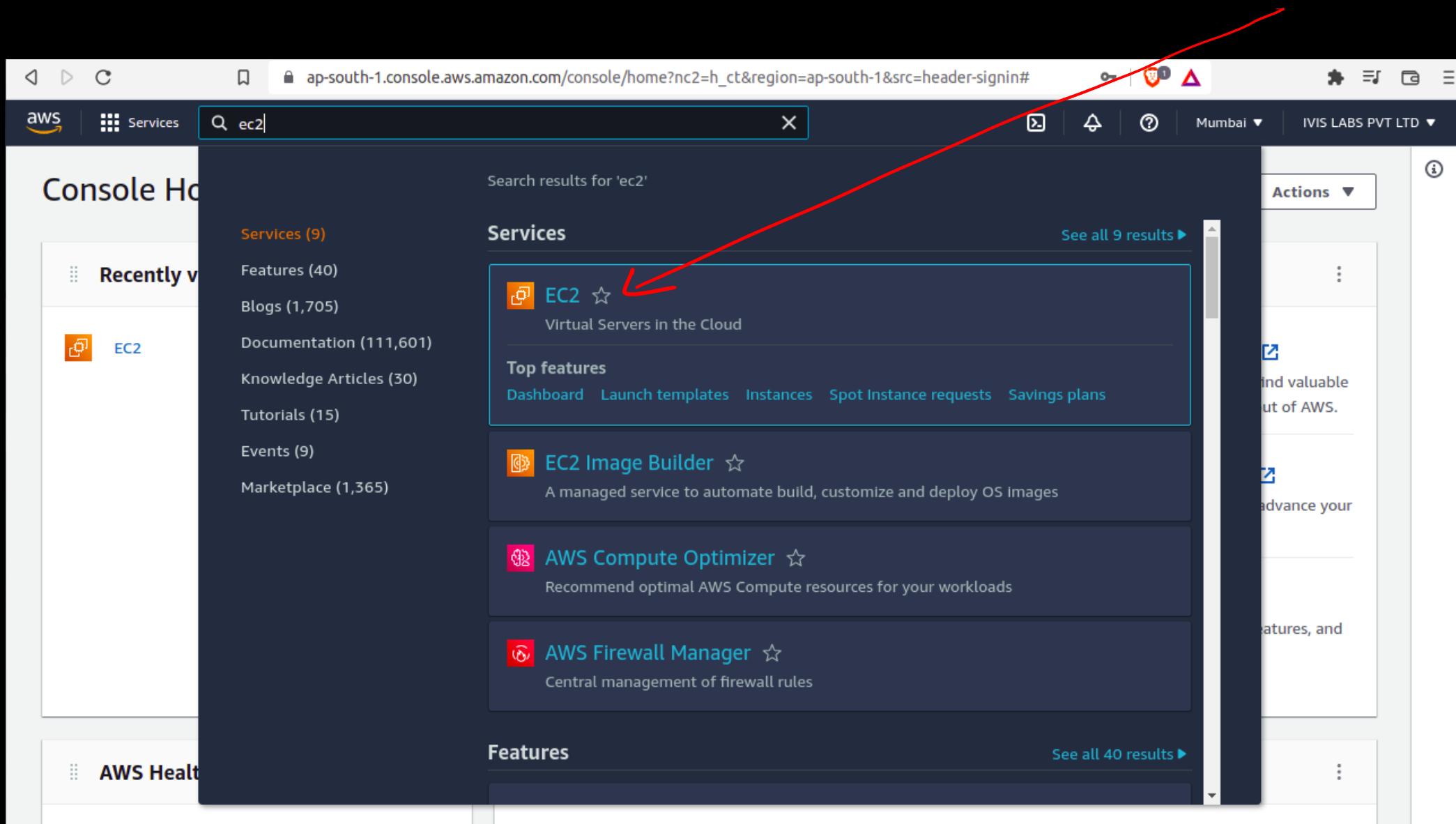
Web Server

NGINX DEMO



Screen Shots

Login → AWS console → Search → EC2 → Click



Launch EC2 Instance & Search Ubuntu

You've been invited to try an early, beta iteration of the new launch instance wizard. We will continue to improve the experience over the next few months. We're asking customers for their feedback on this early release. To exit the new launch instance wizard at any time, choose the **Cancel** button.

Try it now!

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search for services, features, blogs, docs, and more [Alt+S]

Cancel and Exit

Quick Start (7)

My AMIs (0)

AWS Marketplace (1117)

Community AMIs (16188)

Free tier only (i)

Search by Systems Manager parameter

K < 1 to 7 of 7 AMIs > |

64-bit (x86)
 64-bit (Arm)

Select

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-0851b76e8b1bce90b (64-bit x86) / ami-0491e5015eb6e7a9b (64-bit Arm)

Free tier eligible

Ubuntu Server 20.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Select

Feedback English (US) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Select

choose t2.micro [free tier] & Review & Launch

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance families ▾ Current generation ▾ Show/Hide Columns

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

Feedback English (US) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Launch

The screenshot shows the Step 7: Review Instance Launch page of the AWS Instance Launch Wizard. The page title is "Step 7: Review Instance Launch". A navigation bar at the top includes links for "1. Choose AMI", "2. Choose Instance Type", "3. Configure Instance", "4. Add Storage", "5. Add Tags", "6. Configure Security Group", and "7. Review". The "7. Review" link is highlighted with an orange underline.

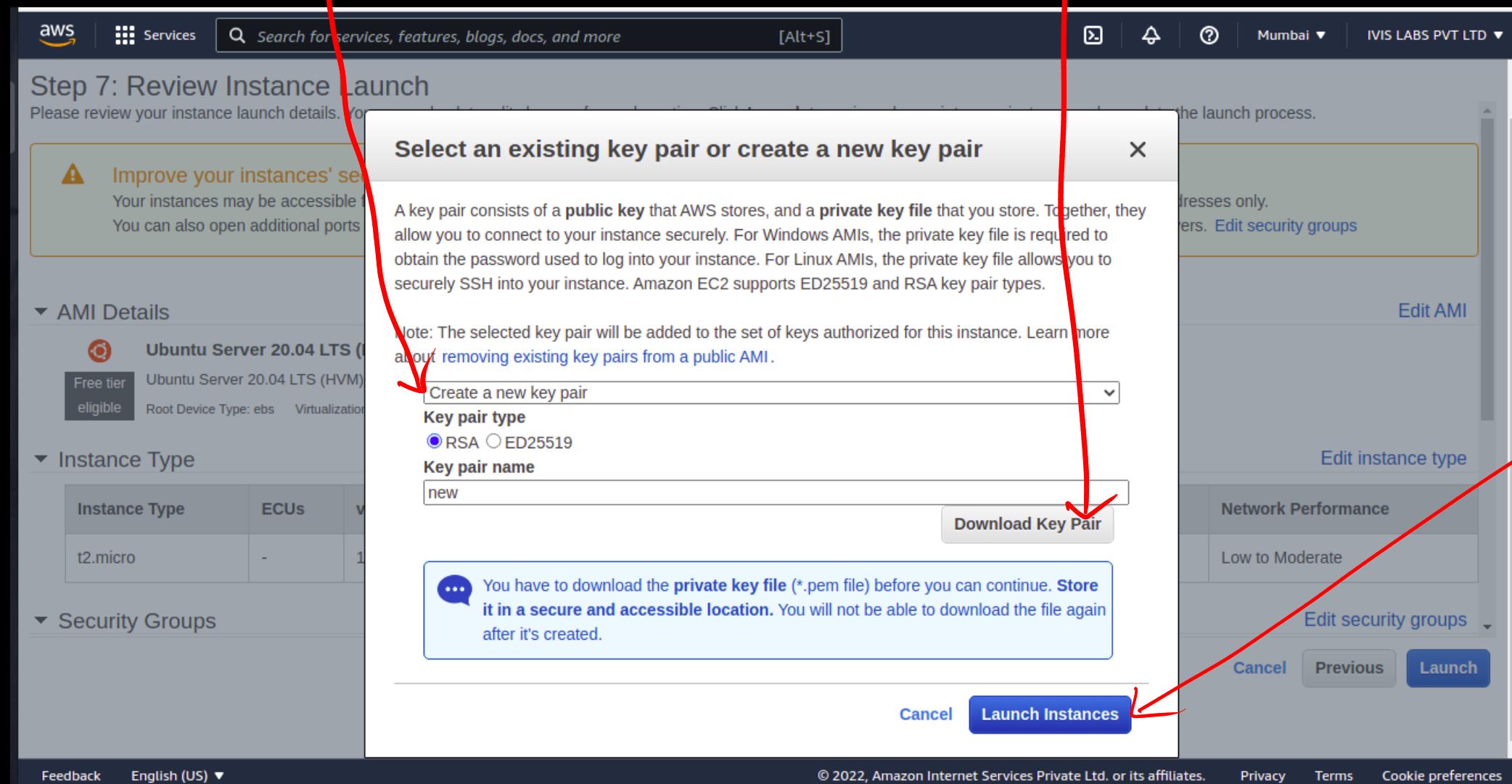
The main content area displays the following sections:

- AMI Details:** Shows the selected AMI as "Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-0851b76e8b1bce90b". It indicates "Free tier eligible".
- Instance Type:** Shows the selected instance type as "t2.micro" with details: ECUs 1, vCPUs 1, Memory 1 GiB, Instance Storage EBS only, EBS-Optimized Available -, and Network Performance Low to Moderate.
- Security Groups:** Shows the selected security group as "Launch-wizard-4".

At the bottom right of the page are three buttons: "Cancel", "Previous", and a large blue "Launch" button. The "Launch" button is highlighted with a red arrow pointing to it from the handwritten "Launch" at the top left.

Page footer includes links for "Feedback", "English (US)", "© 2022, Amazon Internet Services Private Ltd. or its affiliates.", "Privacy", "Terms", and "Cookie preferences".

Create a new key pair & Download [Keep it safe]



Lanch
Instance

Instance started : You will get a public IP
New instance id

The screenshot shows the AWS EC2 Instances page. At the top, there's a search bar and a breadcrumb navigation path: EC2 > Instances > i-021f80aa8ea530cde. A red arrow points from the handwritten note "New instance id" to the instance ID in the breadcrumb. Another red arrow points from the handwritten note "Public IP" to the Public IPv4 address field, which contains "13.234.66.35".

Instance summary for i-021f80aa8ea530cde

Updated less than a minute ago

Instance ID	Public IPv4 address	Private IPv4 addresses
i-021f80aa8ea530cde	13.234.66.35 open address	172.31.2.66
IPv6 address	Instance state	Public IPv4 DNS
-	Running	ec2-13-234-66-35.ap-south-1.compute.amazonaws.com open address
Hostname type	Private IP DNS name (IPv4 only)	Answer private resource DNS name
IP name: ip-172-31-2-66.ap-south-1.compute.internal	ip-172-31-2-66.ap-south-1.compute.internal	-
Instance type	Elastic IP addresses	VPC ID
t2.micro	-	vpc-deb547b5
AWS Compute Optimizer finding	IAM Role	Subnet ID
Opt-in to AWS Compute Optimizer for recommendations. Learn more	-	subnet-ea8ac3a6

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Feedback English (US) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Select Security tab & click the Security group link

The screenshot shows the AWS EC2 instance details page. A red arrow points from the handwritten instruction at the top left down to the 'Security' tab in the navigation bar. Another red arrow points from the 'Security' tab to the 'Security groups' section, where a blue link labeled 'sg-0c80c0a2c6618ffc6 (launch-wizard-4)' is highlighted.

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

Details **Security** Networking Storage Status checks Monitoring Tags

Security details

IAM Role	Owner ID	Launch time
-	907359805180	Sun Mar 06 2022 08:56:34 GMT+0530 (India Standard Time)

Security groups

sg-0c80c0a2c6618ffc6 (launch-wizard-4)

Inbound rules

Filter rules	<	1	>	
Security group rule ID	Port range	Protocol	Source	Security groups
sgr-04809fc8c33eca006	22	TCP	0.0.0.0/0	launch-wizard-4

Outbound rules

https://ap-south-1.console.aws.amazon.com/ec2/v2/home... © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Edit Inbound Rules

The screenshot shows the AWS EC2 Security Groups interface. At the top, there's a summary card for a security group named "launch-wizard-4". Below it, the "Inbound rules" tab is selected. A prominent red arrow points from the handwritten title "Edit Inbound Rules" at the top left towards the "Edit inbound rules" button in the top right of the main content area. The main content area displays the "Inbound rules (1/1)" table, which contains one rule: "sgr-04809fc8c33eca006" (Name), IPv4 (IP version), SSH (Type), and TCP (Protocol). A modal message box is visible, stating "You can now check network connectivity with Reachability Analyzer" with a "Run Reachability Analyzer" button.

Name	Security group rule ID	IP version	Type	Protocol
-	sgr-04809fc8c33eca006	IPv4	SSH	TCP

Feedback English (US) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Add Rules

[Note: adding 8000 & 9000]

The screenshot shows the AWS CloudFront Rules configuration page. There are eight rules listed, all targeting port 8000. The first seven rules use standard protocols (SSH, HTTP, TCP, HTTPS), while the eighth rule is a custom TCP rule. A red arrow points from the handwritten note to the eighth rule, indicating it is the one being added.

Protocol	Port	Source IP Range	Action
SSH	22	Custom: 0.0.0.0/0	Allow
HTTP	80	Anywhere: 0.0.0.0/0	Allow
HTTP	80	Anywhere: ::/0	Allow
HTTPS	443	Anywhere: 0.0.0.0/0	Allow
HTTPS	443	Anywhere: ::/0	Allow
Custom TCP	8000	Anywhere: 0.0.0.0/0	Allow
Custom TCP	8000	Anywhere: ::/0	Allow

Feedback English (US) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Adding 9000 port

The screenshot shows the AWS CloudFront Rules configuration page. At the top, there is a search bar and a navigation bar with 'Mumbai' and 'IVIS LABS PVT LTD'. Below the search bar, there is a table listing seven existing rules. A red arrow points from the handwritten note 'Adding 9000 port' down to the 'Custom TCP' rule with port 9000, which has a red box around its port number. Another red arrow points from the same handwritten note to the 'Add rule' button at the bottom left.

	Protocol	Port	Target	Action
-	HTTPS	443	Anywhere	::/0
-	HTTPS	443	Anywhere	0.0.0.0/0
-	Custom TCP	8000	Anywhere	::/0
-	Custom TCP	8000	Anywhere	0.0.0.0/0
-	Custom TCP	8000	Anywhere	::/0
-	Custom TCP	9000	Anywhere	0.0.0.0/0
-	Custom TCP	9000	Anywhere	::/0

Add rule

Feedback English (US) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Save rules

The screenshot shows the AWS CloudFront Rules configuration interface. At the top, there's a search bar and navigation links for Mumbai and IVIS LABS PVT LTD. Below the search bar, there's a list of five rules:

- Rule 1: HTTPS, TCP, port 443, Anywhere, 0.0.0.0/0
- Rule 2: Custom TCP, TCP, port 8000, Anywhere, 0.0.0.0/0
- Rule 3: Custom TCP, TCP, port 8000, Anywhere, ::/0
- Rule 4: Custom TCP, TCP, port 9000, Anywhere, 0.0.0.0/0
- Rule 5: Custom TCP, TCP, port 9000, Anywhere, ::/0

Each rule row includes a delete button on the far right. At the bottom left is an "Add rule" button. At the bottom right are three buttons: "Cancel", "Preview changes", and "Save rules". The "Save rules" button is highlighted with a red border.

Come back to Instance & You can connect using SSH PEM key [which you have downloaded] OR Connect

The screenshot shows the AWS EC2 Instances page for an instance with ID i-021f80aa8ea530cde. The 'Connect' button in the top navigation bar is highlighted with a red arrow. The page displays various details about the instance, including its ID, public and private IP addresses, state, and VPC information.

Instance ID	Public IPv4 address	Private IPv4 addresses
i-021f80aa8ea530cde	13.234.66.35 open address	172.31.2.66
IPv6 address	Instance state	Public IPv4 DNS
-	Running	ec2-13-234-66-35.ap-south-1.compute.amazonaws.com open address
Hostname type	Private IP DNS name (IPv4 only)	Answer private resource DNS name
IP name: ip-172-31-2-66.ap-south-1.compute.internal	ip-172-31-2-66.ap-south-1.compute.internal	-
Instance type	Elastic IP addresses	VPC ID
t2.micro	-	vpc-deb547b5
AWS Compute Optimizer finding	IAM Role	Subnet ID
Opt-in to AWS Compute Optimizer for recommendations. Learn more	-	subnet-ea8ac3a6

Via
browser

Connect

The screenshot shows the 'Connect to instance' page in the AWS Management Console. At the top, there's a navigation bar with the AWS logo, a 'Services' dropdown, a search bar containing 'Search for services, features, blogs, docs, and more', and a keyboard shortcut '[Alt+S]'. To the right are icons for notifications, help, location ('Mumbai'), and account ('IVIS LABS PVT LTD').

The main content area has a title 'Connect to instance' with an 'Info' link. Below it, a sub-header says 'Connect to your instance i-021f80aa8ea530cde using any of these options'. There are four tabs: 'EC2 Instance Connect' (which is selected and highlighted in orange), 'Session Manager', 'SSH client', and 'EC2 Serial Console'.
Under the tabs, there are fields for 'Instance ID' (with a value 'i-021f80aa8ea530cde') and 'Public IP address' (with a value '13.234.66.35'). A 'User name' field contains the value 'ubuntu'. A note below the user name field says: 'Connect using a custom user name, or use the default user name ubuntu for the AMI used to launch the instance.'
A callout box contains a note: 'Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.'
At the bottom right of the main content area are two buttons: 'Cancel' and a large orange 'Connect' button. A red arrow points from the handwritten 'Connect' at the top left of the image to the orange 'Connect' button at the bottom right.

https://ap-south-1.console.aws.amazon.com/ec2/v2/connect/ubuntu/i-021f80aa8ea530cde © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

You have entered into the console
↳ Update the Ubuntu OS

System information as of Sun Mar 6 03:31:19 UTC 2022

System load: 0.01	Processes: 102
Usage of /: 18.2% of 7.69GB	Users logged in: 0
Memory usage: 20%	IPv4 address for eth0: 172.31.2.66
Swap usage: 0%	

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

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-172-31-2-66:~$ sudo apt update
```

i-021f80aa8ea530cde

Public IPs: 13.234.66.35 Private IPs: 172.31.2.66

① Check nginx installed?

```
ubuntu@ip-172-31-2-66:~$ sudo systemctl status nginx  
Unit nginx.service could not be found.  
ubuntu@ip-172-31-2-66:~$ sudo systemctl status apache2  
Unit apache2.service could not be found.  
ubuntu@ip-172-31-2-66:~$ sudo systemctl status nginx  
Unit nginx.service could not be found.  
ubuntu@ip-172-31-2-66:~$
```

①

②

check apache2
installed?

You will get
not found,
msg

i-021f80aa8ea530cde

Public IPs: 13.234.66.35 Private IPs: 172.31.2.66

① Install nginx

```
ubuntu@ip-172-31-2-66:~$ sudo apt 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 libjbig0 libjpeg-turbo8 libjpeg8
  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 libjbig0 libjpeg-turbo8 libjpeg8
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream libtiff5
  libwebp6 libxpm4 nginx nginx-common nginx-core
0 upgraded, 17 newly installed, 0 to remove and 78 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
```

② Continue type 'y'

i-021f80aa8ea530cde

Public IPs: 13.234.66.35 Private IPs: 172.31.2.66

① Check nginx installed?

```
Setting up fontconfig-config (2.13.1-2ubuntu3) ...
Setting up libnginx-mod-stream (1.18.0-0ubuntu1.2) ...
Setting up libtiff5:amd64 (4.1.0+git191117-2ubuntu0.20.04.2) ...
Setting up libfontconfig1:amd64 (2.13.1-2ubuntu3) ...
Setting up libgd3:amd64 (2.2.5-5.2ubuntu2.1) ...
Setting up libnginx-mod-http-image-filter (1.18.0-0ubuntu1.2) ...
Setting up nginx-core (1.18.0-0ubuntu1.2) ...
Setting up nginx (1.18.0-0ubuntu1.2) ...
Processing triggers for ufw (0.36-6ubuntu1) ...
Processing triggers for systemd (245.4-4ubuntu3.13) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
ubuntu@ip-172-31-2-66:~$ sudo 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 Sun 2022-03-06 03:33:42 UTC; 10s ago
    Docs: man:nginx(8)
 Main PID: 2373 (nginx)
    Tasks: 2 (limit: 1147)
   Memory: 4.9M
      CGrou: /system.slice/nginx.service
           └─2373 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
              ├─2374 nginx: worker process
Mar 06 03:33:42 ip-172-31-2-66 systemd[1]: Starting A high performance web server and a reverse proxy server...
Mar 06 03:33:42 ip-172-31-2-66 systemd[1]: Started A high performance web server and a reverse proxy server.
ubuntu@ip-172-31-2-66:~$ █
```

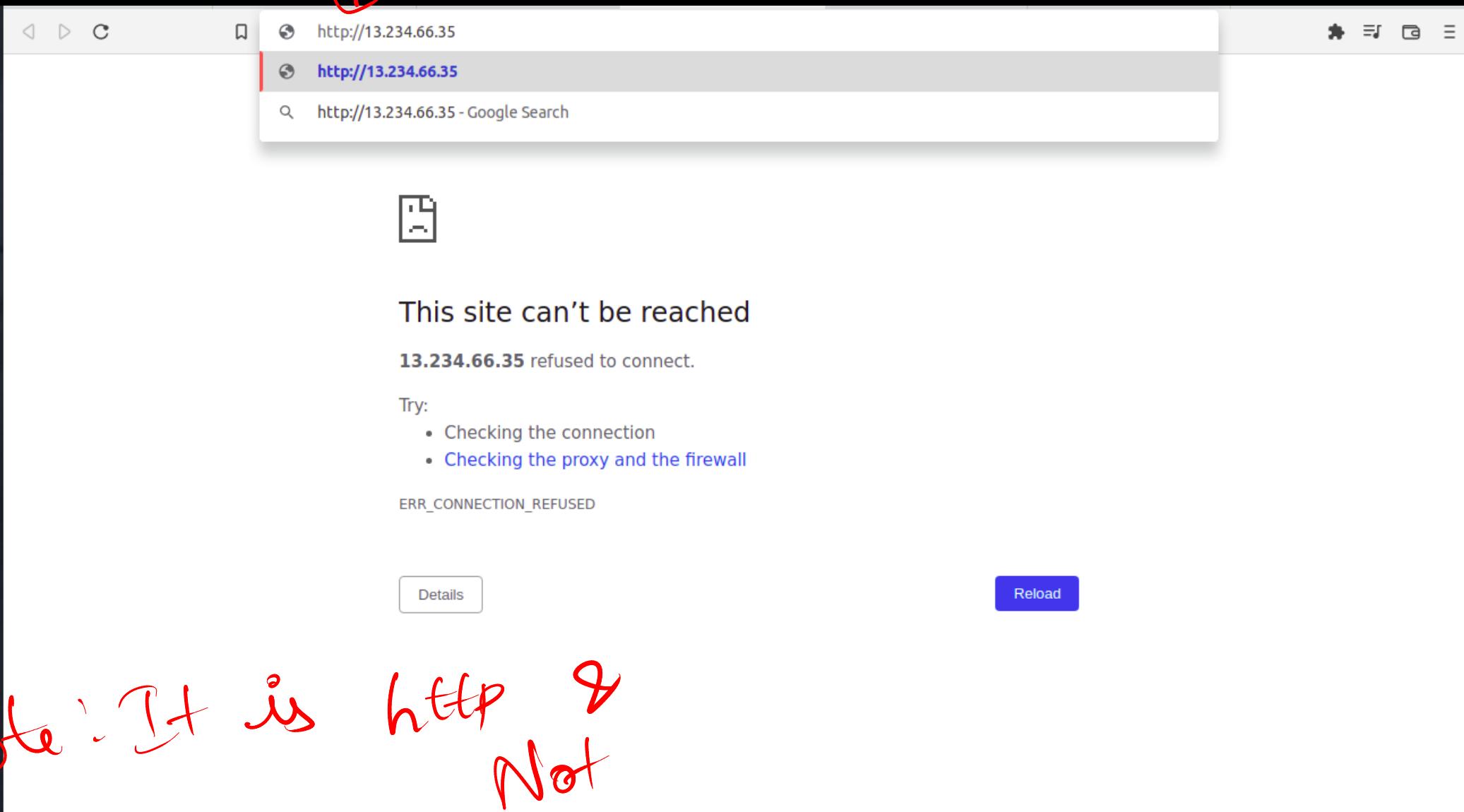
Messag

Copy Public ip from AWS instance

A screenshot of the AWS EC2 Instances Details page for an instance with ID i-021f80aa8ea530cde. A red arrow points to the Public IPv4 address field, which contains the value 13.234.66.35. The page shows various details about the instance, including its state (Running), type (t2.micro), and network configurations.

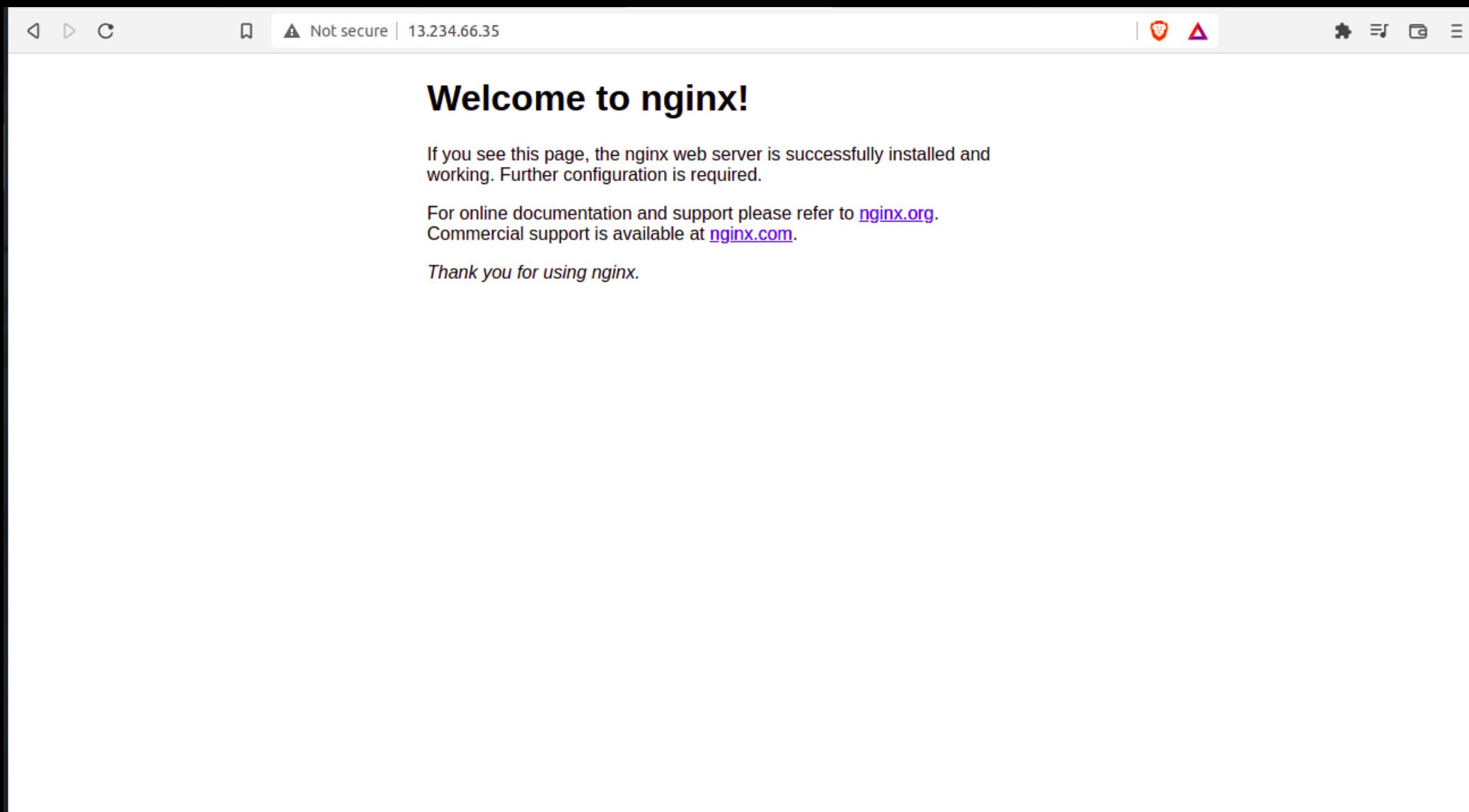
Attribute	Value
Instance ID	i-021f80aa8ea530cde
PublicIPv4 address	13.234.66.35
Private IPv4 addresses	172.31.2.66
IPv6 address	-
Instance state	Running
Private IP DNS name (IPv4 only)	ip-172-31-2-66.ap-south-1.compute.internal
Elastic IP addresses	-
VPC ID	vpc-deb547b5
Hostname type	IP name: ip-172-31-2-66.ap-south-1.compute.internal
Subnet ID	subnet-ea8ac3a6
Instance type	t2.micro
IAM Role	-
AWS Compute Optimizer finding	Opt-in to AWS Compute Optimizer for recommendations. Learn more

New Tab: http://public_ip & Enter



Note: It is http &
Not https

You will get default nginx page



- × Host a static website & listen at 8000 port
 - ↳ Create a configuration file [amar.local.conf]

The image shows a terminal window with a black background and white text. Three red arrows point from the numbers 1, 2, and 3 at the top to specific lines of code in the terminal window.

```
ubuntu@ip-172-31-2-66:~$ cd /etc/nginx/conf.d/  
ubuntu@ip-172-31-2-66:/etc/nginx/conf.d$ ls  
ubuntu@ip-172-31-2-66:/etc/nginx/conf.d$ sudo nano amar.local.conf
```

- ① Points to the command `cd /etc/nginx/conf.d/`.
- ② Points to the command `ls`.
- ③ Points to the command `sudo nano amar.local.conf`.

Configuration Content [Note: check syntax & spaces]

① Listen to 8000 port

② Search for index.html page

③ Server name

④ my website root file directory

⑤ do `:wq` & type 'Y' for saving

```
GNU nano 4.8                               amar.local.conf                                Modified
server{
    listen 8000 default_server;
    index index.html;
    server_name amar.local;
    root /var/www/amar.local;
}
Save modified buffer?  Y Yes
N No      ^C Cancel
i-021f80aa8ea530cde
Public IPs: 13.234.66.35  Private IPs: 172.31.2.66
```

Check configuration file status

```
ubuntu@ip-172-31-2-66:/etc/nginx/conf.d$ sudo nginx -t  
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok  
nginx: configuration file /etc/nginx/nginx.conf test is successful  
ubuntu@ip-172-31-2-66:/etc/nginx/conf.d$ █
```

success message

i-021f80aa8ea530cde

* Create a directory & create a html file

```
ubuntu@ip-172-31-2-66:/etc/nginx/conf.d$ cd /var/www/  
ubuntu@ip-172-31-2-66:/var/www$ ls  
html  
ubuntu@ip-172-31-2-66:/var/www$ sudo mkdir amar.local  
ubuntu@ip-172-31-2-66:/var/www$ cd amar.local/  
ubuntu@ip-172-31-2-66:/var/www/amar.local$ sudo nano index.html
```

① Navigate

② Listing

③ making a new
directory

④ Navigate

⑤ Create a html file
or

You can copy from local
to server

[which is
mentioned
in configu
file]

* Write html content & Save

```
GNU nano 4.8                               index.html                                Modified
<html>
  <body>
    Welcome to the beautiful world!
    <br/>
    I am listening to 8000 port
    <br/>

    Copyright 2022, aicopia.com
  </body>
</html>
```

① content

② save (ctrl+x & type 'y')

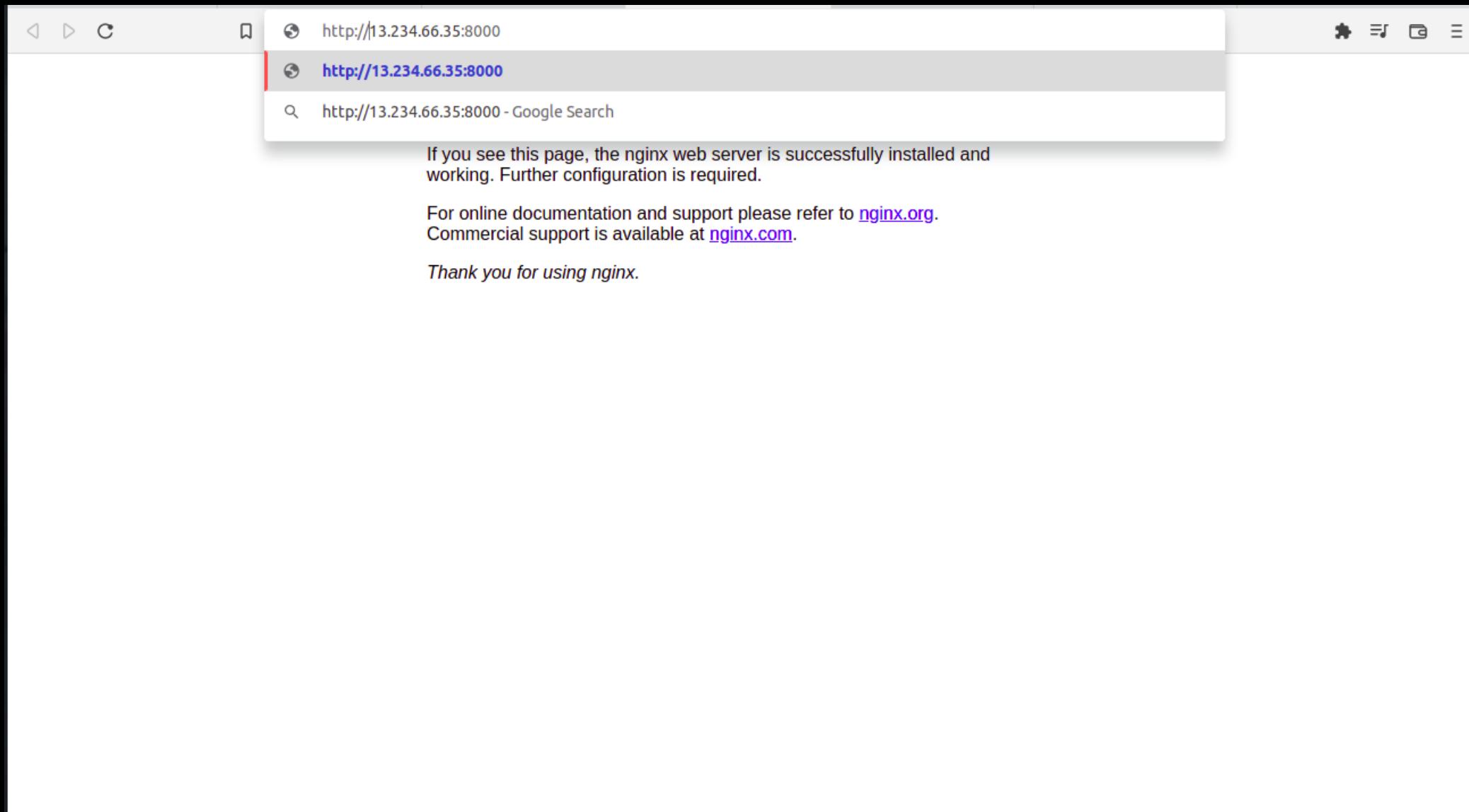
Save modified buffer?

Restart Nginx server

① Restart

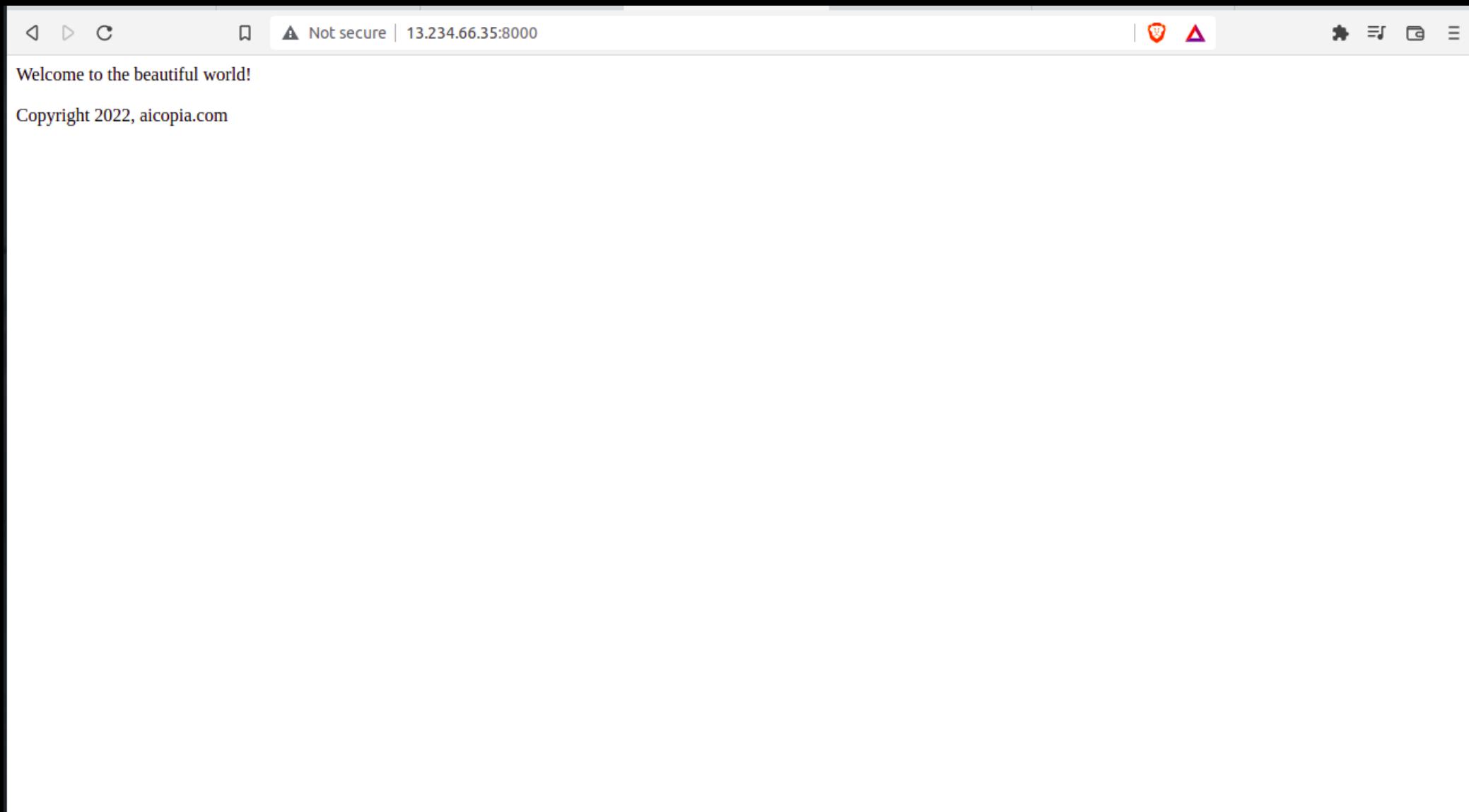
```
ubuntu@ip-172-31-2-66:/etc/nginx/conf.d$ sudo systemctl restart nginx
```

Open new tab & type `http://public_ip:8000`



Bingo! You should able to see your html page

Note: It is http , but not https



* Let us repeat the same process listening at 9000 port

① Navigate

② List

③ Create a new conf file

```
ubuntu@ip-172-31-2-66:/var/www/amar.local$ cd /etc/nginx/conf.d/  
ubuntu@ip-172-31-2-66:/etc/nginx/conf.d$ ls  
amar.local.conf  
ubuntu@ip-172-31-2-66:/etc/nginx/conf.d$ sudo nano amar1.local.conf
```

i-021f80aa8ea530cde

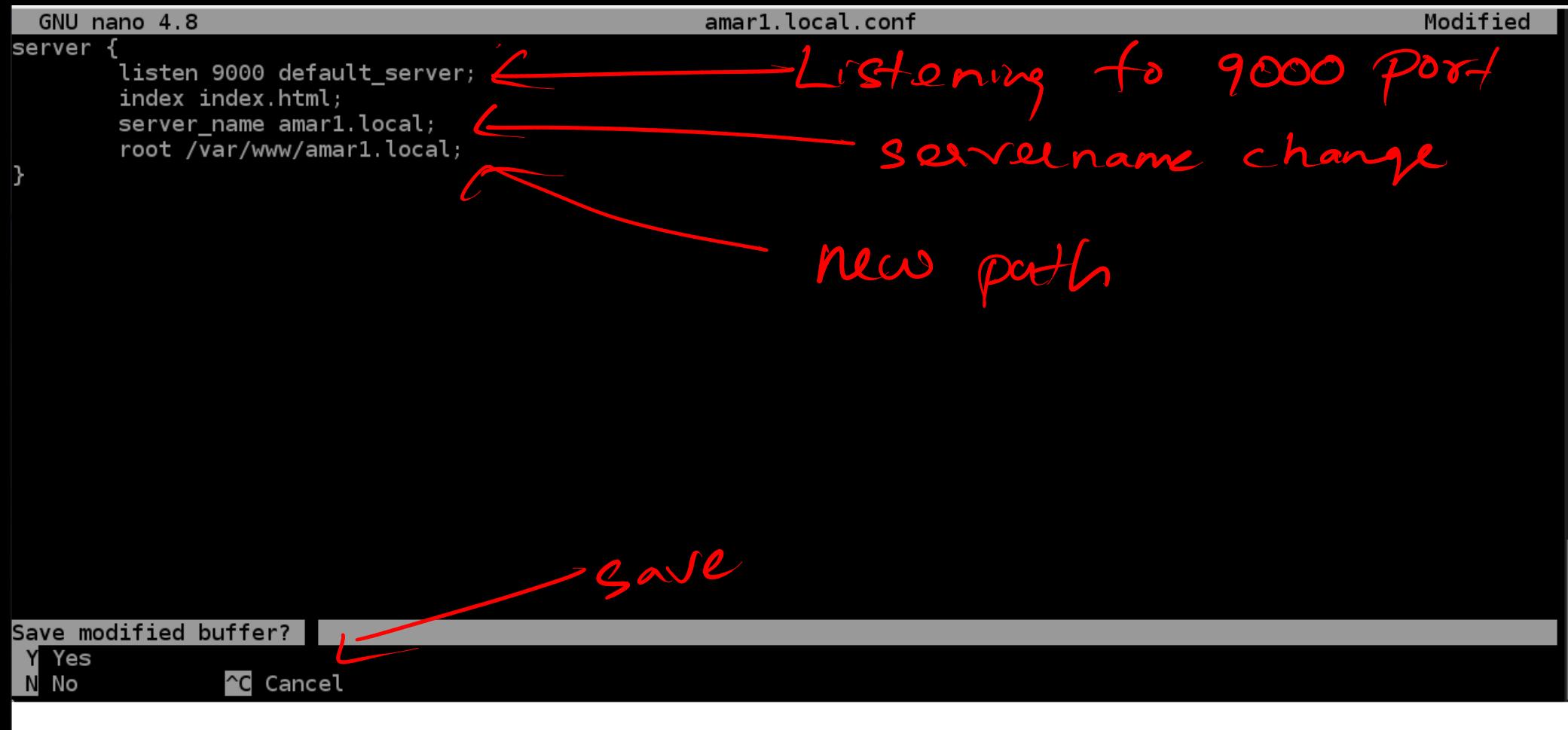
Create the configuration setup

```
GNU nano 4.8                               amar1.local.conf                         Modified
server {
    listen 9000 default_server;
    index index.html;
    server_name amar1.local;
    root /var/www/amari1.local;
}
```

Listening to 9000 Port
servername change
new path

Save

Save modified buffer? [Y/N] ^C Cancel



Check configuration & create a new page in new directory

① Check conf. file

```
ubuntu@ip-172-31-2-66:/etc/nginx/conf.d$ sudo nginx -t  
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok  
nginx: configuration file /etc/nginx/nginx.conf test is successful  
ubuntu@ip-172-31-2-66:/etc/nginx/conf.d$ cd /var/www/  
ubuntu@ip-172-31-2-66:/var/www$ ls  
amar.local html  
ubuntu@ip-172-31-2-66:/var/www$ sudo mkdir amar1.local  
ubuntu@ip-172-31-2-66:/var/www$ ls  
amar.local amar1.local html  
ubuntu@ip-172-31-2-66:/var/www$ cd amar1.local/  
ubuntu@ip-172-31-2-66:/var/www/amar1.local$ ls  
ubuntu@ip-172-31-2-66:/var/www/amar1.local$ sudo nano index.html
```

② Success message

③ Navigate

④ List

⑤ Create a new folder

⑥ Listing

⑦ Navigate

⑧ Create a new file

i-021f80aa8ea530cde

Add the below content

GNU nano 4.8 index.html Modified

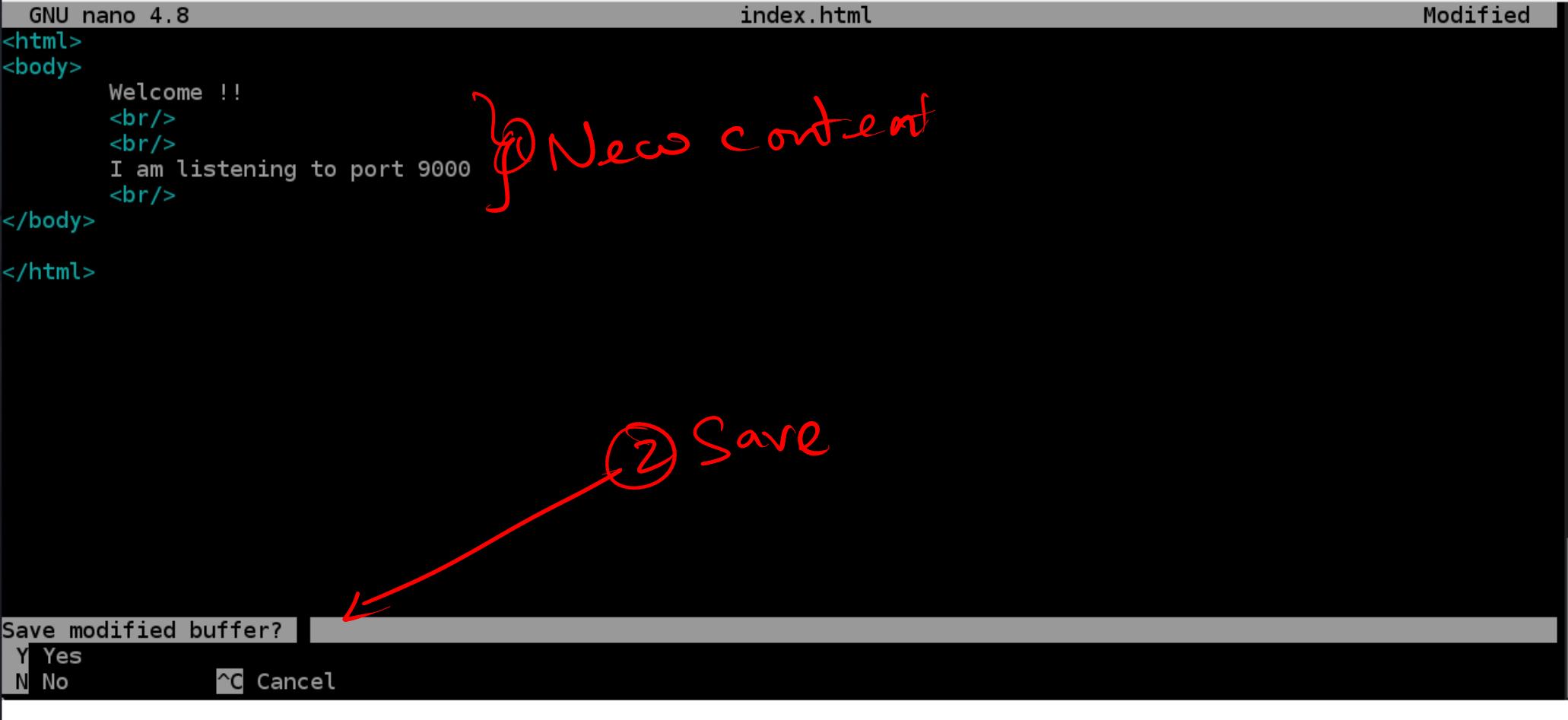
```
<html>
<body>
    Welcome !!
    <br/>
    <br/>
    I am listening to port 9000
    <br/>
</body>
</html>
```

{New content}

② Save

Save modified buffer?

Y Yes
N No ^C Cancel

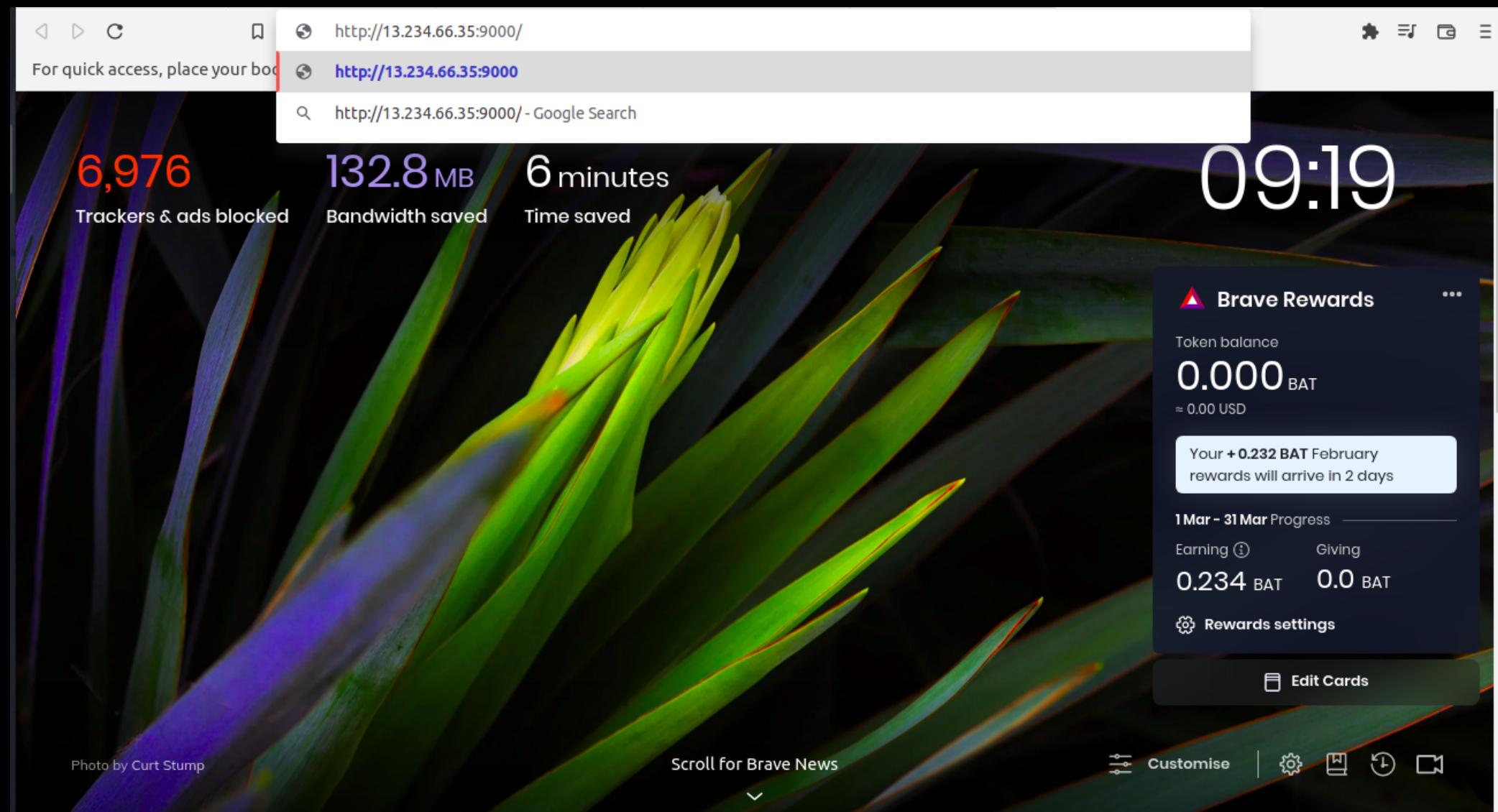


Restart the server

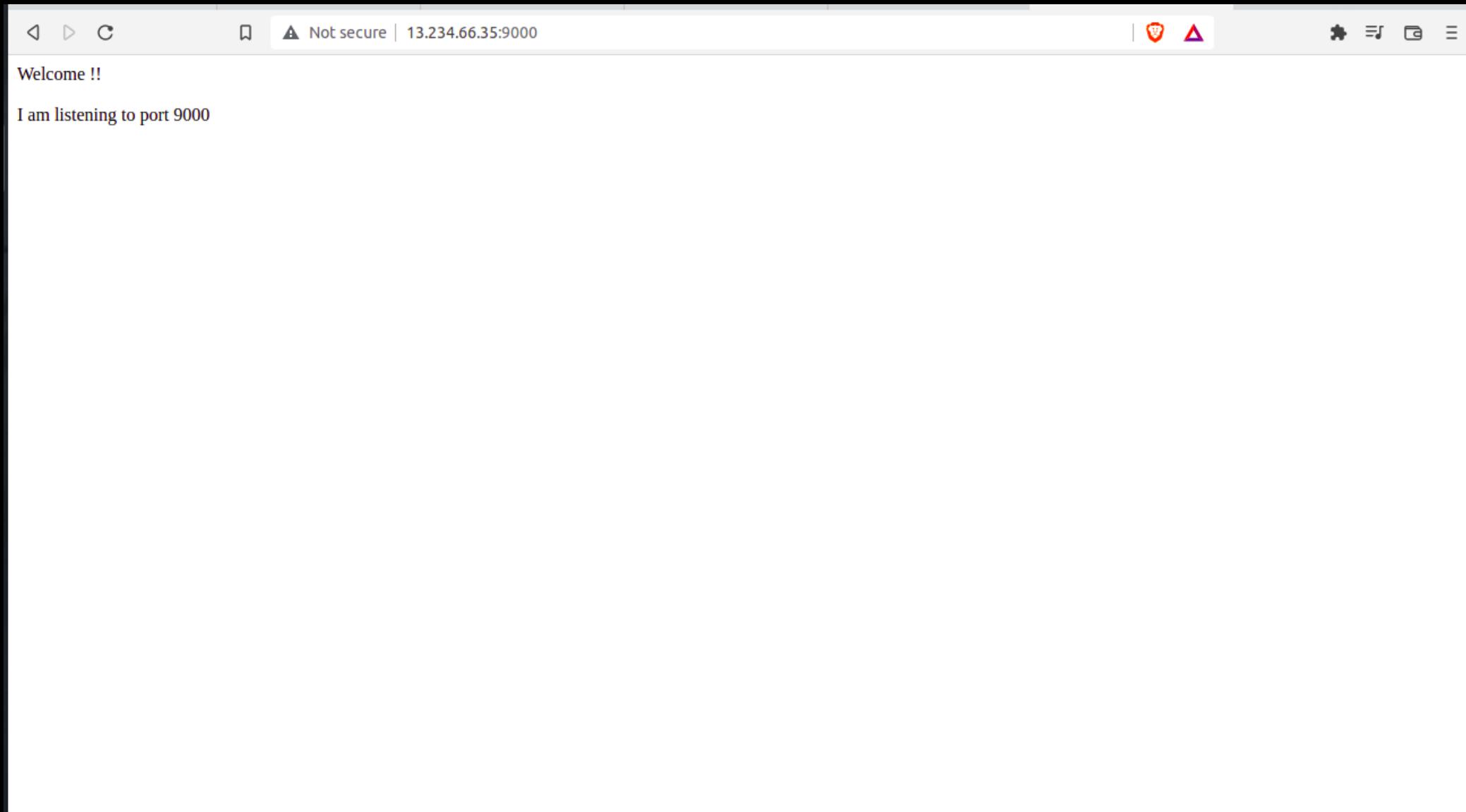
```
ubuntu@ip-172-31-2-66:/etc/nginx/conf.d$ sudo systemctl restart nginx
```



Open New tab: <http://public-ip:9000>



Bingo! You will see your web page



CREATING A PROXY SERVER

① → We have two ports

$\text{http://public-ip:8000}$ [/var/www/amar.local/index.html]

$\text{http://public-ip:9000}$ [/var/www/amar.local/index.html]

② The Default

http://public-ip [Default port is 80]

* Let us make this is a proxy server

Change the default configuration file

```
ubuntu@ip-172-31-2-66:/etc/nginx/conf.d$ cd /etc/nginx/sites-available/  
ubuntu@ip-172-31-2-66:/etc/nginx/sites-available$ ls  
default  
ubuntu@ip-172-31-2-66:/etc/nginx/sites-available$
```

① Navigate

② List

Take a backup of default config because we are making changes

```
ubuntu@ip-172-31-2-66:/etc/nginx/sites-available$ ls  
default  
ubuntu@ip-172-31-2-66:/etc/nginx/sites-available$ sudo cp default default.back  
ubuntu@ip-172-31-2-66:/etc/nginx/sites-available$ ls  
default default.back
```

① Backup

② we have backup

Edit Default configuration file

① Open

```
ubuntu@ip-172-31-2-66:/etc/nginx/conf.d$ sudo systemctl restart nginx
ubuntu@ip-172-31-2-66:/etc/nginx/conf.d$ cd /etc/nginx/sites-available/
ubuntu@ip-172-31-2-66:/etc/nginx/sites-available$ ls
default
ubuntu@ip-172-31-2-66:/etc/nginx/sites-available$ sudo nano default
```



Comment all the lines using '# ' symbol before the line & Add new content

```
vi /etc/nginx/sites-available/default
# server_name example.com;
#
# root /var/www/example.com;
# index index.html;
#
# location / {
#     try_files $uri $uri/ =404;
# }
#
upstream backend{
    server 13.234.66.35:8000;
    server 13.234.66.35:9000;
}
server{
    listen 80;
    location / {
        proxy_pass http://backend;
    }
}

Save modified buffer?
Y Yes
N No      ^C Cancel
```

① Comment

② New content

③ public ip & ports

④ proxy with '80' default port

& Check your configuration file & Restart the nginx server

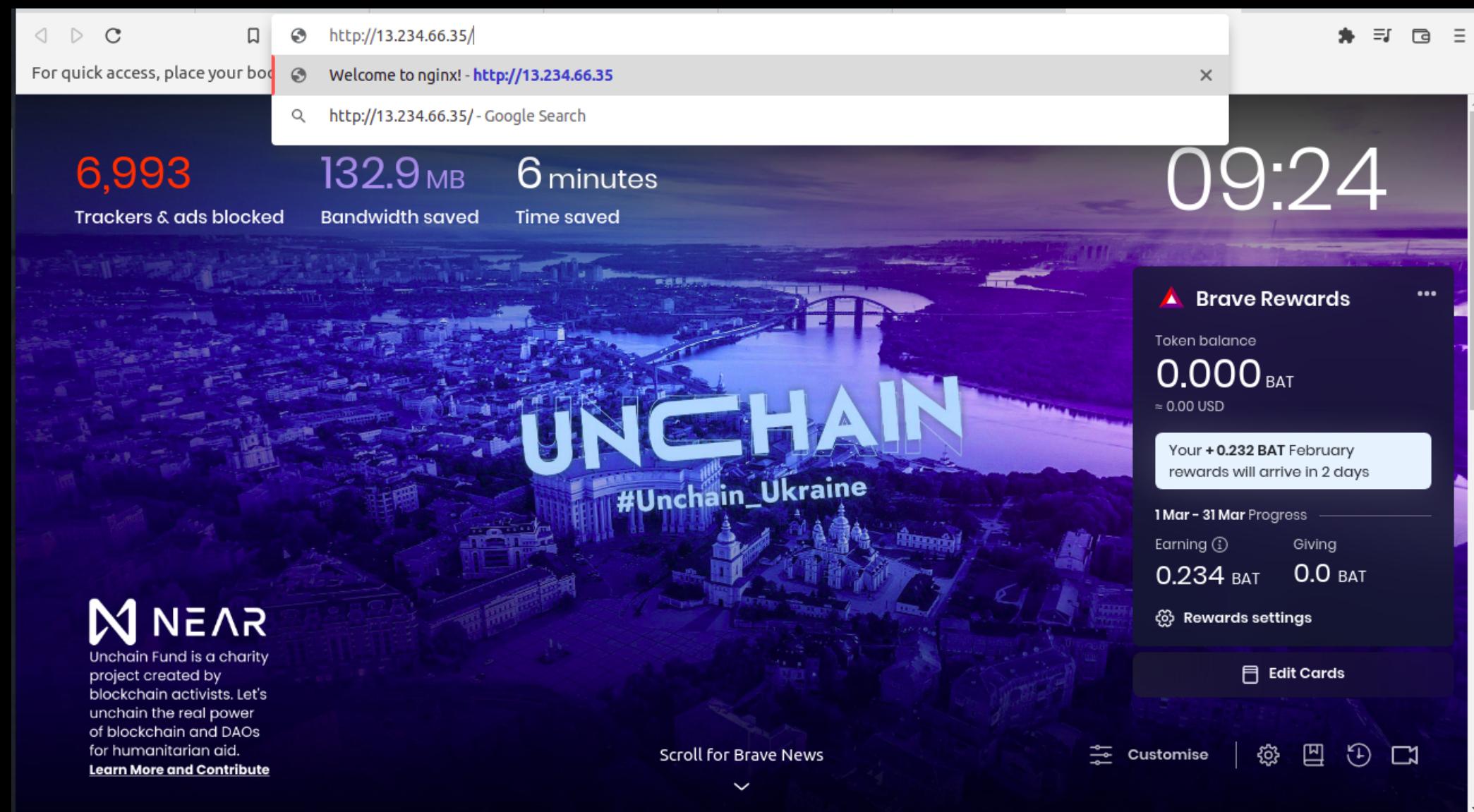
```
ubuntu@ip-172-31-2-66:/etc/nginx/sites-available$ sudo nginx -t  
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok  
nginx: configuration file /etc/nginx/nginx.conf test is successful  
ubuntu@ip-172-31-2-66:/etc/nginx/sites-available$ sudo systemctl restart nginx
```

① Check nginx

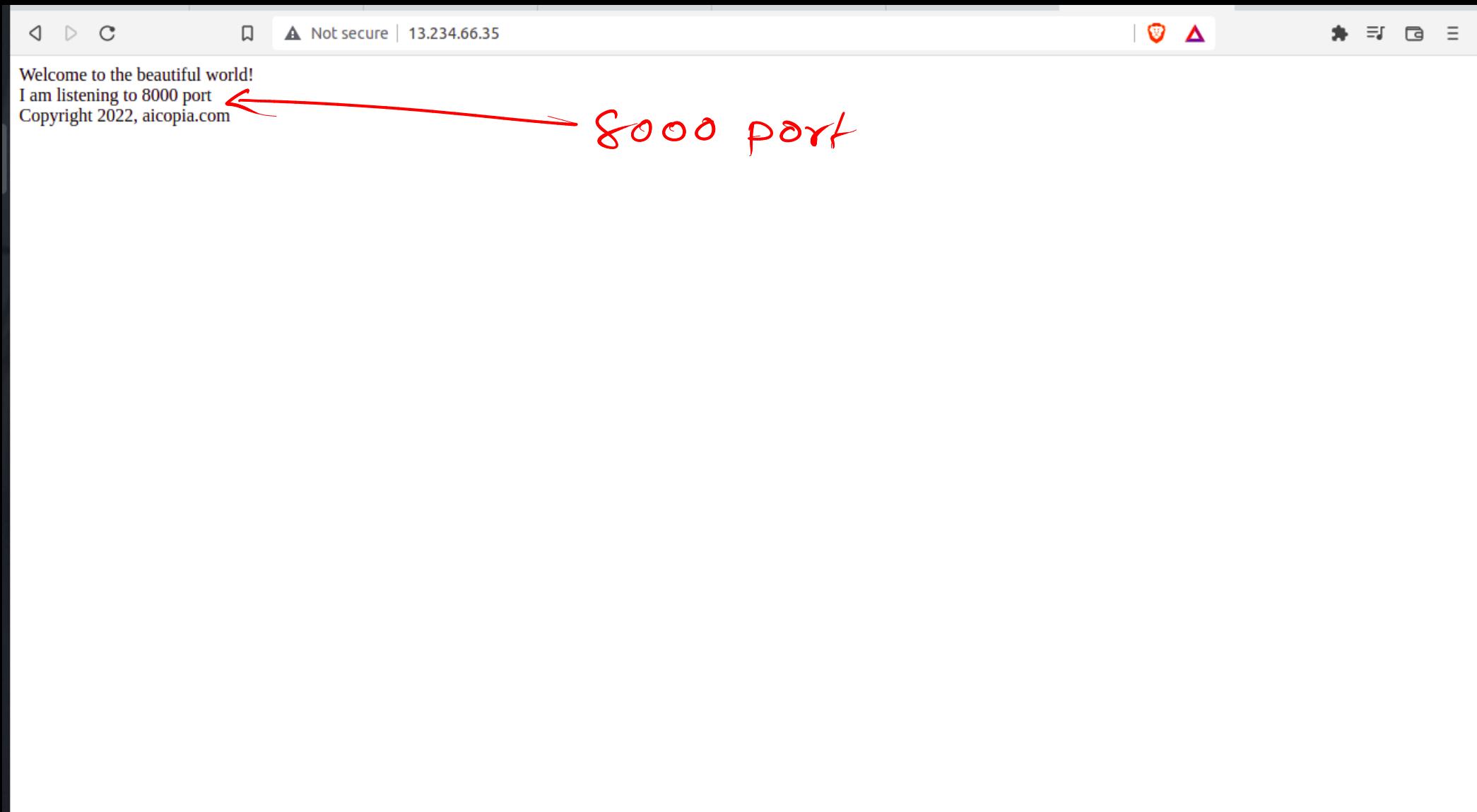
② Message

③ Restart

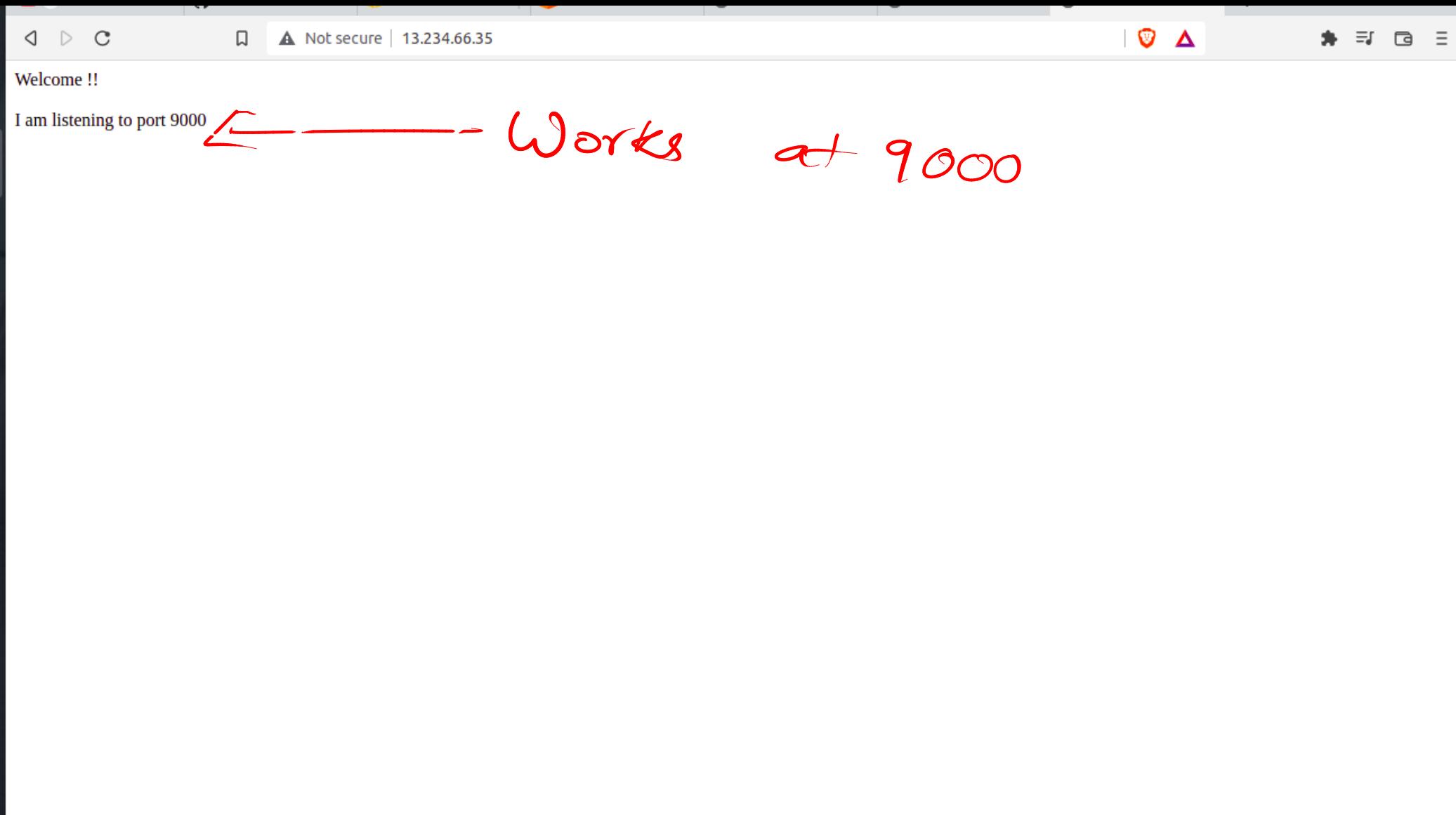
* Open a new tab : <http://public-ip>



Bingo! Proxy works & Refresh it changes the Content



Proxy [Refresh the page



We have experimented

① Proxy

② 8000

③ 9000

in the same instance [1 instance]

We can also explore having 3 different instances.

Website : <https://sites.google.com/view/amarnath-r>

Email : amarnathresearch@gmail.com

Mobile : +91 - 9894104930

COPYRIGHT :

<https://www.aicopia.com/>

- * Only for personal use only
- * Copying partially or fully and replication is under the copyright act
- * Distribution by any means is restricted