

DailyCode



AWS Deploying (EC2) (1 / 8)

Step 1 – What is AWS

The image shows the large, dark blue AWS logo in its signature sans-serif font.

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**



DailyCode



AWS Deploying (EC2) (2 / 8)

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 CloudFront service page. On the left, there's a sidebar with navigation links like 'Amazon S3', 'Buckets', 'Storage Lens', etc. The main area has a search bar at the top with 'ec2' typed in. Below it, there are two sections: 'Services' and 'Features'. The 'Services' section is expanded, showing items like EC2, EC2 Image Builder, Recycle Bin, and Amazon Inspector. The 'Features' section is collapsed, showing a 'Dashboard' link. The overall interface is dark-themed.



DailyCode



AWS Deploying (EC2) (3 / 8)

Step 3 – Creating a new EC2 server

1. Click on Launch a new instance

The screenshot shows the AWS EC2 Instances page. At the top, there's a search bar and a toolbar with various filters and actions. The main area displays 50 instances. A prominent red arrow points from the bottom-left towards the 'Launch instances' button in the toolbar.

2. Give a name

The screenshot shows the 'Launch an instance' wizard. It starts with a summary of the instance configuration, including the number of instances (1), AMI, instance type (t2.micro), security group, and storage. Below this is a detailed configuration section with tabs for 'Name and tags', 'Application and OS Images (Amazon Machine Image)', and 'Advanced options'. The 'Name and tags' tab has a 'Name' input field containing 'backend'. A red arrow points to this field. Another red arrow points to the 'Launch instance' button at the bottom right. A callout box on the right side provides information about the free tier.

3. Select an OS

The screenshot shows the AWS Lambda console interface. A red arrow points from the 'HelloWorld' function name in the top navigation bar down to the 'Code' tab, which displays the Lambda@Edge code for the function.

4. Select size

The screenshot shows the AWS Lambda console interface. A red arrow points from the 'HelloWorld' function name in the top navigation bar down to the 'Code' tab, which displays the Lambda@Edge code for the function.

5. Create a new Key pair

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

Select

Please choose a key pair or choose the option to proceed with a key pair

Create new key pair

Summary

Number of instances [Info](#)
1

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

Virtual server type (instance type)
t2.micro

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 [X](#)

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 [Edit](#)

0 x File systems

Advanced details [Info](#)

Summary

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. [X](#)

Cancel [Launch instance](#) Review commands

7. Allow traffic on http/https

Select ▼

[Create new key pair](#)

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

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

⚠️ 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. X

▼ 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. X

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

DailyCode



AWS Deploying (EC2) (4 / 8)

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/>

```
ubuntu@ip-172-31-11-253:~$ ping google.com
ping: google.com: Temporary failure in name resolution
ubuntu@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
```



6. Start backend

```
node index.js
```



DailyCode

A series of small, light-gray rounded rectangular icons with symbols: a left arrow, a square, another square, a sun-like icon, an empty square, a square with a downward arrow, and a circular profile picture of a person.

AWS Deploying (EC2) (5 / 8)

Step 5 – Install the repo

Clone the repo

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





DailyCode



AWS Deploying (EC2) (6 / 8)

Step 6 – Try hitting the server

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

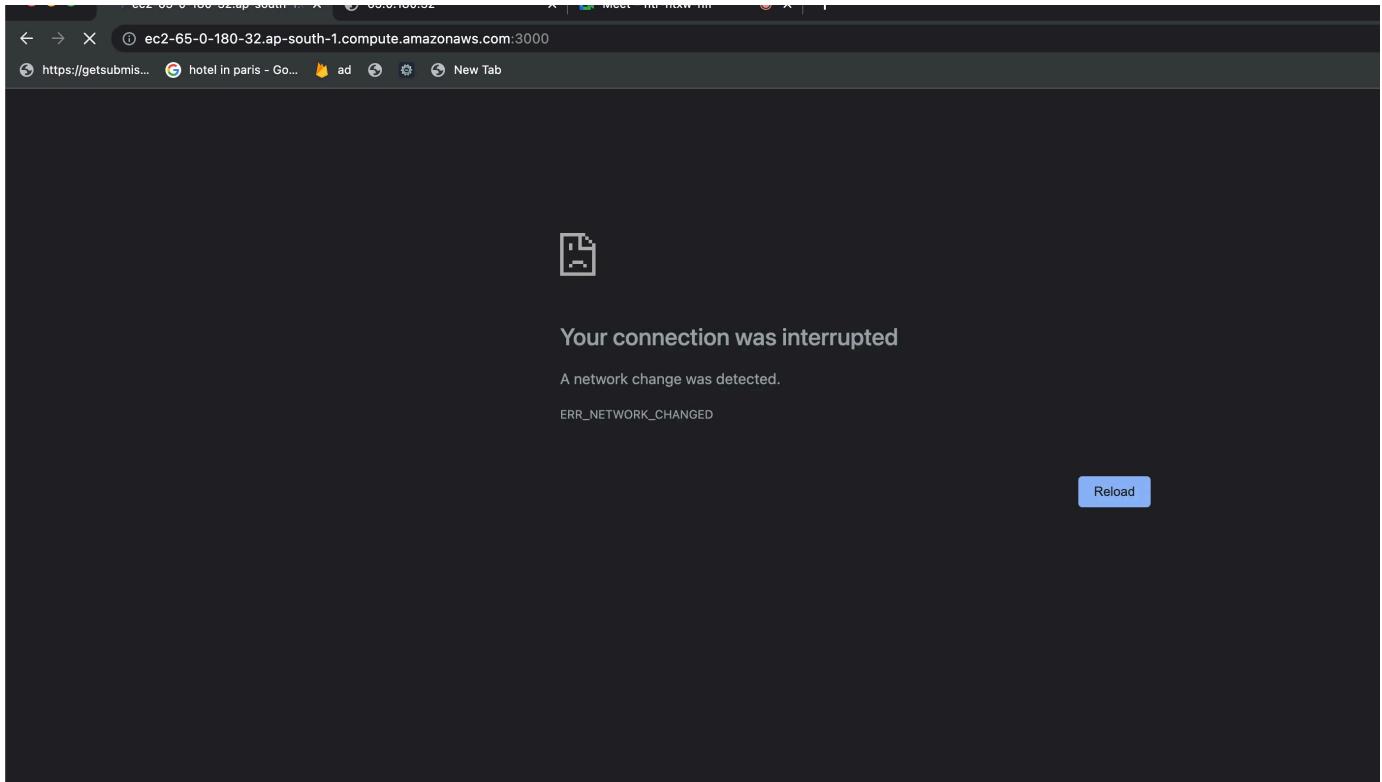
Instance ID = i-0e4f854af3b210f99		X	Clear filters								
▼	Instance type	▼	Status check	Alarm status	Availability Zone	▼	Public IPv4 DNS	▼	Public IPv4 ...	▼	Elastic IP
Q	t2.micro		✔ 2/2 checks passed	View alarms +	ap-south-1b		ec2-65-0-180-32.ap-so...		65.0.180.32		-

Try visiting the backend

your_domain:3000



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



Security group

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

Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic...
i-0e4f854af3b210f99	Running	t2.micro	2/2 checks passed	View alarms	ap-south-1b	ec2-65-0-180-32.ap-so...	65.0.180.32	-

Inbound rules

Name	Security group rule ID	Port range	Protocol	Source	Security groups	Description
-	sgr-0706ed28fc7d64005	80	TCP	::/0	launch-wizard-82	-
-	sgr-00af5afb172ae42b3	22	TCP	0.0.0.0/0	launch-wizard-82	-
-	sgr-03176ca91948cd3aa	22	TCP	::/0	launch-wizard-82	-
-	sgr-0484c02d49ee84067	443	TCP	0.0.0.0/0	launch-wizard-82	-
-	sgr-06a2f4a09be104504	80	TCP	0.0.0.0/0	launch-wizard-82	-
-	sgr-04ddc60cb64ffdd420	443	TCP	::/0	launch-wizard-82	-

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

Inbound rules		Outbound rules		Tags					
Inbound rules (6)									
<input type="text"/> Search									
	Name	Security group rule ID	IP version	Type	Protocol				
<input type="checkbox"/>	-	sgr-0706ed28fc7d64005	IPv6	HTTP	TCP				
<input type="checkbox"/>	-	sgr-00af5afb172ae42b3	IPv4	SSH	TCP				
<input type="checkbox"/>	-	sgr-03176ca91948cd3aa	IPv6	SSH	TCP				
<input type="checkbox"/>	-	sgr-0484c02d49ee840...	IPv4	HTTPS	TCP				
<input type="checkbox"/>	-	sgr-06a2f4a09be104504	IPv4	HTTP	TCP				
<input type="checkbox"/>	-	sgr-04ddc60cb64fd420	IPv6	HTTPS	TCP				

Inbound rules [Info](#)

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info
sgr-0ece62ac54b5c7518	Custom TCP <input type="button" value="▼"/>	TCP <input type="button" value="TCP"/>	8080 <input type="button" value="8080"/>	Custom <input type="button" value="▼"/>	<input type="text"/> 0.0.0.0/0 <input type="button" value="X"/>

http://your_domain:8080

DailyCode



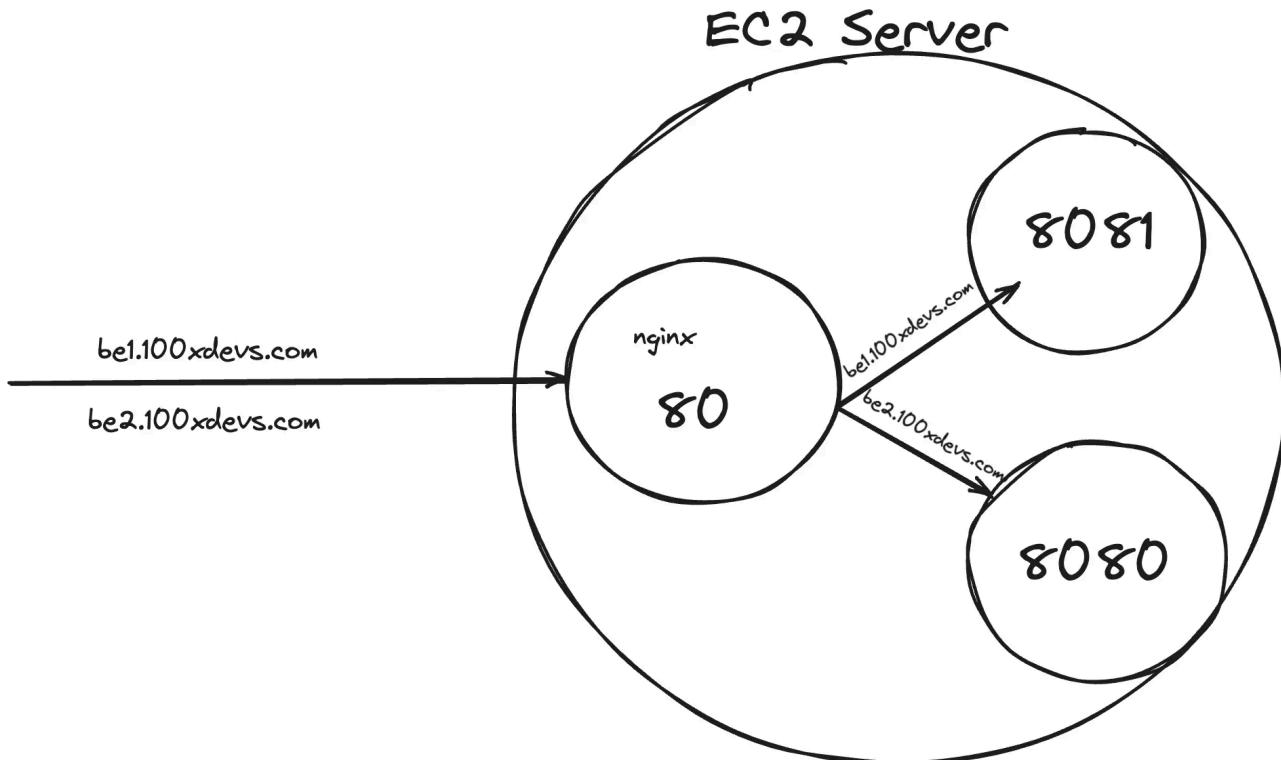
AWS Deploying (EC2) (7 / 8)

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?



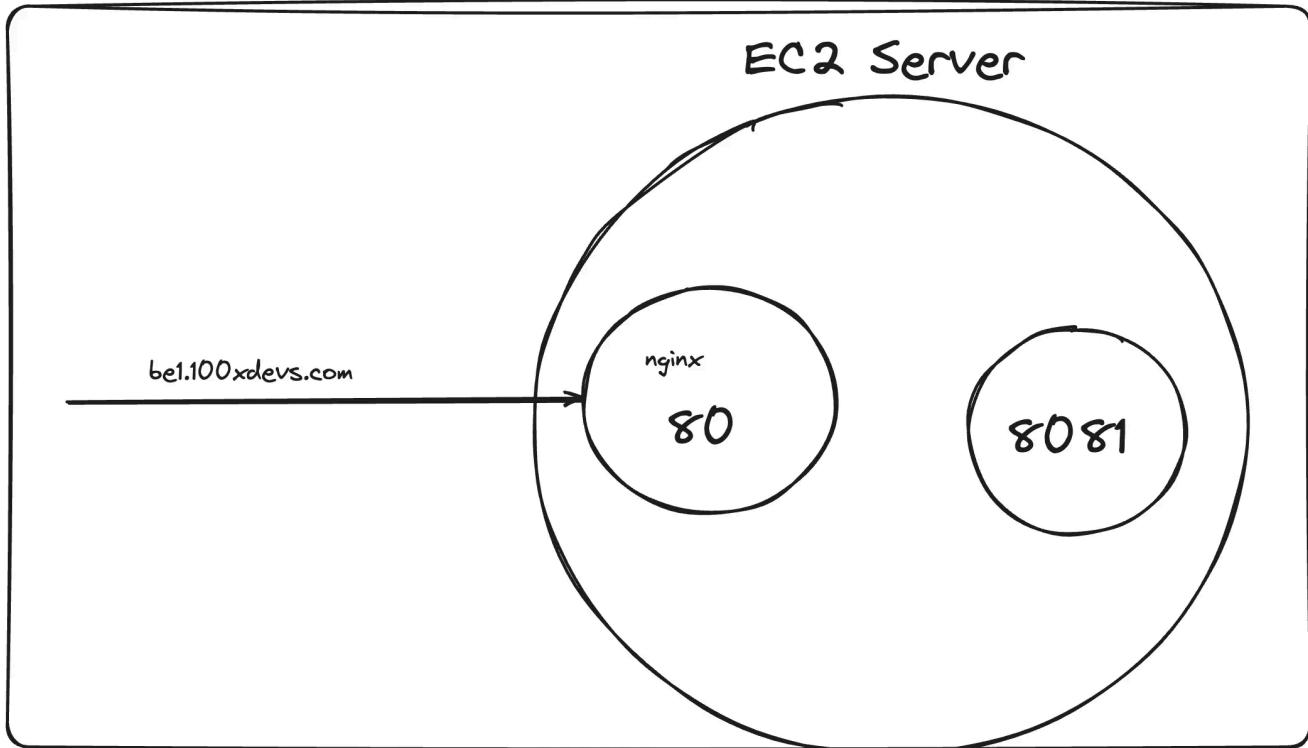
Installing nginx



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

This should start a **nginx server** on port 80

Try visiting the website



Create reverse proxy



```
sudo rm /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



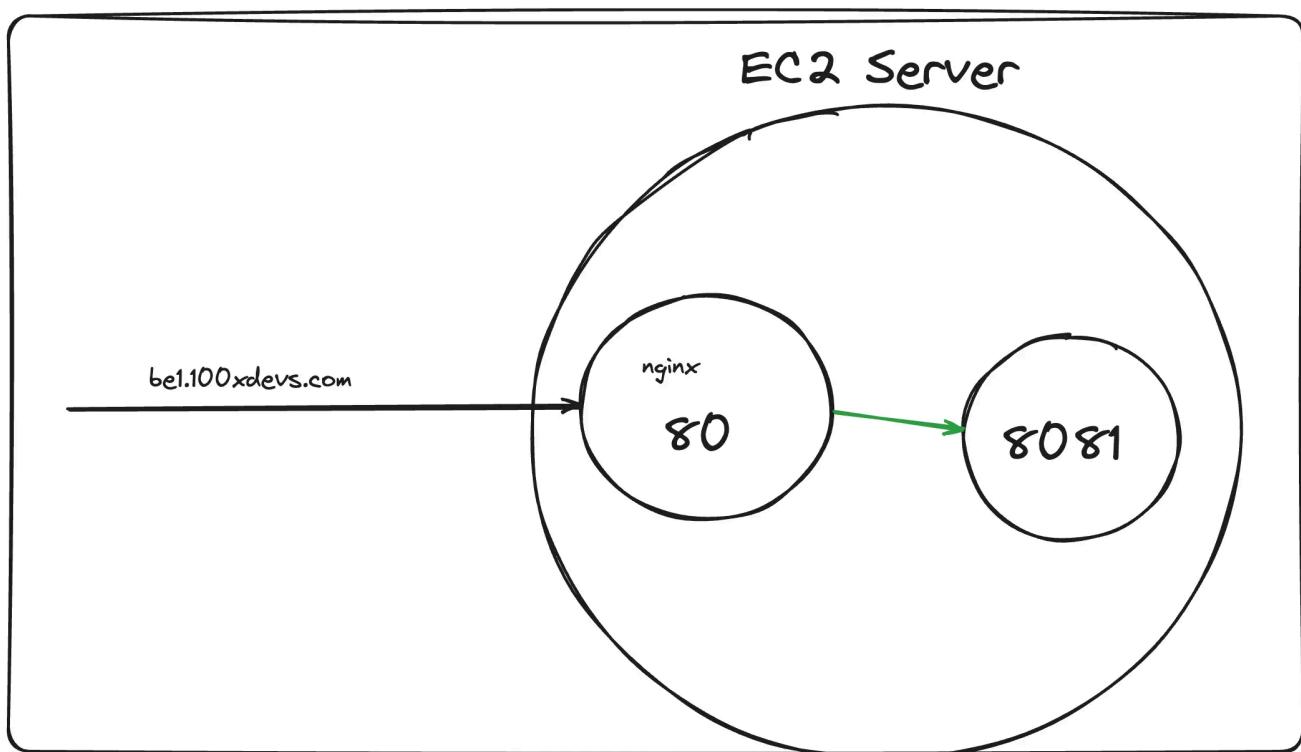
Start the Backend server

node index.js



Visit the website

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





DailyCode



AWS Deploying (EC2) (8 / 8)

Step 8 – Certificate management

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



