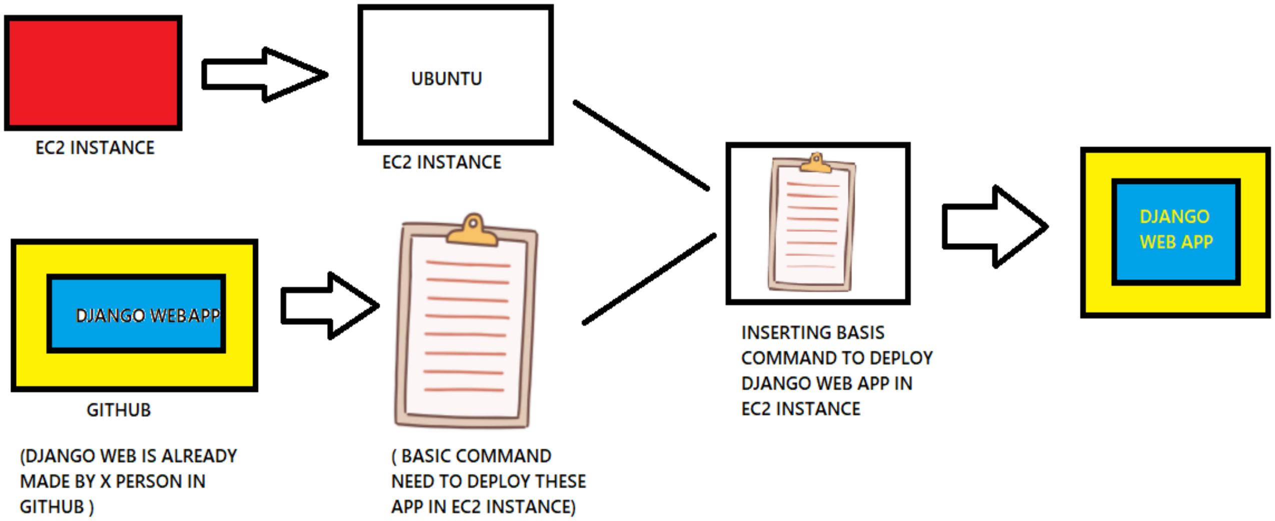
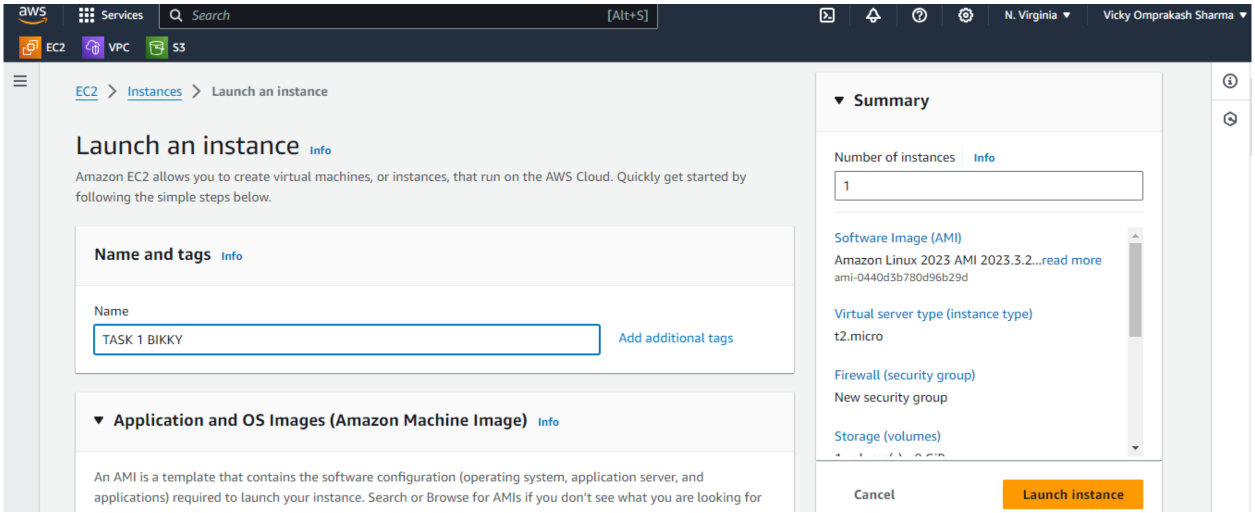
AIM- DEPLOYING DJANGO WEB APPLICATION USING EC2 INSTANCE

ARCHITECTURE:-


EC2 INSTANCE
UBUNTU
EC2 INSTANCE
DJANGO WEBAPP
GITHUB
(DJANGO WEB IS ALREADY MADE BY X PERSON IN GITHUB)
(BASIC COMMAND NEED TO DEPLOY THESE APP IN EC2 INSTANCE)
INSERTING BASIS COMMAND TO DEPLOY DJANGO WEB APP IN EC2 INSTANCE
DJANGO WEB APP

STEP 1-

CREATE AN EC2 INSTANCE FIRST



EC2 > Instances > Launch an instance

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

TASK 1 BIKKY

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

Summary

Number of instances Info

1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.3.2...read more
ami-0440d3b780d96b29d

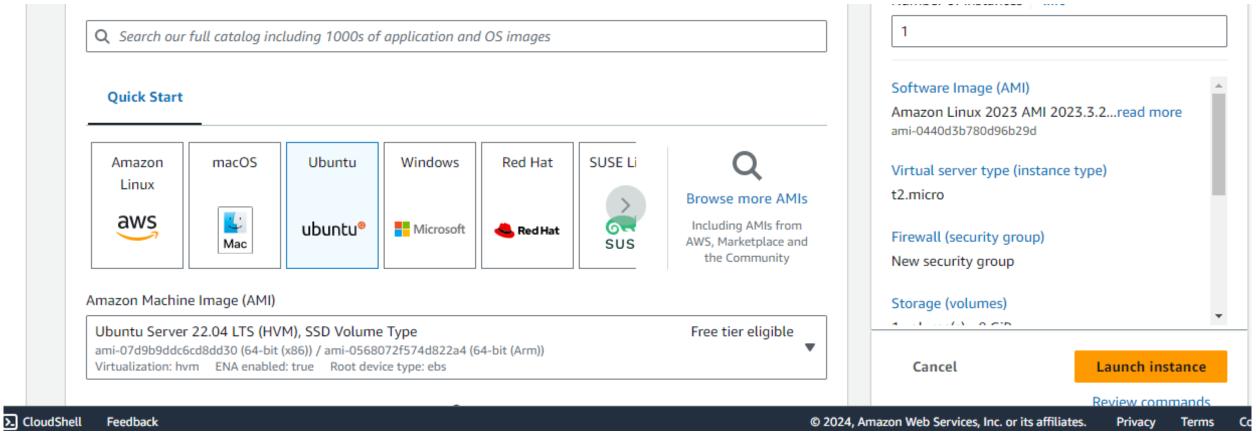
Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)

Cancel

Launch instance


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

Quick Start

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type
Free tier eligible
ami-07d9b9ddc6cd8dd30 (64-bit (x86)) / ami-0568072f574d822a4 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.3.2...read more
ami-0440d3b780d96b29d

Virtual server type (instance type)
t2.micro

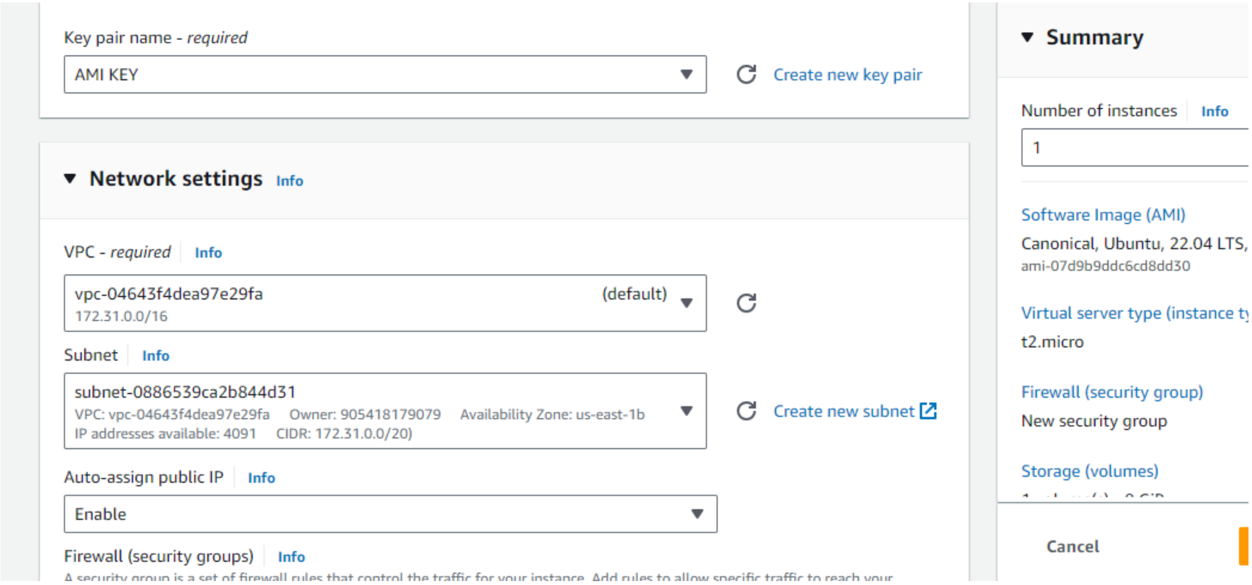
Firewall (security group)
New security group

Storage (volumes)

Cancel

Launch instance

Review commands


Key pair name - *required*

AMI KEY

Create new key pair

Network settings Info

VPC - *required* Info

vpc-04643f4dea97e29fa (default)
172.31.0.0/16

Subnet Info

subnet-0886539ca2b844d31
VPC: vpc-04643f4dea97e29fa Owner: 905418179079 Availability Zone: us-east-1b
IP addresses available: 4091 CIDR: 172.31.0.0/20)

Create new subnet

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

Summary

Number of instances Info

1

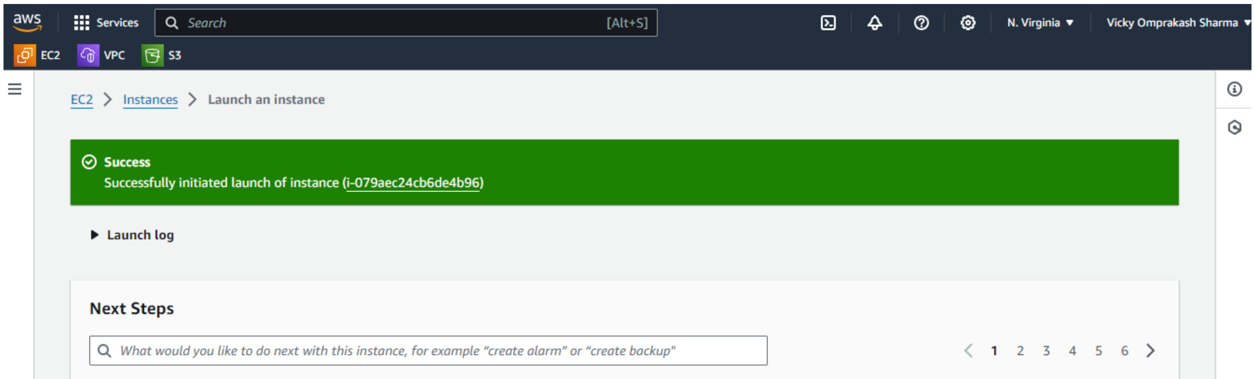
Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS,
ami-07d9b9ddc6cd8dd30

Virtual server type (instance t
t2.micro

Firewall (security group)
New security group

Storage (volumes)

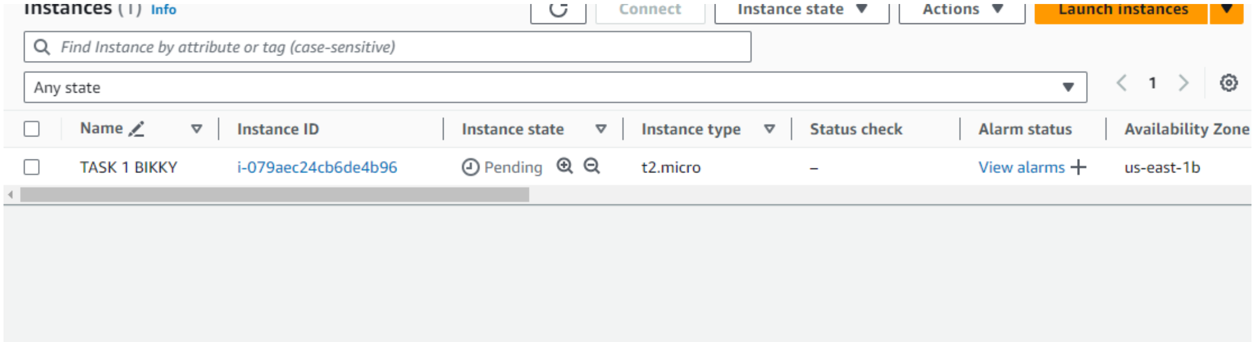
Cancel


aws
Services
Search
[Alt+S]
N. Virginia ▾
Vicky Omprakash Sharma ▾
EC2
VPC
S3
EC2 > Instances > Launch an instance

Success
Successfully initiated launch of instance (i-079aec24cb6de4b96)

Next Steps

What would you like to do next with this instance, for example "create alarm" or "create backup"
< 1 2 3 4 5 6 >



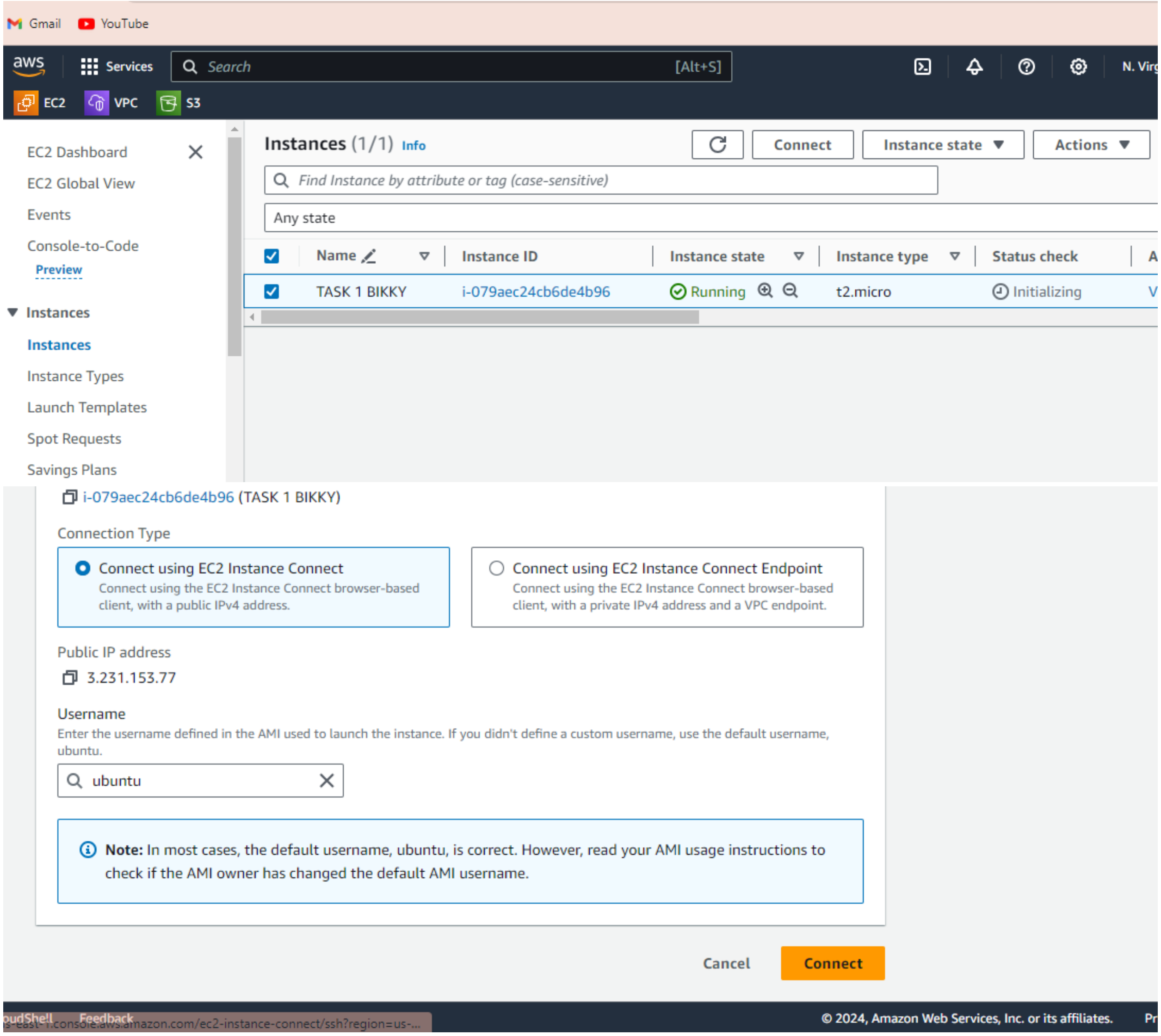
□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
□	TASK 1 BIKKY	i-079aec24cb6de4b96	Pending	t2.micro	–	View alarms +	us-east-1b

STEP 2- NOW GO TO GITHUB AND SELECT ONE ALREADY DJANGO WEP APP FOR OUR EC2 INSTANCE

https://github.com/yeshwanthlm/django-on-ec2

ABOVE IS THE LINK WHICH I SELECT TO DEPLOY IT

STEP-3 CONNECT YOUR INSTANCE



EC2 Dashboard
EC2 Global View
Events
Console-to-Code
Preview
▼ Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans

Instances (1/1) Info

Find Instance by attribute or tag (case-sensitive)

Any state

☑	Name	Instance ID	Instance state	Instance type	Status check	A
☑	TASK 1 BIKKY	i-079aec24cb6de4b96	Running	t2.micro	Initializing	V

i-079aec24cb6de4b96 (TASK 1 BIKKY)

Connection Type

● Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

○ Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address
3.231.153.77

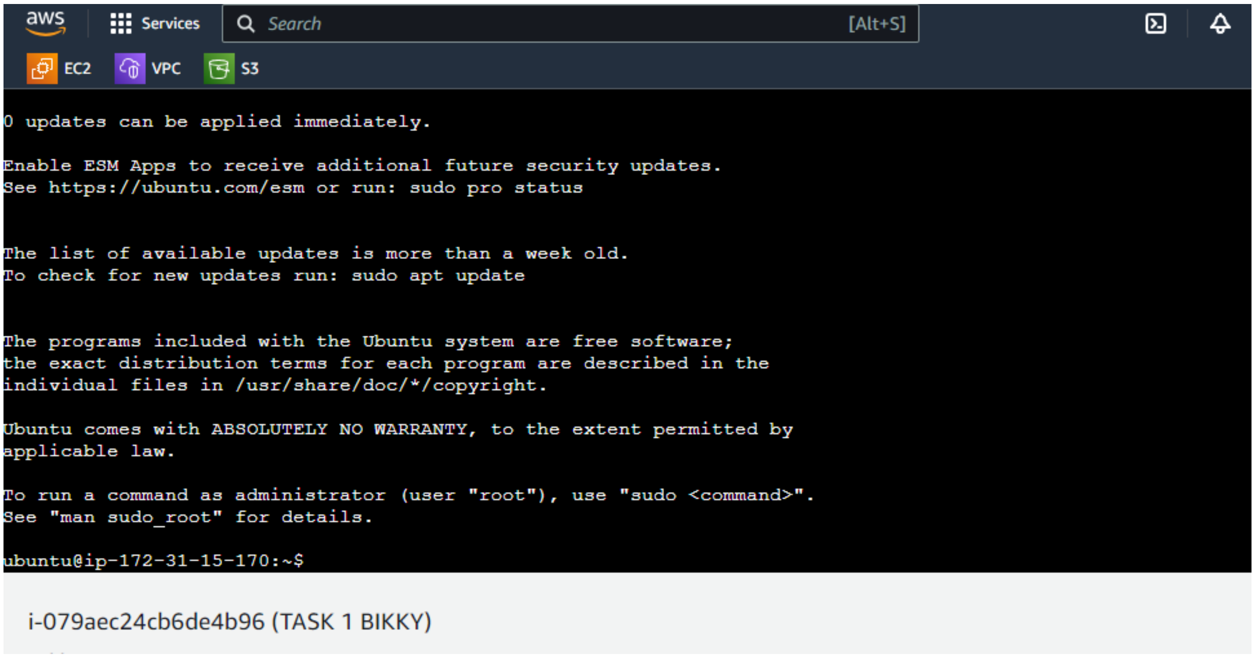
Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.

ubuntu

Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel
Connect
oudShell
Feedback
g-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-...

© 2024, Amazon Web Services, Inc. or its affiliates.

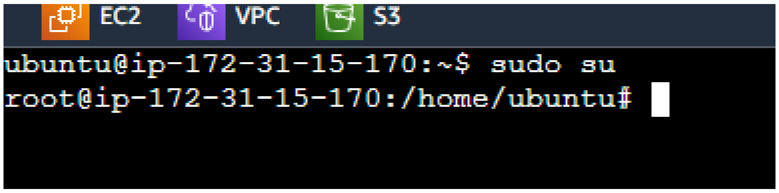


```
0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


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-172-31-15-170:~$
```

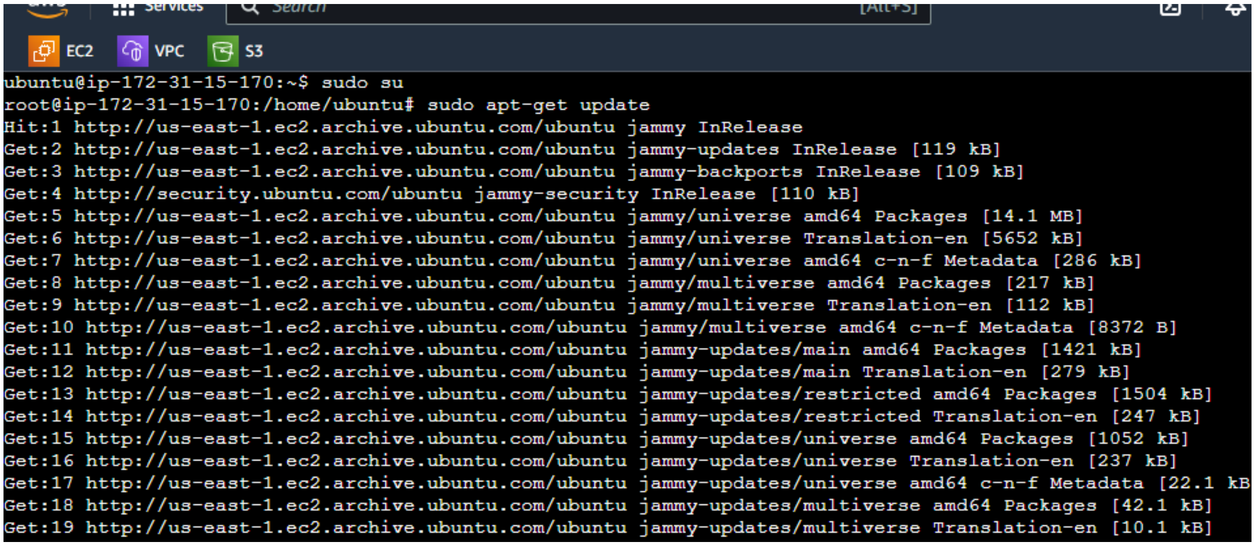
i-079aec24cb6de4b96 (TASK 1 BIKKY)

STEP-3 FIRST ALL OF UPDATE SYSTEM

COMMAND –apt-get update



```
ubuntu@ip-172-31-15-170:~$ sudo su
root@ip-172-31-15-170:/home/ubuntu#
```

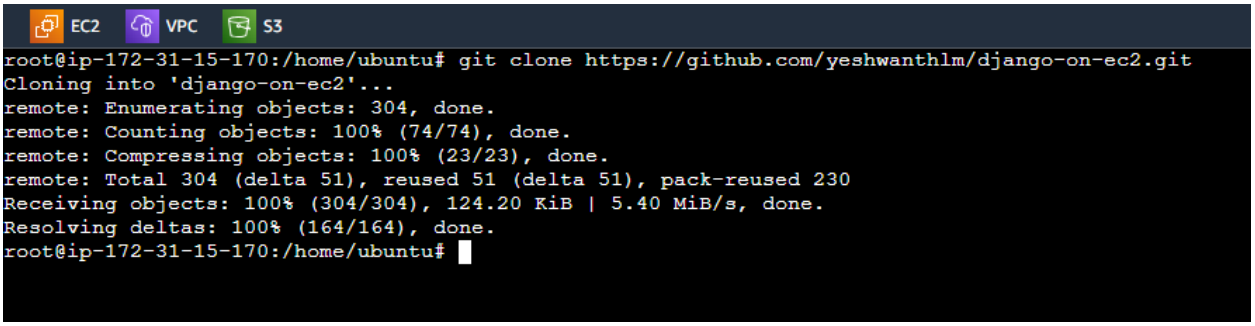


```
ubuntu@ip-172-31-15-170:~$ sudo su
root@ip-172-31-15-170:/home/ubuntu# sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1421 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [279 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1504 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [247 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1052 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [237 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [22.1 kB
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [42.1 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [10.1 kB]
```

Step 4- CLONE A REPOSITORY OF DJANGO WEB APP

COMMAND-

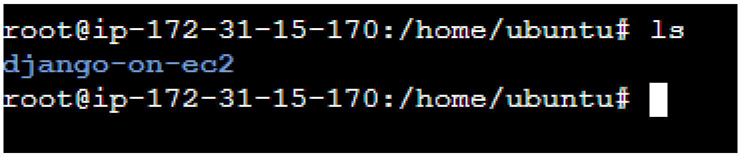
git clone https://github.com/yeshwanthlm/django-on-ec2.git



```
root@ip-172-31-15-170:/home/ubuntu# git clone https://github.com/yeshwanthlm/django-on-ec2.git
Cloning into 'django-on-ec2'...
remote: Enumerating objects: 304, done.
remote: Counting objects: 100% (74/74), done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 304 (delta 51), reused 51 (delta 51), pack-reused 230
Receiving objects: 100% (304/304), 124.20 KiB | 5.40 MiB/s, done.
Resolving deltas: 100% (164/164), done.
root@ip-172-31-15-170:/home/ubuntu#
```

Step-5

ENSURE THAT YOU ARE IN RIGHT DIRECTORIES OF YOUR PROJECT BU VERIFYING YOUR PROJECT NAME

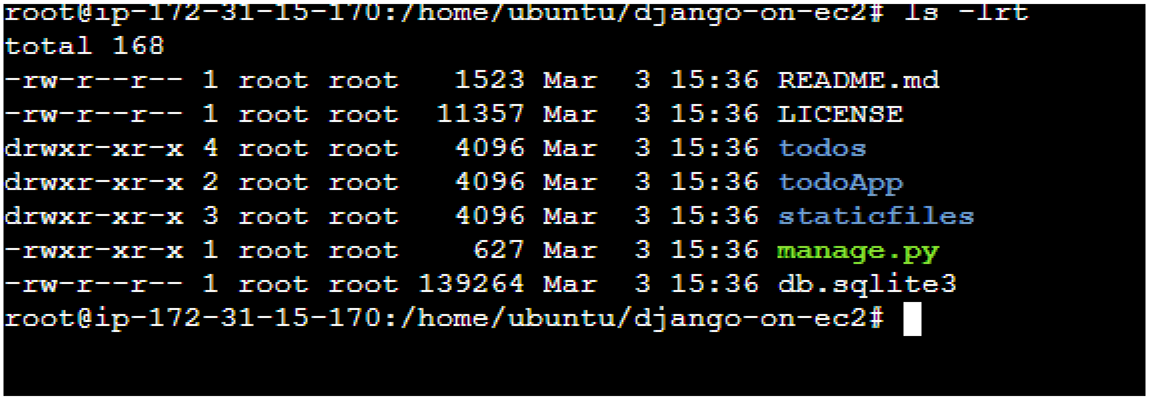


```
root@ip-172-31-15-170:/home/ubuntu# ls
django-on-ec2
root@ip-172-31-15-170:/home/ubuntu#
```

To check whether complete file detail

Command –cd Project Name

i.e cd Django-on-ec2



```
root@ip-172-31-15-170:/home/ubuntu/django-on-ec2# ls -lrt
total 168
-rw-r--r-- 1 root root   1523 Mar  3 15:36 README.md
-rw-r--r-- 1 root root  11357 Mar  3 15:36 LICENSE
drwxr-xr-x 4 root root   4096 Mar  3 15:36 todos
drwxr-xr-x 2 root root   4096 Mar  3 15:36 todoApp
drwxr-xr-x 3 root root   4096 Mar  3 15:36 staticfiles
-rwxr-xr-x 1 root root    627 Mar  3 15:36 manage.py
-rw-r--r-- 1 root root 139264 Mar  3 15:36 db.sqlite3
root@ip-172-31-15-170:/home/ubuntu/django-on-ec2# 
```

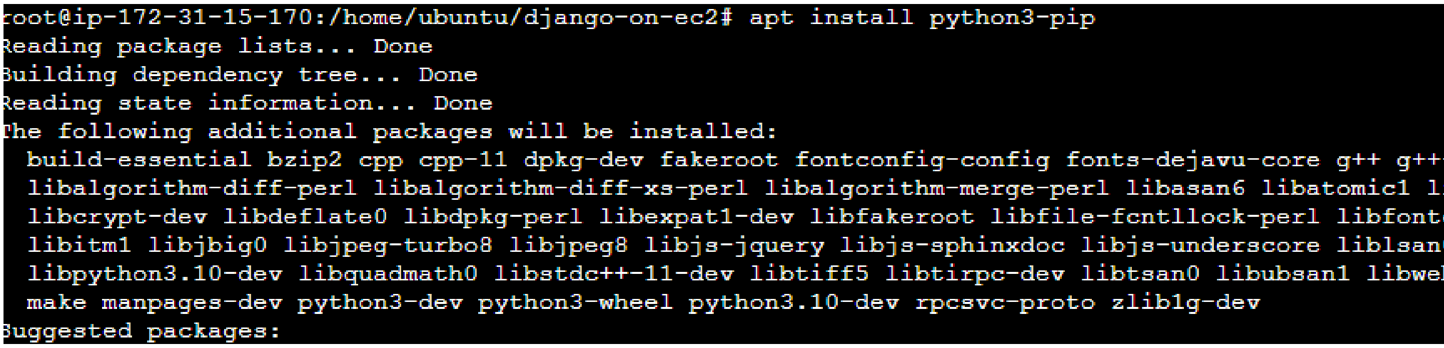
We successfully clone all file of Django web app in ec2 instance

Step 6-

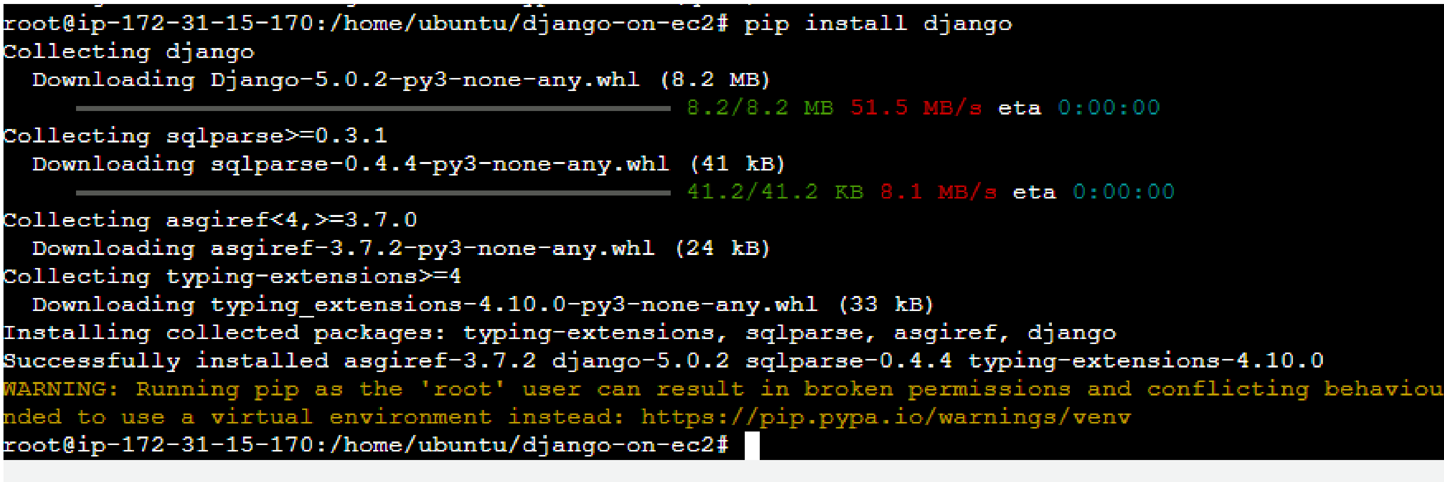
DOWNLOAD DJANGO USING PIP

COMMAND-1) apt install python3-pip

2)pip install django



```
root@ip-172-31-15-170:/home/ubuntu/django-on-ec2# apt install python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  build-essential bzip2 cpp cpp-11 dpkg-dev fakeroot fontconfig-config fonts-dejavu-core g++ g++-
  libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan6 libatomic1 l
  libcrypt-dev libdeflate0 libdpkg-perl libexpat1-dev libfakeroot libfile-fcntllock-perl libfont
  libitm1 libjbig0 libjpeg-turbo8 libjpeg8 libjs-jquery libjs-sphinxdoc libjs-underscore liblsan
  libpython3.10-dev libquadmath0 libstdc++-11-dev libtiff5 libtirpc-dev libtsan0 libubsan1 libwe
  make manpages-dev python3-dev python3-wheel python3.10-dev rpcsvc-proto zlib1g-dev
Suggested packages:
```

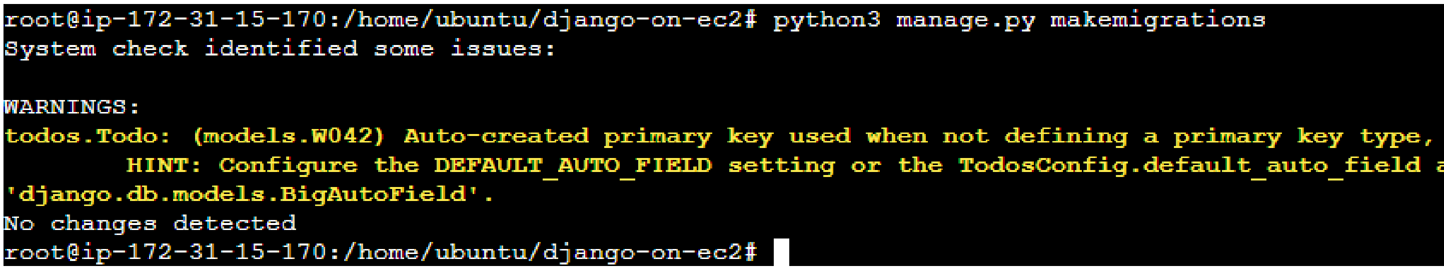


```
root@ip-172-31-15-170:/home/ubuntu/django-on-ec2# pip install django
Collecting django
  Downloading Django-5.0.2-py3-none-any.whl (8.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.2/8.2 MB 51.5 MB/s eta 0:00:00
Collecting sqlparse>=0.3.1
  Downloading sqlparse-0.4.4-py3-none-any.whl (41 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 41.2/41.2 KB 8.1 MB/s eta 0:00:00
Collecting asgiref<4,>=3.7.0
  Downloading asgiref-3.7.2-py3-none-any.whl (24 kB)
Collecting typing-extensions>=4
  Downloading typing_extensions-4.10.0-py3-none-any.whl (33 kB)
Installing collected packages: typing-extensions, sqlparse, asgiref, django
Successfully installed asgiref-3.7.2 django-5.0.2 sqlparse-0.4.4 typing-extensions-4.10.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviou
nded to use a virtual environment instead: https://pip.pypa.io/warnings/venv
root@ip-172-31-15-170:/home/ubuntu/django-on-ec2# 
```

STEP -7

Once you have downloaded django, go to the cloned repo directory and run the following command

Command- python3 manage.py makemigrations



```
root@ip-172-31-15-170:/home/ubuntu/django-on-ec2# python3 manage.py makemigrations
System check identified some issues:

WARNINGS:
todos.Todo: (models.W042) Auto-created primary key used when not defining a primary key type,
        HINT: Configure the DEFAULT_AUTO_FIELD setting or the TodosConfig.default_auto_field a
'django.db.models.BigAutoField'.
No changes detected
root@ip-172-31-15-170:/home/ubuntu/django-on-ec2# 
```

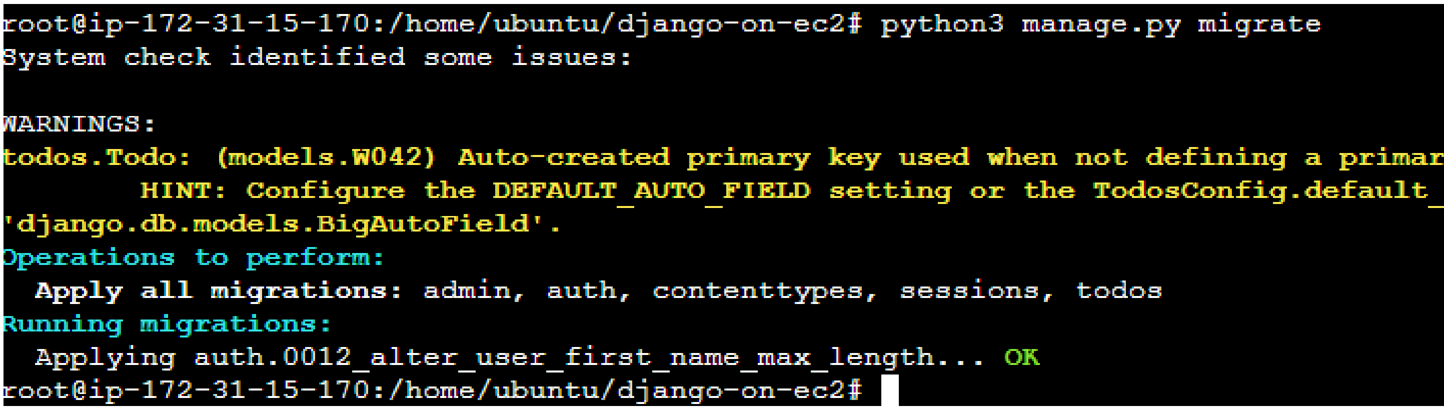
This will create all the migrations file (database migrations) required to run this App.

STEP 9-

ONCE YOU CREATE ALL MIGRATION FILE WHICH NEED TO RUN AN APPLICATION SO AFTER THAT NOW APPLY THAT CREATED MIGRATE FILE

TO APPLY THE MIGRATE FILE RUN COMMAND

COMMAND-



```
root@ip-172-31-15-170:/home/ubuntu/django-on-ec2# python3 manage.py migrate
System check identified some issues:

WARNINGS:
todos.Todo: (models.W042) Auto-created primary key used when not defining a primar
        HINT: Configure the DEFAULT_AUTO_FIELD setting or the TodosConfig.default_
'django.db.models.BigAutoField'.
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions, todos
Running migrations:
  Applying auth.0012_alter_user_first_name_max_length... OK
root@ip-172-31-15-170:/home/ubuntu/django-on-ec2#
```

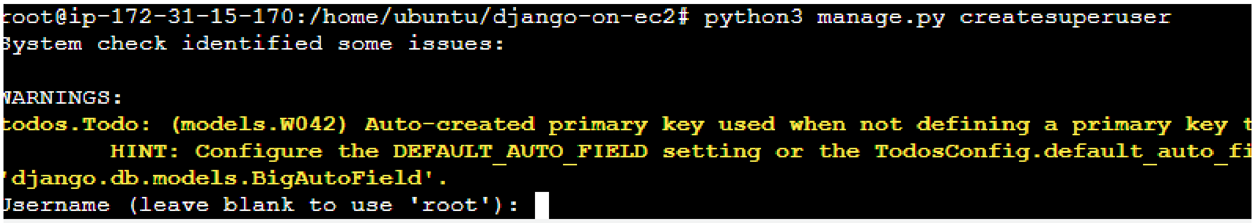
STEP 10-

NOW ALMOST WE DONE 80% PROCESS BUT FOR OUR WEB APP

WE NEED TO CREATE ADMIN USER TO RUN THESE APP

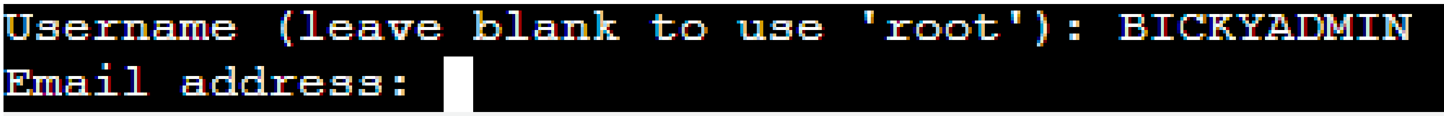
SO CREATE A SUPERUSER NOW

COMMAND - python3 manage.py createsuperuser



```
root@ip-172-31-15-170:/home/ubuntu/django-on-ec2# python3 manage.py createsuperuser
System check identified some issues:

WARNINGS:
todos.Todo: (models.W042) Auto-created primary key used when not defining a primary key t
        HINT: Configure the DEFAULT_AUTO_FIELD setting or the TodosConfig.default_auto_fi
'django.db.models.BigAutoField'.
Username (leave blank to use 'root'):
```

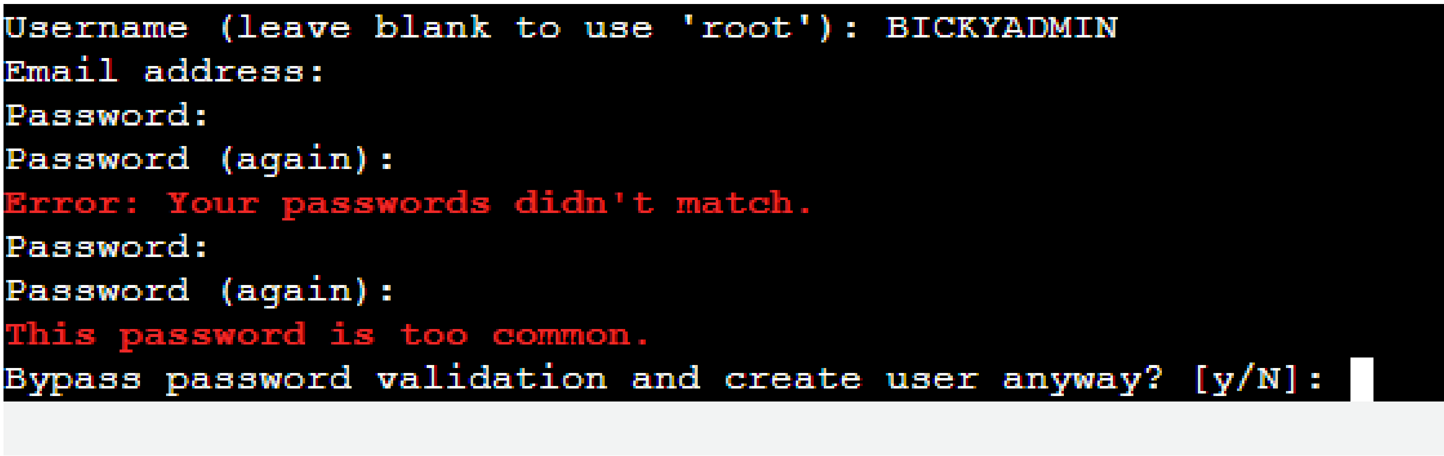
ENTER A PARICULAR USERNAME LET IT BE



```
Username (leave blank to use 'root'): BICKYADMIN
Email address:
```

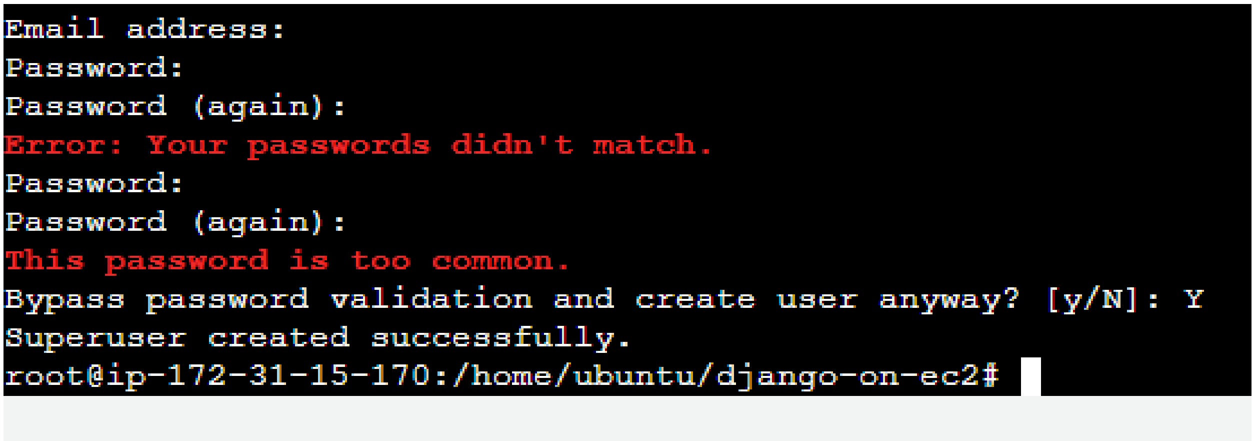
NOW GIVE AN EMAIL if you WANT OTHERWISE PRESS ENTER

AND SET PASSWORD



```
Username (leave blank to use 'root'): BICKYADMIN
Email address:
Password:
Password (again):
Error: Your passwords didn't match.
Password:
Password (again):
This password is too common.
Bypass password validation and create user anyway? [y/N]:
```

SELECT Y

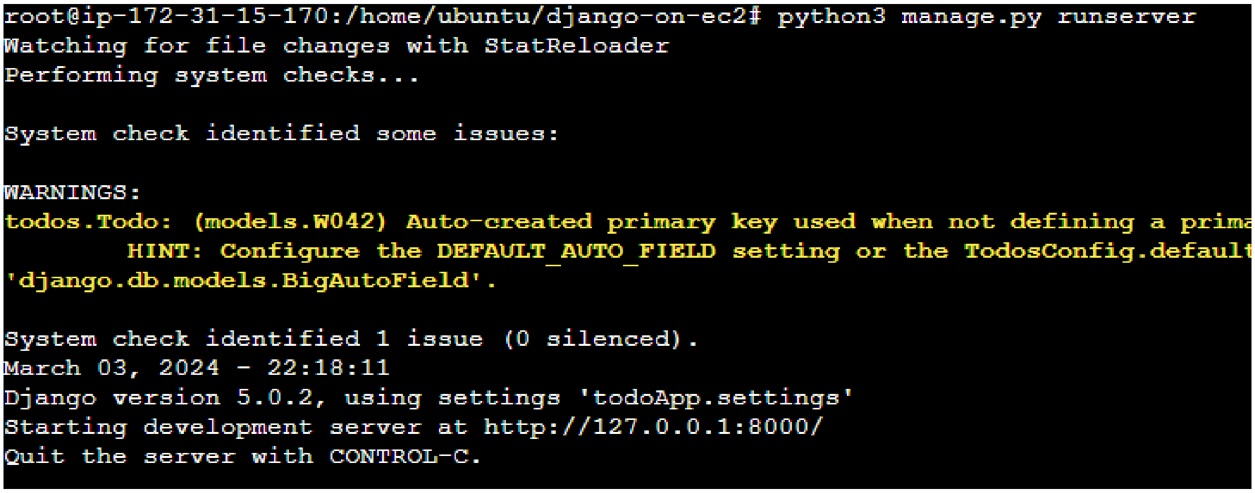


```
Email address:
Password:
Password (again):
Error: Your passwords didn't match.
Password:
Password (again):
This password is too common.
Bypass password validation and create user anyway? [y/N]: Y
Superuser created successfully.
root@ip-172-31-15-170:/home/ubuntu/django-on-ec2#
```

Step 11-

Now simply start the server so we can connect our app live

Command- python3 manage.py runserver



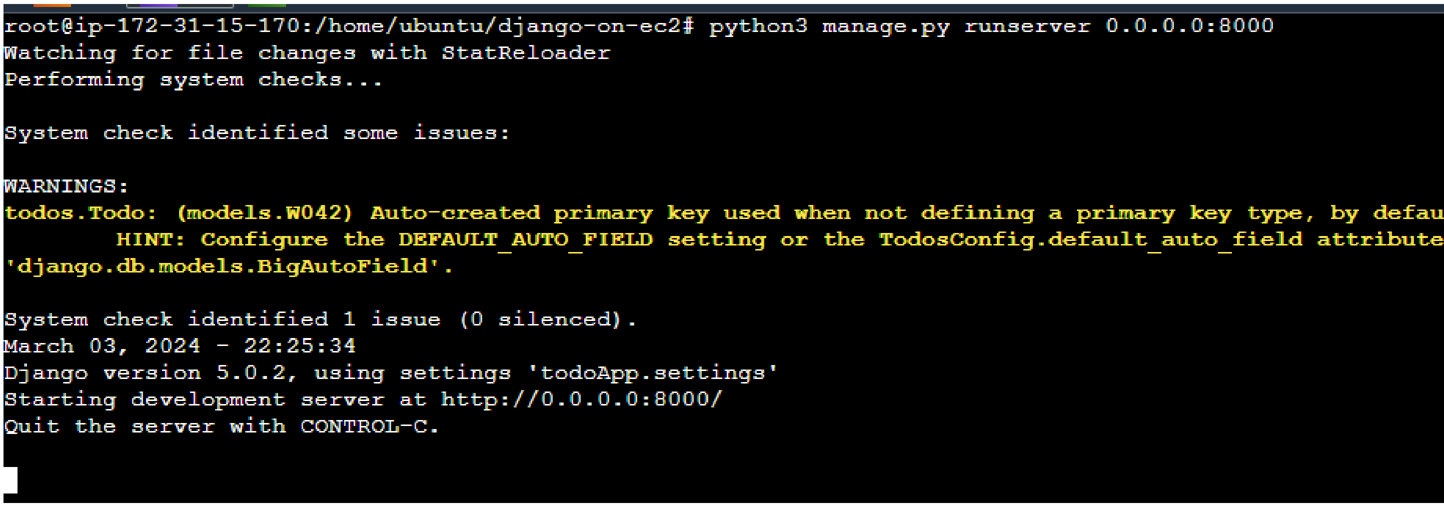
```
root@ip-172-31-15-170:/home/ubuntu/django-on-ec2# python3 manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified some issues:

WARNINGS:
todos.Todo: (models.W042) Auto-created primary key used when not defining a prima
        HINT: Configure the DEFAULT_AUTO_FIELD setting or the TodosConfig.default
'django.db.models.BigAutoField'.

System check identified 1 issue (0 silenced).
March 03, 2024 - 22:18:11
Django version 5.0.2, using settings 'todoApp.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

WE SUCCESSFULLY RUN OUR SERVER BUT HERE WE CONNECT TO PARTICULAR HOST WHICH IS 127.0.0.1/8000

SO WE NEED TO MAKE IT 0.0.0.0/8000

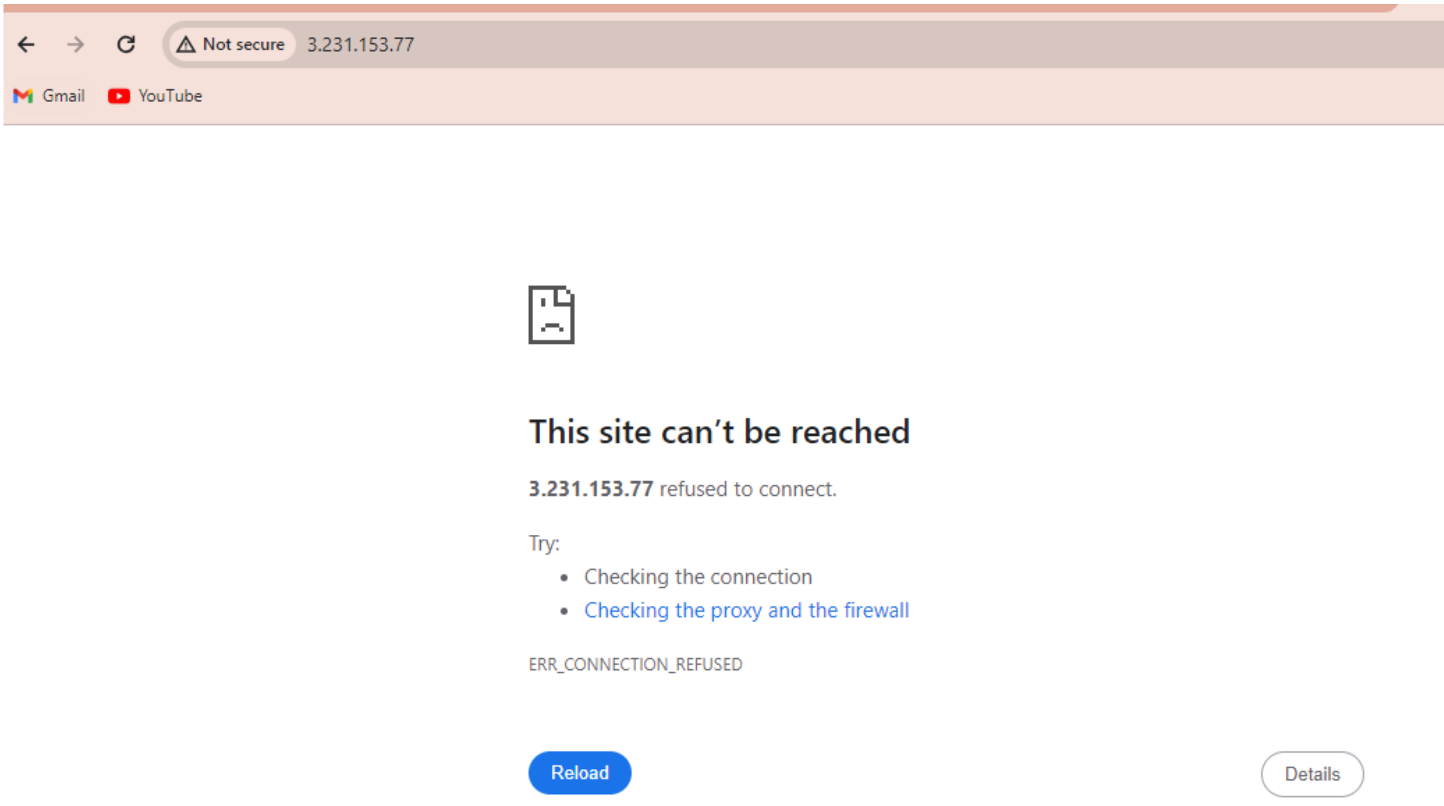
FOR THESE CONFIGURE AGAIN

COMMAND - python3 manage.py runserver 0.0.0.0:8000



```
root@ip-172-31-15-170:/home/ubuntu/django-on-ec2# python3 manage.py runserver 0.0.0.0:8000
Watching for file changes with StatReloader
Performing system checks...

System check identified some issues:

WARNINGS:
todos.Todo: (models.W042) Auto-created primary key used when not defining a primary key type, by defau
        HINT: Configure the DEFAULT_AUTO_FIELD setting or the TodosConfig.default_auto_field attribute
'django.db.models.BigAutoField'.

System check identified 1 issue (0 silenced).
March 03, 2024 - 22:25:34
Django version 5.0.2, using settings 'todoApp.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.
```

Now we are ready to go and see our deploy django web app TODO



This site can't be reached

3.231.153.77 refused to connect.

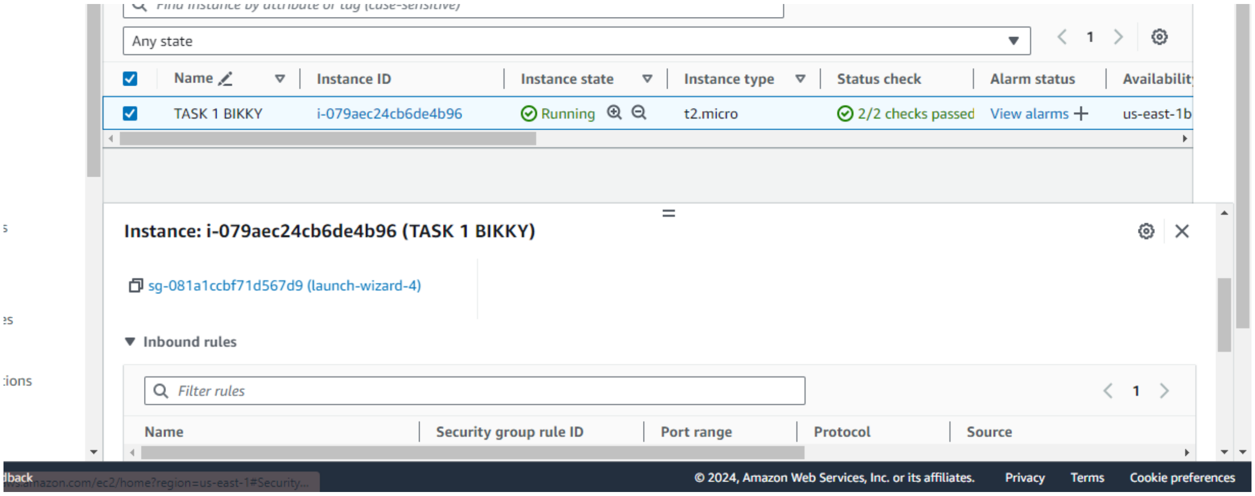
Try:

- Checking the connection
- Checking the proxy and the firewall

ERR_CONNECTION_REFUSED

\

HERE WE GOT OUR FIRST ERROR NOW TO REOLVE THESE GO AND EDIT INBOUND SECURITY OF INSTANT



Any state

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availabilit
☑	TASK 1 BIKKY	i-079aec24cb6de4b96	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b

Instance: i-079aec24cb6de4b96 (TASK 1 BIKKY)

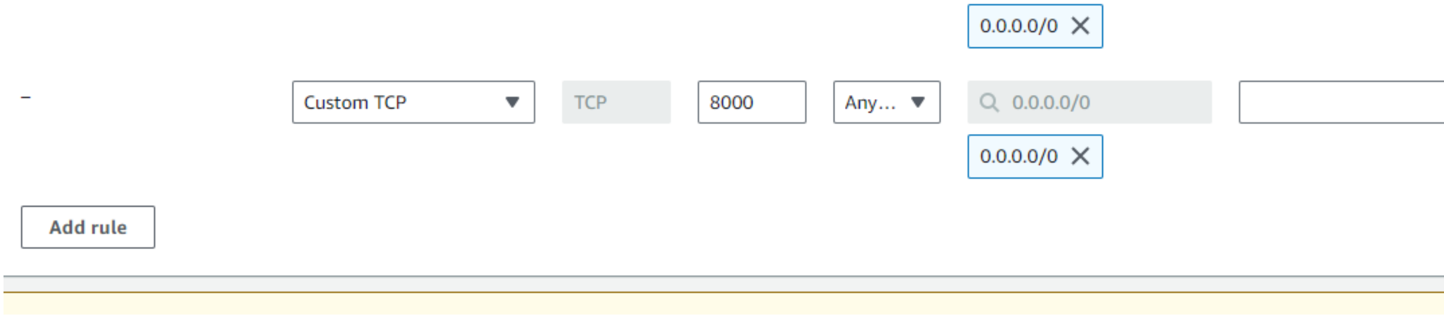
sg-081a1ccbf71d567d9 (launch-wizard-4)

▼ Inbound rules

Name	Security group rule ID	Port range	Protocol	Source

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

ADD THESE CUSTOM PORT TO ALLOW TRAFFIC



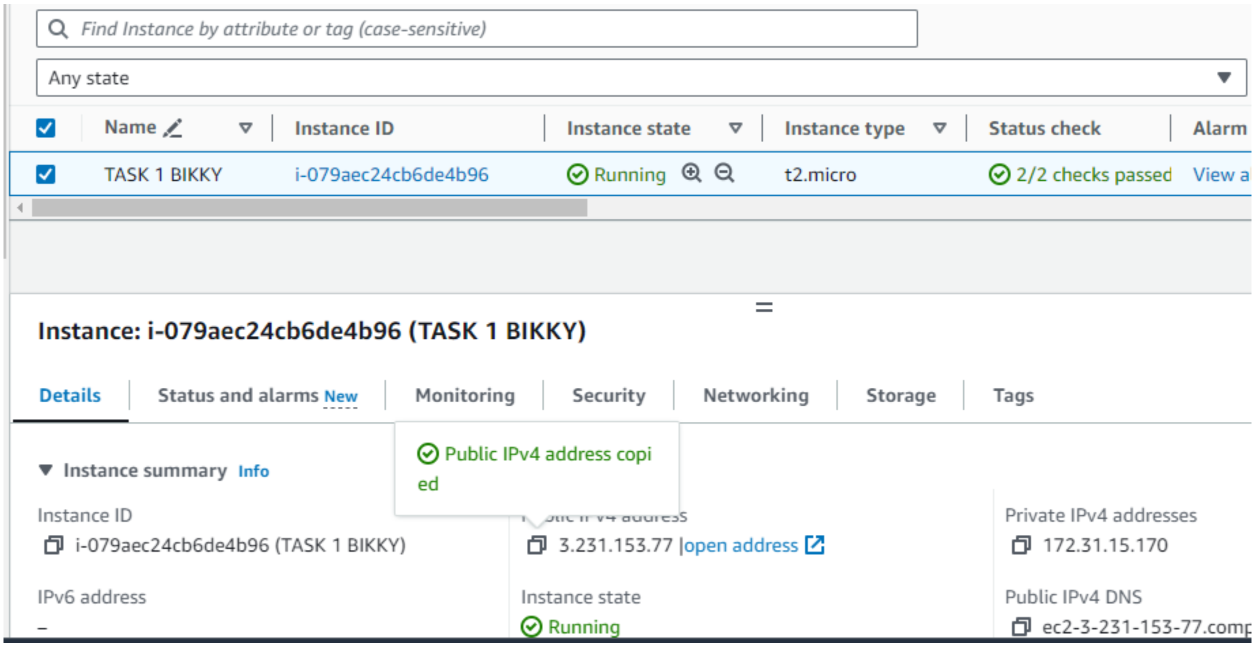
0.0.0.0/0 ✕

–	Custom TCP ▼	TCP	8000	Any… ▼	0.0.0.0/0

0.0.0.0/0 ✕

Add rule

NOW CHECK BY COPYING PUBLIC IP



Name	Instance ID	Instance state	Instance type	Status check	Alarm
TASK 1 BIKKY	i-079aec24cb6de4b96	Running	t2.micro	2/2 checks passed	View a

Instance: i-079aec24cb6de4b96 (TASK 1 BIKKY)

Public IPv4 address copied

Instance summary Info

Instance ID: i-079aec24cb6de4b96 (TASK 1 BIKKY)

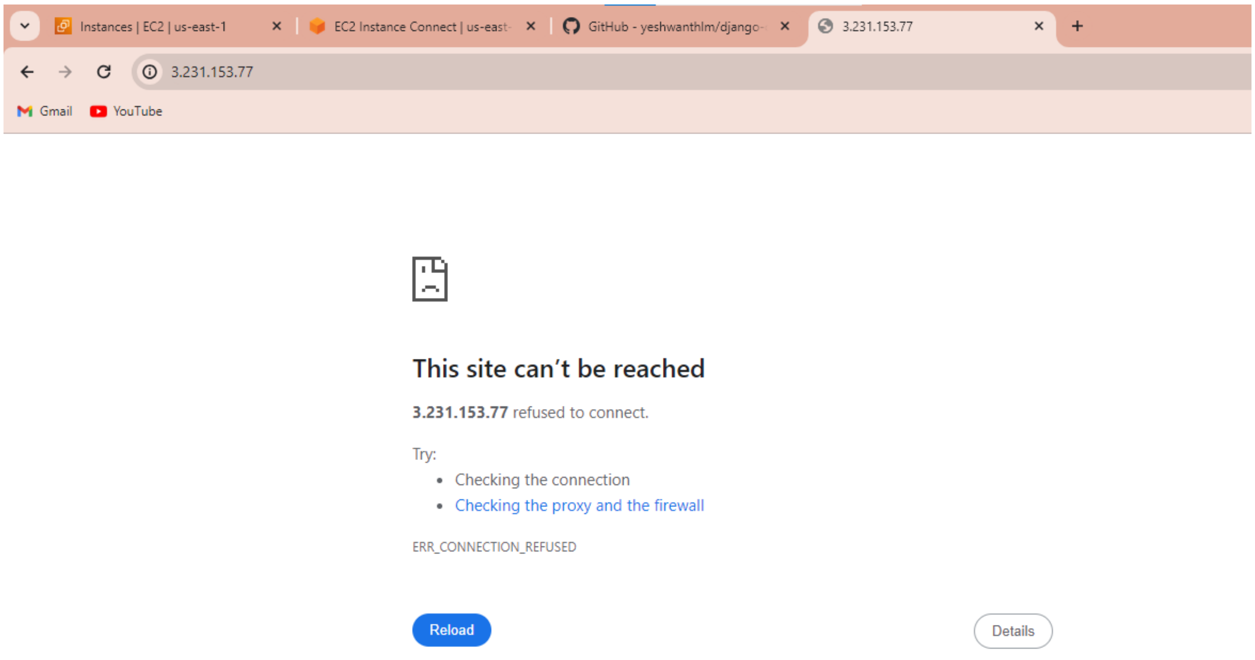
Public IPv4 address: 3.231.153.77 |open address

Private IPv4 addresses: 172.31.15.170

IPv6 address: –

Instance state: Running

Public IPv4 DNS: ec2-3-231-153-77.comp



Instances | EC2 | us-east-1
EC2 Instance Connect | us-east-
GitHub - yeshwanthlm/django-
3.231.153.77
Gmail
YouTube

3.231.153.77

This site can’t be reached

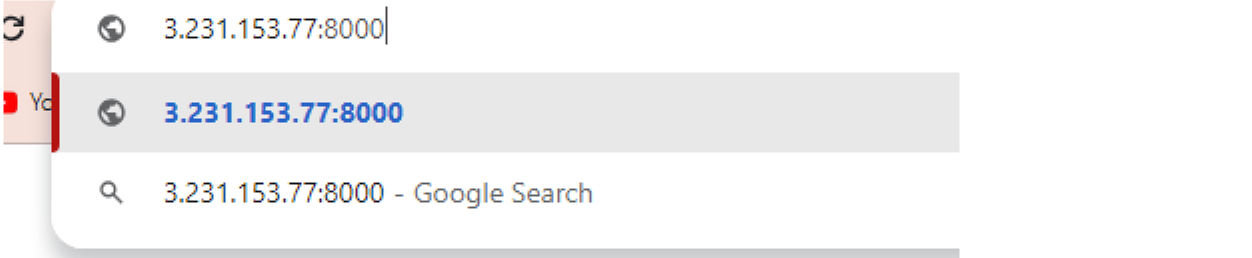
3.231.153.77 refused to connect.

Try:

- Checking the connection
- Checking the proxy and the firewall

ERR_CONNECTION_REFUSED

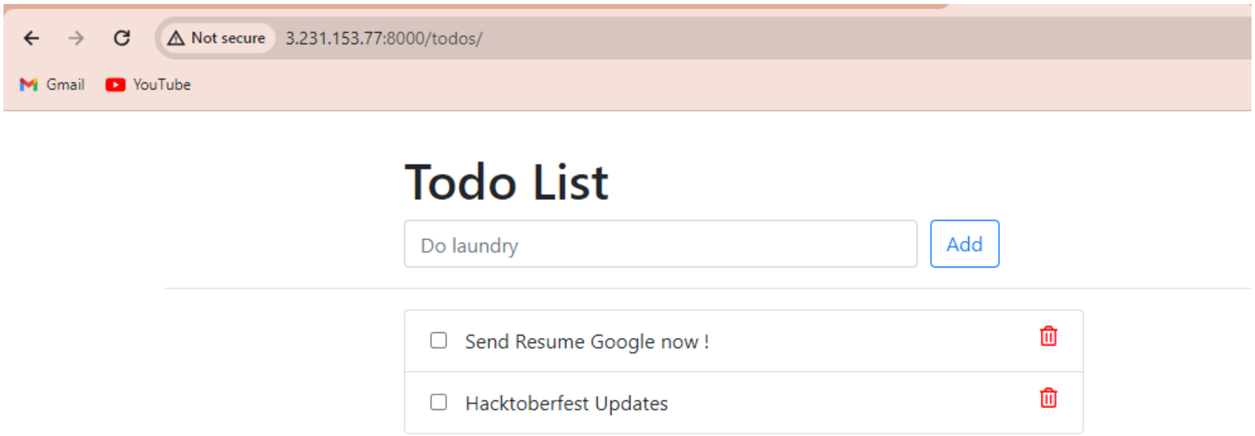
BUT STILL IT’S NOT WORKING SO MANAGE PUBLIC IP AND ADD YOUR CONFIRGURE PORT NUMBER



3.231.153.77:8000

3.231.153.77:8000

3.231.153.77:8000 - Google Search

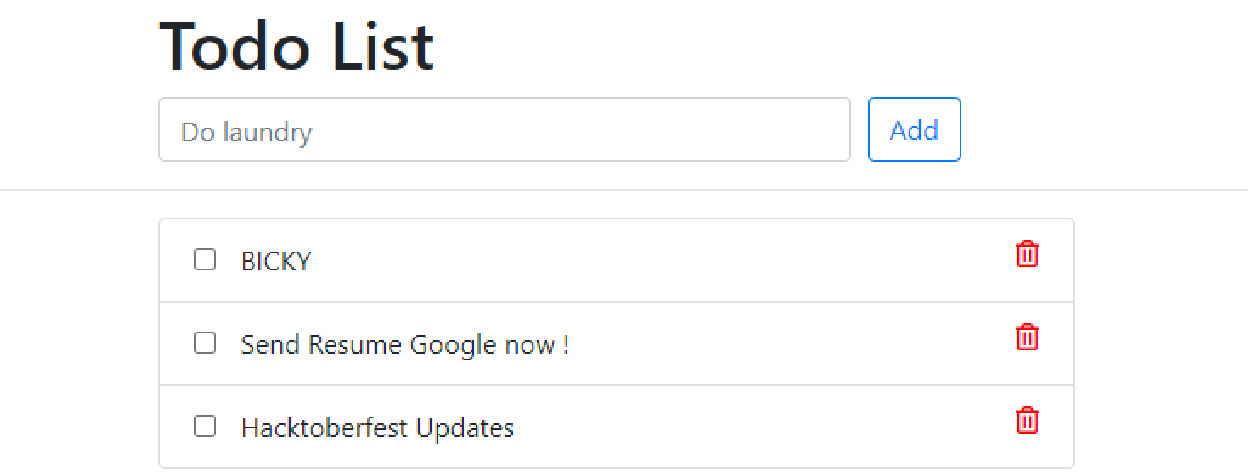


Todo List

Do laundry

☐ Send Resume Google now !

☐ Hacktoberfest Updates



Todo List

Do laundry

☐ BICKY

☐ Send Resume Google now !

☐ Hacktoberfest Updates

FINALLY WE GOT OUR WEB APP DEPLOY USING EC2 INSTANT