

# Step 1 - What is AWS



AWS is Amazon's **cloud** service.

It lets you

1. Rent servers
2. Manage domains
3. Upload objects (mp4 files, jpgs, mp3s ...)
4. Autoscale servers
5. Create k8s clusters

...

The offering we will be focussing on today is **Renting servers**

# Step 2 - EC2 servers

VMs on AWS are called **EC2 Servers**

EC2 stands for Elastic compute Version 2.

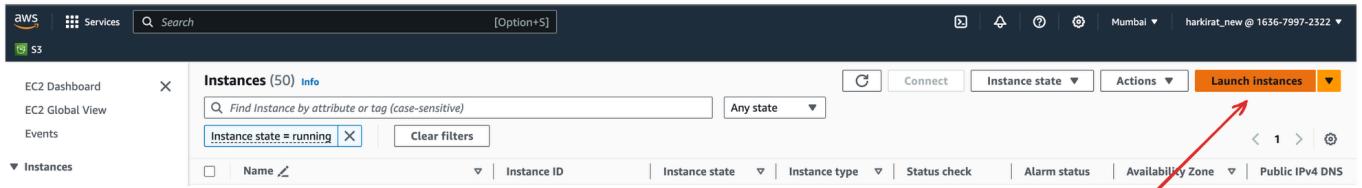
1. **Elastic** - Can increase/decrease the size of the machine
2. **Compute** - It is a machine

You can spin up a new EC2 instance from the aws dashboard

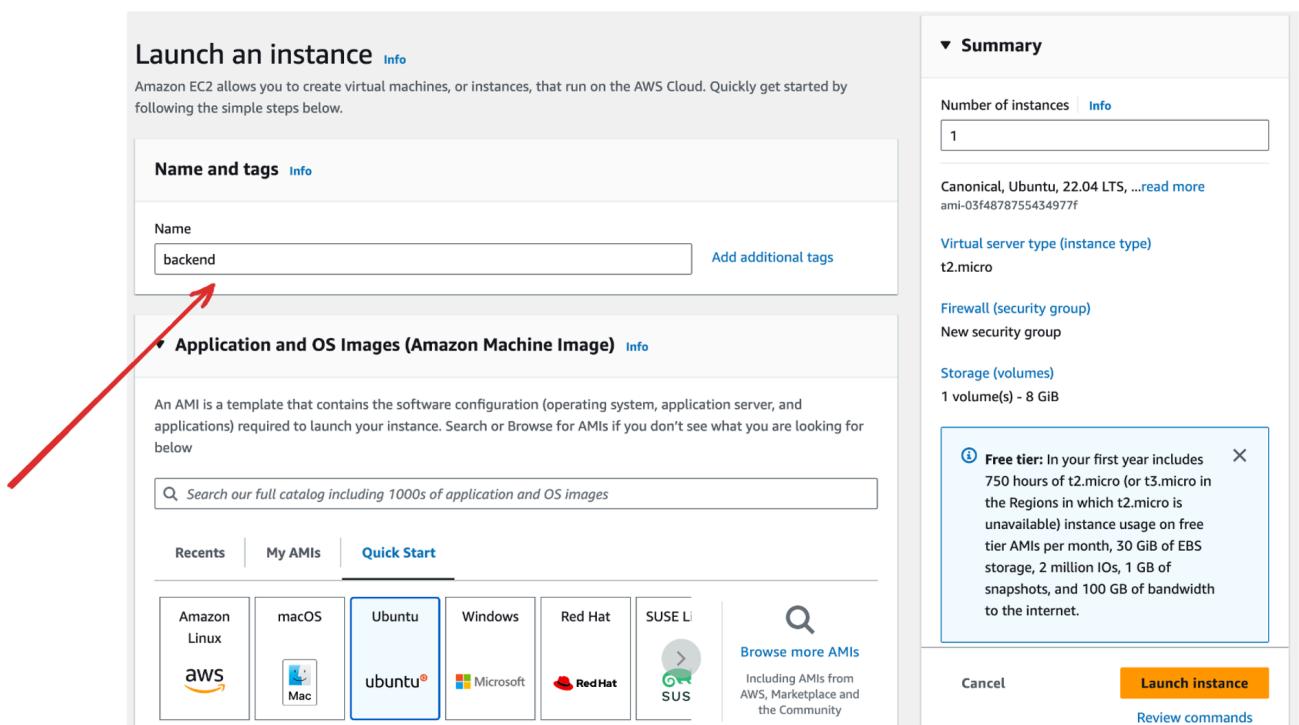
The screenshot shows the AWS Management Console search results for 'ec2'. The search bar at the top contains 'ec2'. On the left, there is a sidebar for 'Amazon S3' with sections for Buckets, Features (57), Resources (New), Documentation (34,444), Knowledge Articles (608), Marketplace (3,317), Blogs (2,123), Events (30), and Tutorials (21). The main area displays search results under 'Services' and 'Features'. Under 'Services', the 'EC2' service is listed with the description 'Virtual Servers in the Cloud'. Under 'Features', the 'EC2 Image Builder', 'Recycle Bin', and 'Amazon Inspector' services are listed. Each service item has a 'See all 13 results' link. On the right side of the search results, there are buttons for 'Edit' and 'Delete'.

# Step 3 - Creating a new EC2 server

## 1. Click on **Launch a new instance**



## 2. Give a name



## 3. Select an OS

## Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

### Name and tags Info

Name  
backend Add additional tags

### Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

Recents | My AMIs | **Quick Start**

Amazon Linux | macOS | **Ubuntu** | Windows | Red Hat | SUSE Linux Enterprise Server

Browse more AMIs Including AMIs from AWS, Marketplace and the Community

### Summary

Number of instances Info

1

Canonical, Ubuntu, 22.04 LTS, ...[read more](#)  
ami-03f4878755434977f

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
New security group

Storage (volumes)  
1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

[Cancel](#)

**Launch instance**

[Review commands](#)

## 4. Select size

Instance type

**t2.micro** Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true  
On-Demand Linux base pricing: 0.0124 USD per Hour  
On-Demand Windows base pricing: 0.017 USD per Hour  
On-Demand RHEL base pricing: 0.0724 USD per Hour  
On-Demand SUSE base pricing: 0.0124 USD per Hour

**t2.nano** Free tier eligible

Family: t2 1 vCPU 0.5 GiB Memory Current generation: true  
On-Demand SUSE base pricing: 0.0062 USD per Hour  
On-Demand Linux base pricing: 0.0062 USD per Hour  
On-Demand Windows base pricing: 0.0085 USD per Hour

**t2.micro** Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true  
On-Demand Linux base pricing: 0.0124 USD per Hour  
On-Demand Windows base pricing: 0.017 USD per Hour  
On-Demand RHEL base pricing: 0.0724 USD per Hour  
On-Demand SUSE base pricing: 0.0124 USD per Hour

**t2.small**

Family: t2 1 vCPU 2 GiB Memory Current generation: true  
On-Demand SUSE base pricing: 0.0548 USD per Hour  
On-Demand Linux base pricing: 0.0248 USD per Hour  
On-Demand RHEL base pricing: 0.0848 USD per Hour  
On-Demand Windows base pricing: 0.034 USD per Hour

**t2.medium**

Family: t2 2 vCPU 4 GiB Memory Current generation: true  
On-Demand Linux base pricing: 0.0496 USD per Hour  
On-Demand Windows base pricing: 0.0676 USD per Hour  
On-Demand RHEL base pricing: 0.1096 USD per Hour  
On-Demand SUSE base pricing: 0.1496 USD per Hour

**t2.large**

Auto-assign public IP Info

Enable

### Summary

Number of instances Info

1

Canonical, Ubuntu, 22.04 LTS, ...[read more](#)  
ami-03f4878755434977f

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
New security group

Storage (volumes)  
1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

[Cancel](#)

**Launch instance**

[Review commands](#)

## 5. Create a new Key pair

## 6. Select Size

**Configure storage** [Info](#)

Advanced

1x 8 GiB gp2 Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

Add new volume

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

Click refresh to view backup information

The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems

**Advanced details** [Info](#)

Canonical, Ubuntu, 22.04 LTS, ...[read more](#)  
ami-03f4878755434977f

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
New security group

Storage (volumes)  
1 volume(s) - 8 GiB

**Free tier:** In your first year includes  
750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel [Launch instance](#) [Review commands](#)

## 7. Allow traffic on http/https

Select [Create new key pair](#)

**Network settings** [Info](#)

Network [Info](#)  
vpc-bdc9c2d5 | VPC-Live

Subnet [Info](#)  
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)  
Enable

Firewall (security groups) [Info](#)  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group [Select existing security group](#)

We'll create a new security group called 'launch-wizard-82' with the following rules:

- Allow SSH traffic from Anywhere  
Helps you connect to your instance
- Allow HTTPS traffic from the internet  
To set up an endpoint, for example when creating a web server
- Allow HTTP traffic from the internet  
To set up an endpoint, for example when creating a web server

**Summary**

Number of instances [Info](#)

1

Canonical, Ubuntu, 22.04 LTS, ...[read more](#)  
ami-03f4878755434977f

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
New security group

Storage (volumes)  
1 volume(s) - 8 GiB

**Free tier:** In your first year includes  
750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

# Step 4 - SSH into server

## 1. Give ssh key permissions

```
chmod 700 kirat-class.pem
```

Copy

## 2. ssh into machine

```
ssh -i kirat-class.pem ubuntu@ec2-65-0-180-32.ap-south-1.compute.amazonaws.com
```

Copy

## 3. Clone repo

```
git clone https://github.com/hkirat/sum-server
```

Copy

 If your aws machine shows you the following error, your aws machine doesn't have access to the internet

Solution - <https://www.tecmint.com/resolve-temporary-failure-in-name-resolution/>

```
[root@ip-172-31-11-253 ~]# ping google.com
ping: google.com: Temporary failure in name resolution
[root@ip-172-31-11-253 ~]#
```

## 4. Install Node.js

 <https://www.digitalocean.com/community/tutorials/how-to-install-node-js-on-ubuntu-20-04>

## 5. Install all dependencies

```
cd sum-server
npm install
```

Copy

## 6. Start backend

```
node index.js
```

Copy

# Step 5 - Install the repo

Clone the repo

<https://github.com/hkirat/sum-sei> Copy

# Step 6 - Try hitting the server

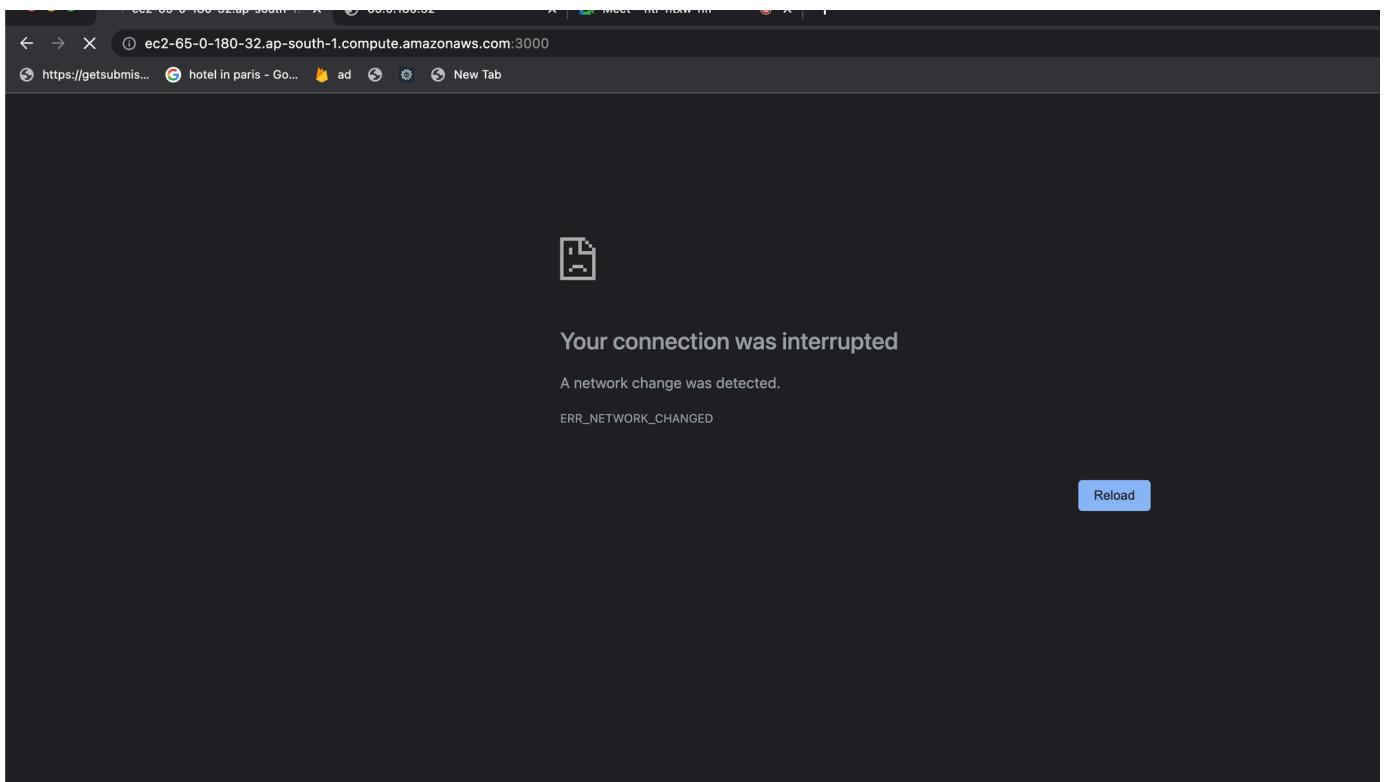
You have an ip/DNS that you can hit to access your ec2 server

Instance ID = i-0e4f854af3b210f99	X	Clear filters	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
			t2.micro	⌚ 2/2 checks passed	<a href="#">View alarms +</a>	ap-south-1b	ec2-65-0-180-32.ap-so...	65.0.180.32	-

Try visiting the backend



Notice you **can't** visit the website during this time



## Security group

Instance: i-0e4f854af3b210f99 (kirat-test-backend)

sg-02dfb75955cbab399 (launch-wizard-82)

Inbound rules

Name	Security group rule ID	Port range	Protocol	Source	Security groups	Description
-	sgr-0706ed28fc7d64005	80	TCP	::/0	<a href="#">launch-wizard-82</a>	-
-	sgr-00af5afb172ae42b3	22	TCP	0.0.0.0/0	<a href="#">launch-wizard-82</a>	-
-	sgr-03176ca91948cd3aa	22	TCP	::/0	<a href="#">launch-wizard-82</a>	-
-	sgr-0484c02d49ee84067	443	TCP	0.0.0.0/0	<a href="#">launch-wizard-82</a>	-
-	sgr-06a2f4a09be104504	80	TCP	0.0.0.0/0	<a href="#">launch-wizard-82</a>	-
-	sgr-04ddc60cb64ffd420	443	TCP	::/0	<a href="#">launch-wizard-82</a>	-

You can either open port 8080, or process on port 80

Inbound rules | Outbound rules | Tags

Inbound rules (6)

Name	Security group rule ID	Type	Protocol	Port range	Source
-	sgr-0706ed28fc7d64005	HTTP	TCP	80	::/0
-	sgr-00af5afb172ae42b3	SSH	TCP	22	0.0.0.0/0
-	sgr-03176ca91948cd3aa	IPv6	TCP	22	::/0
-	sgr-0484c02d49ee840...	IPv4	HTTPS	443	0.0.0.0/0
-	sgr-06a2f4a09be104504	IPv4	HTTP	80	0.0.0.0/0
-	sgr-04ddc60cb64ffd420	IPv6	HTTPS	443	::/0

Inbound rules [Info](#)

Security group rule ID	Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>
sgr-0ece62ac54b5c7518	Custom TCP	TCP	8080	Custom	0.0.0.0/0

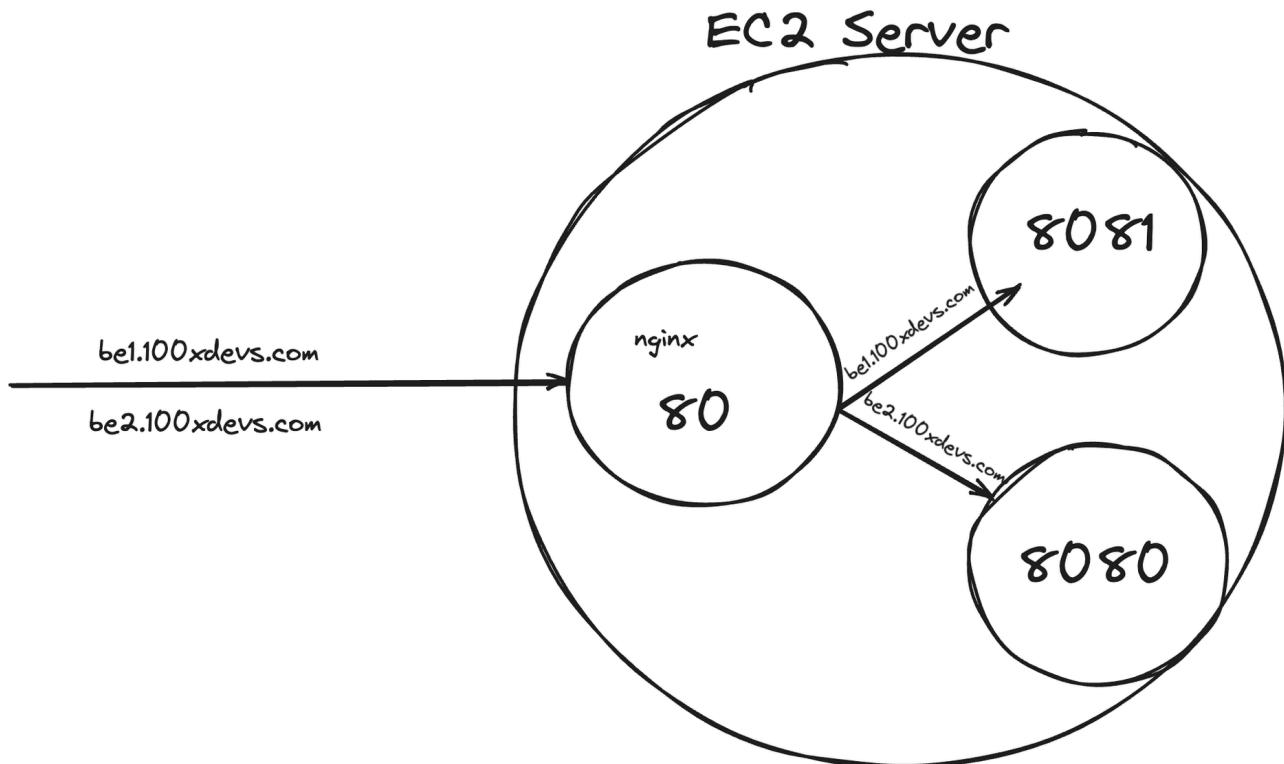
Copy [http://your\\_domain:8080](http://your_domain:8080)

# Step 7 - nginx

<https://www.nginx.com/resources/glossary/nginx/>

NGINX is open source software for web serving, reverse proxying, caching, load balancing, media streaming, and more. It started out as a web server designed for maximum performance and stability. In addition to its HTTP server capabilities, NGINX can also function as a proxy server for email (IMAP, POP3, and SMTP) and a reverse proxy and load balancer for HTTP, TCP, and UDP servers.

## What is a reverse proxy?

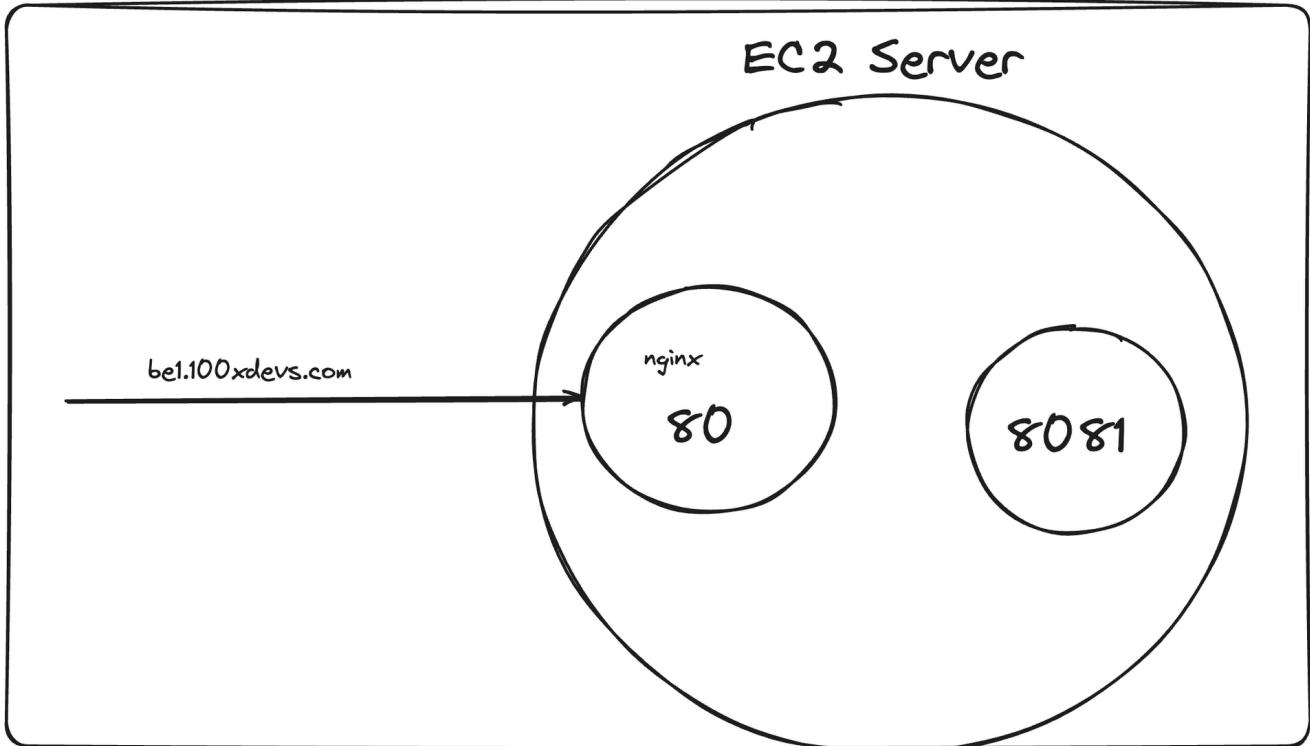


## Installing nginx

```
sudo apt update      Copy  
sudo apt install nginx
```

This should start a `nginx server` on port 80

Try visiting the website



## Create reverse proxy

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

```
events {  
    # Event directives...  
}  
  
http {  
    server {  
        listen 80;  
        server_name be1.100xdevs.com;  
  
        location / {  
            proxy_pass http://localhost:8080;  
            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;  
        }  
    }  
}
```

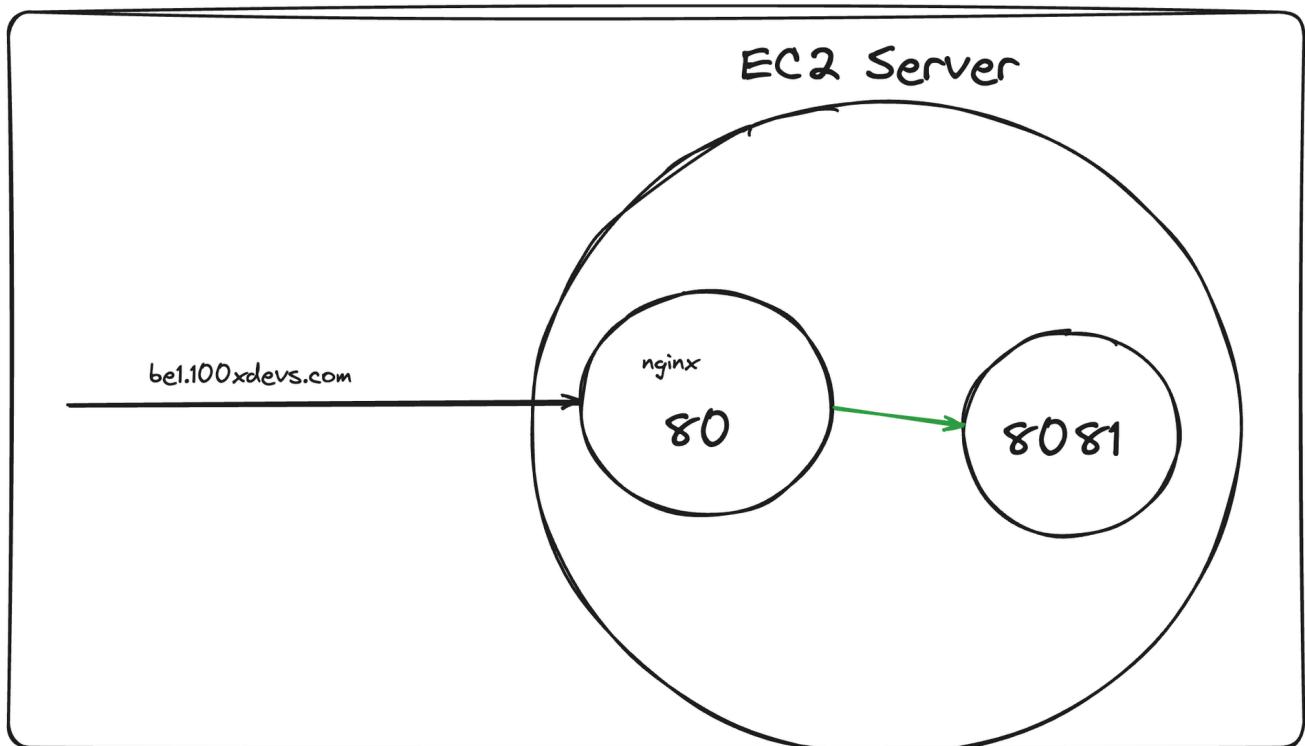
```
sudo nginx -s re...  
Copy
```

## Start the Backend server

```
node index.js  
Copy
```

## Visit the website

<https://be1.100xdevs.com>  
Copy



## Step 8 - Certificate management

Use <https://certbot.eff.org/>

