

Step 1 – What is AWS

aws



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

It let's 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**

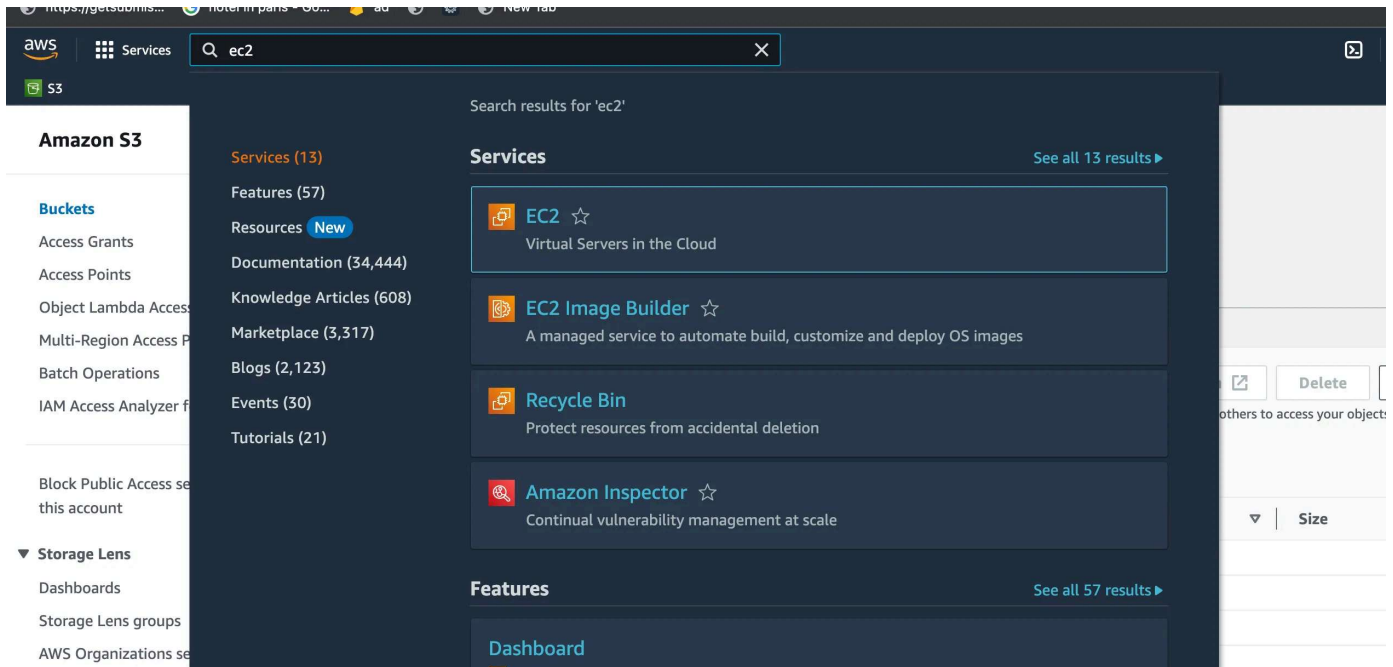
Step 2 – EC2 servers

VMs on AWS are called **EC2 Servers**

EC2 stands for Elastic compute Version 2.

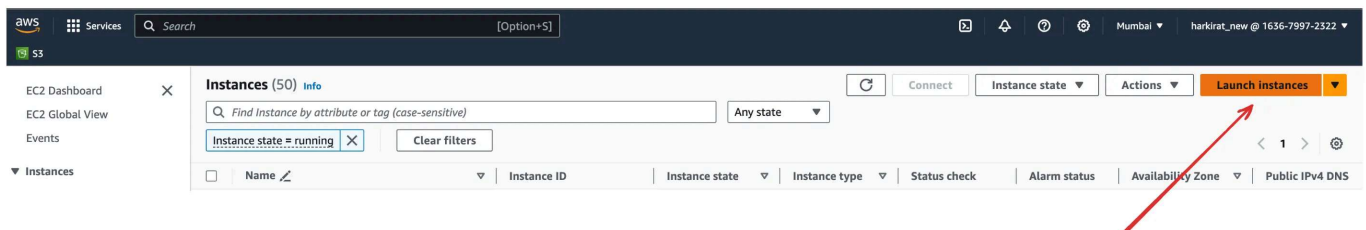
1. **Elastic** – Can increase/decrease the size of the machine
2. **Compute** – It is a machine

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

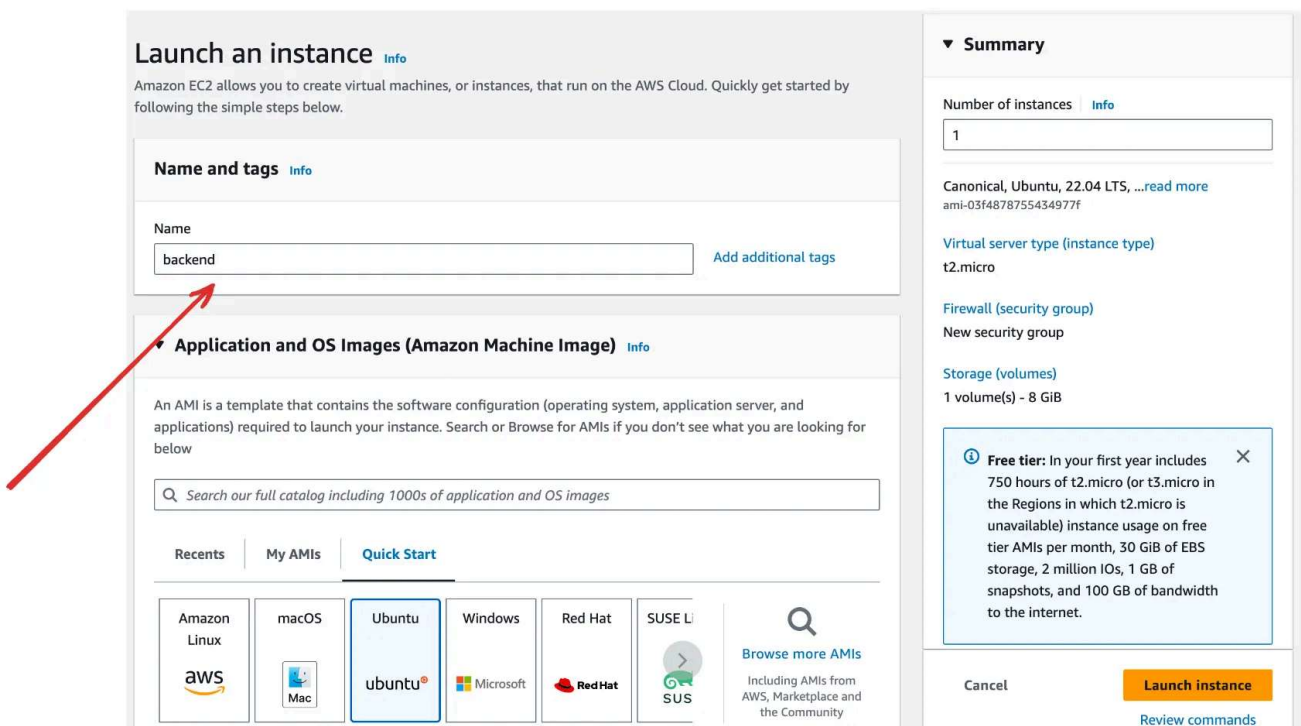


Step 3 – Creating a new EC2 server

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



2. Give a name



3. Select an OS

Hold Cmd and Double-click or press Cmd + Enter to edit points

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
aws

macOS
Mac

Ubuntu
buntu

Windows
Microsoft

Red Hat
Red Hat

SUSE Linux
SUS

[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

Get advice on instance type selection...

t2.nano

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

7. Allow traffic on http/https

Step 4 – SSH into server

1. Give ssh key permissions

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



2. ssh into machine

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



3. Clone repo

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



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/>

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
```



6. Start backend

```
node index.js
```



Step 5 – Install the repo

Clone the repo

<https://github.com/hkirat/sum-server>



Step 6 – Try hitting the server

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

Try visiting the backend

`your_domain:3000`



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

Security group

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

http://your_domain:8080



Step 7 – nginx

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

What is a reverse proxy?

Installing nginx

```
sudo apt update  
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  
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 reload
```



Start the Backend server

```
node index.js
```



Visit the website

```
https://be1.100xdevs.com/
```



Step 8 – Certificate management

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

