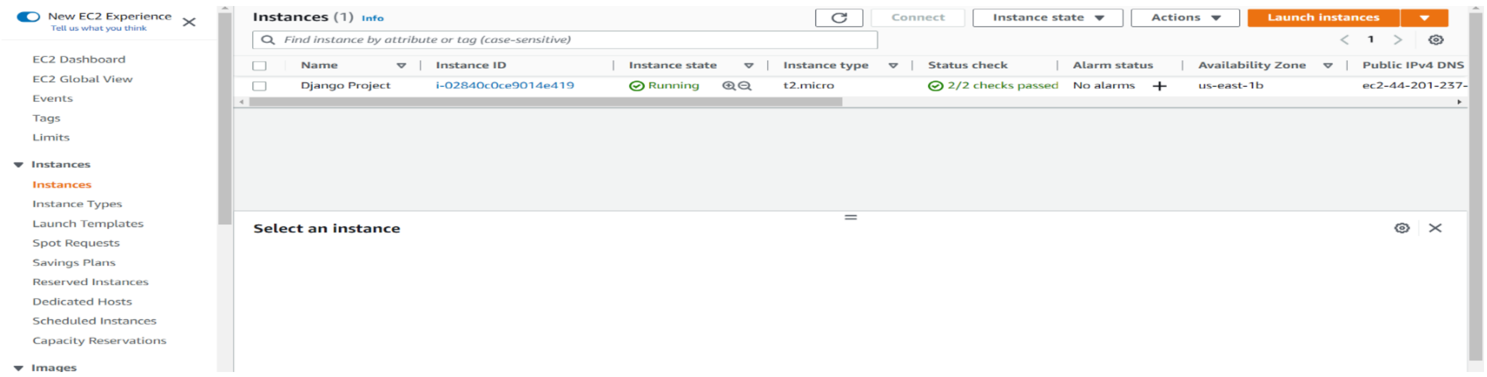
Django Web Application Using AWS & PostGreSql

1) Step 1:- Launch one Amazon EC2 Instance - **2AMI Linux - t2.micro** family with your key-pair & **SSH & All Traffic – Anywhere** Security Group.



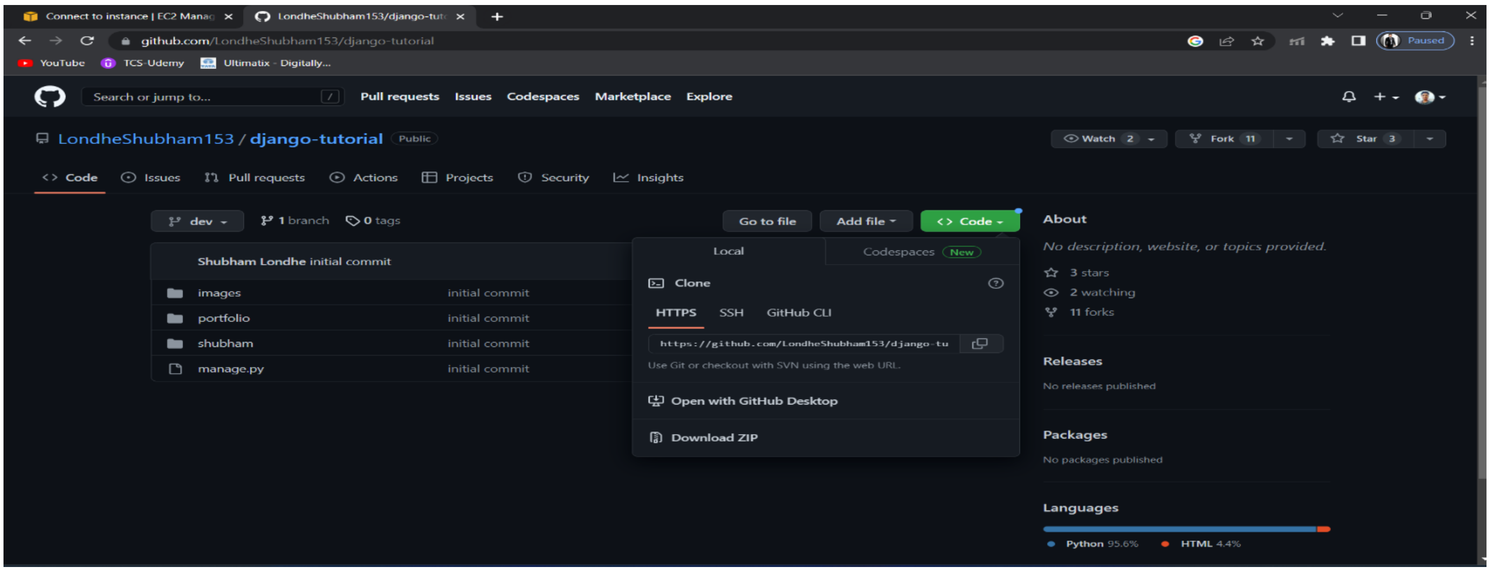
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Django Project	i-02840c0ce9014e419	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-44-201-237-

2) Step 2:- LogIn into EC2 Instance. And Run Following Commands.

```
$ sudo su - root
$ sudo yum update -y
```

3) Step 3:- Install Git. And Run Following Commands.

```
$ git
$ sudo yum install git -y
$ git
$ git clone  https://github.com/LondheShubham153/django-tutorial.git
$ ls
django-tutorial
$ cd django-tutorial/
$ ls
images  manage.py  portfolio  Shubham
```



Shubham Londhe initial commit

images	initial commit
portfolio	initial commit
shubham	initial commit
manage.py	initial commit

4) Step 4:- Install python3 if not present

```
$ sudo yum install python3 -y
```

5) Step 5:- In my case python3 is already present so directly install Django.And downloads dependencies.You can directly download all of these dependencies by making requirement.txt file.

```
$ sudo pip3 install django
$ sudo pip3 install psycopg2-binary
$ sudo pip3 install pillow
```

6) Step 6:- Install PostgreSql

Link :- https://dailyscrawl.com/how-to-install-postgresql-on-amazon-linux-2/

```
$ sudo amazon-linux-extras install postgresql10 vim epel -y
$ sudo yum install -y postgresql-server postgresql-devel
$ /usr/bin/postgresql-setup --initdb
$ sudo systemctl enable postgresql
$ sudo systemctl start postgresql
$ sudo systemctl status postgresql
```

fail

```
$ sudo service postgresql initdb
$ sudo systemctl start postgresql
$ sudo systemctl status postgresql
```

Active

```
$ clear
```

7) Step 7:- Now try to run your Django service.

```
$ sudo -u postgres psql
postgrs# \password
```

root

root

```
postgres# \q
```

Exit

8) Step 8:- Now try to run the project.

```
$ sudo vim shubham/settings.py
```

'USER': 'postgres',

(Save)

```
$ sudo -u postgres psql
postgres# create database portfoliodb;
postgres# \q
$ sudo vim /var/lib/pgsql/data/pg_hba.conf
```

Make all Method :- md5

```
$ sudo service postgresql restart
$ python3 manage.py runserver
$ python3 manage.py makemigrations
$ python3 manage.py migrate
$ psql -U postgres -d portfoliodb
```

Passwd:- root

```
portfoliodb# \dt
portfoliodb=# \q
$ python3 manage.py createsuperuser
```

Superuser:- akashbkochure

Email :-

Passwd:- akash123

Again:- akash123

```
$ python3 manage.py runserver
```

It runs on local host. And we have to run it on our EC2 instance. So follow steps.

```
$ python3 manage.py runserver 0:8000
```

0 means publicly accessible.

```
$ sudo vim shubham/settings.py
```

ALLOWED_HOSTS = ["*"]

```
$ sudo service postgresql restart
$ python3 manage.py runserver 0:8000
```

9) Step 9:- Put your dns public ip url into browser.

```
ec2-54-204-229-64.compute-1.amazonaws.com:8000
ec2-54-204-229-64.compute-1.amazonaws.com:8000/admin
```

10) Step 10:- logIn into Django Web App with your Superuser credential name & password.

Name:- akashbkochure

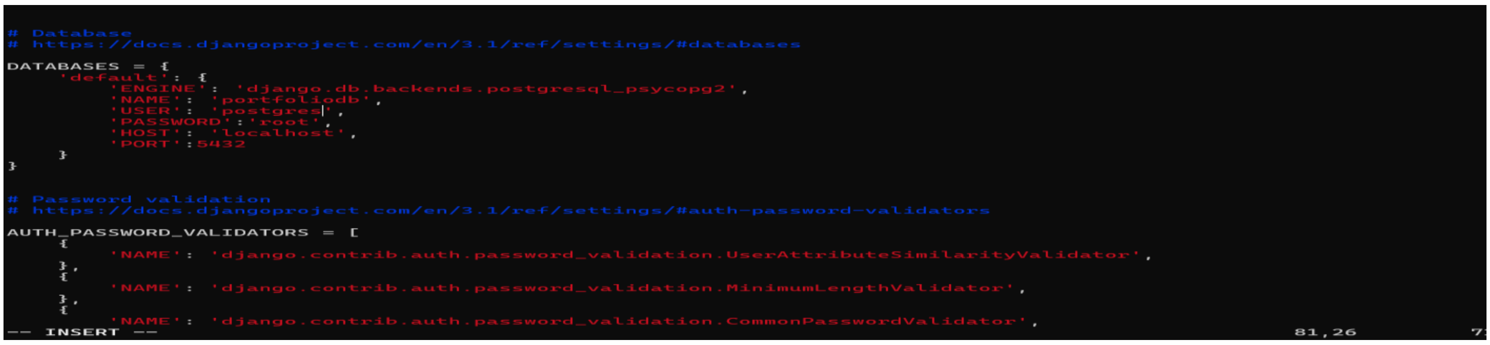
Pass:- akash123

11) Step 11:- Create job

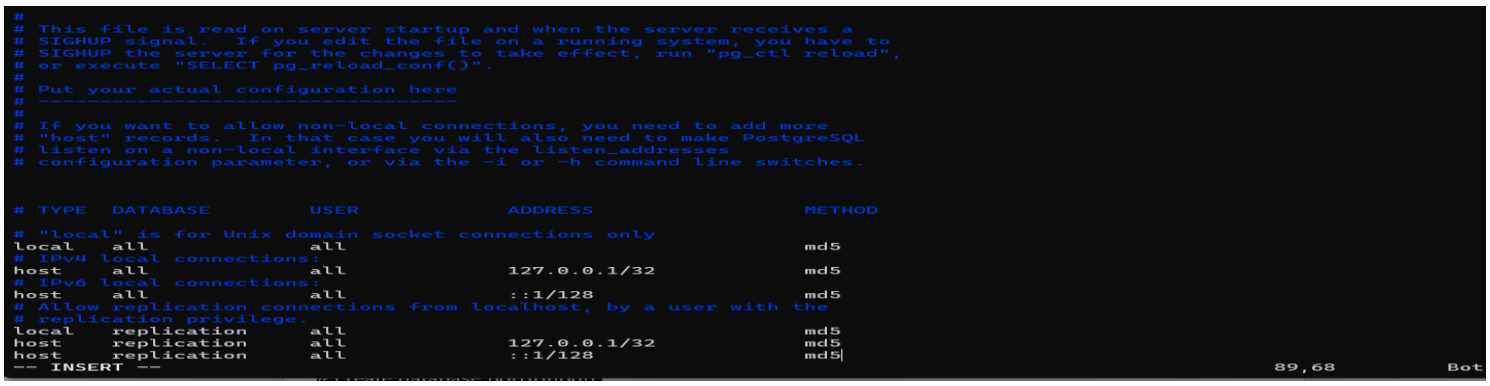
Any upload file

12) Step 12:- Put into Browser

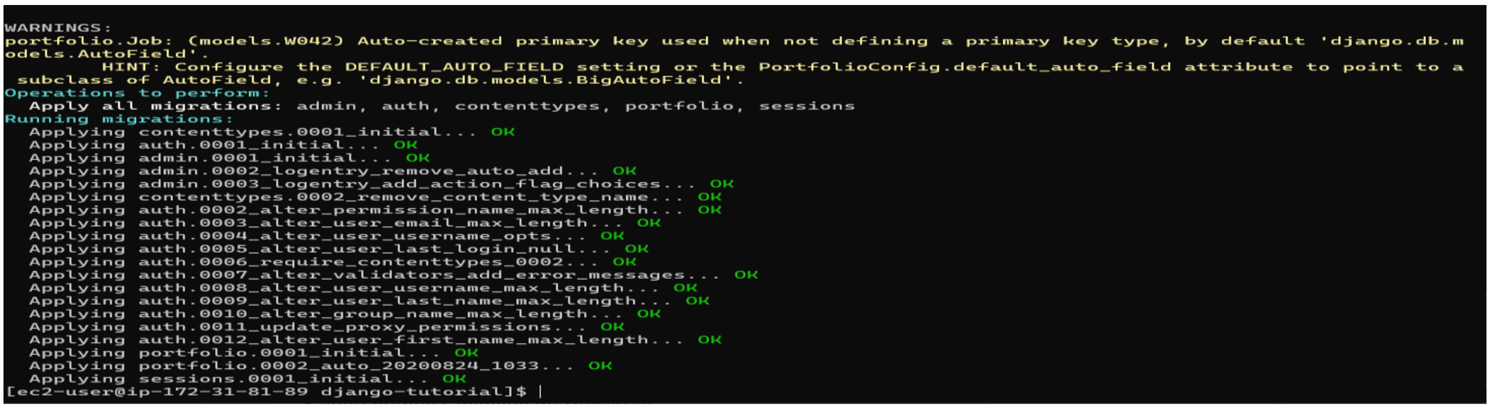
ec2-54-204-229-64.compute-1.amazonaws.com:8000/postgres/job



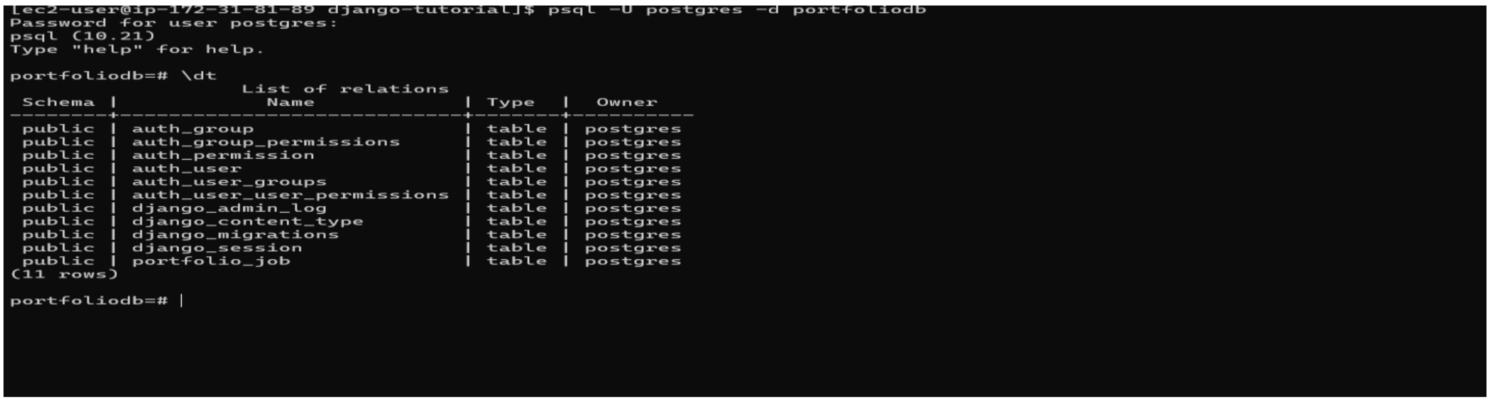
```
# Database
# https://docs.djangoproject.com/en/3.1/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'portfoliodb',
        'USER': 'postgres',
        'PASSWORD':'root',
        'HOST': 'localhost',
        'PORT':5432
    }
}

# Password validation
# https://docs.djangoproject.com/en/3.1/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
-- INSERT --                                                                    81,26          7
```



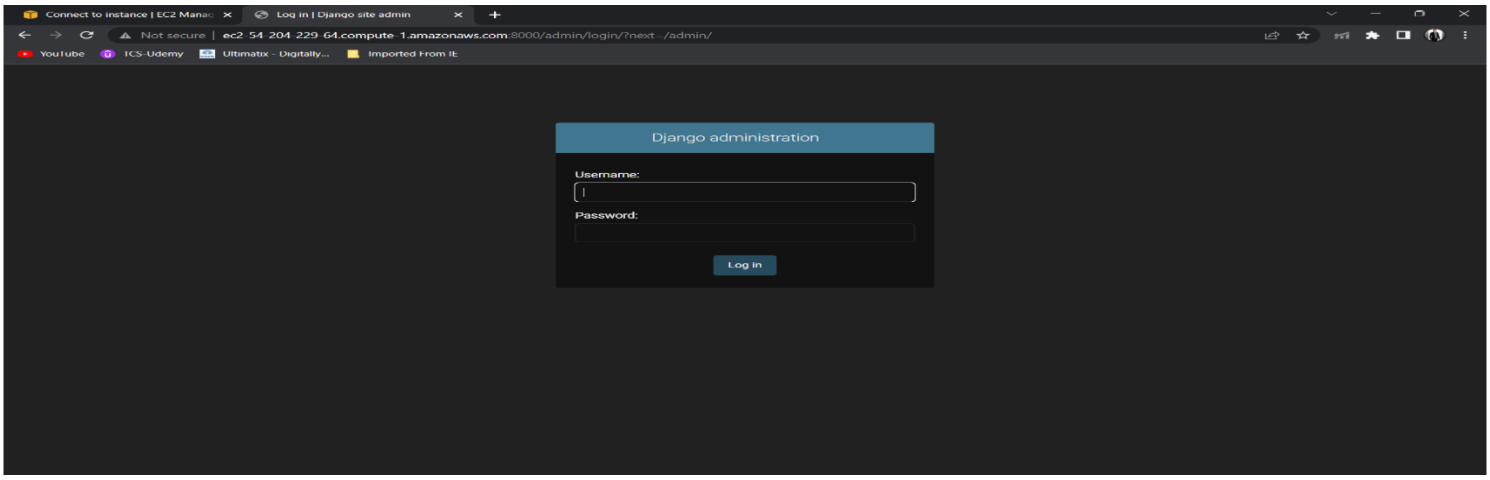
```
#
# This file is read on server startup and when the server receives a
# SIGHUP signal.  If you edit the file on a running system, you have to
# SIGHUP the server for the changes to take effect, run "pg_ctl reload",
# or execute "SELECT pg_reload_conf()".
#
# Put your actual configuration here
# ----------------------------------
#
# If you want to allow non-local connections, you need to add more
# "host" records.  In that case you will also need to make PostgreSQL
# listen on a non-local interface via the listen_addresses
# configuration parameter, or via the -i or -h command line switches.


# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     md5
# IPv4 local connections:
host    all             all             127.0.0.1/32            md5
# IPv6 local connections:
host    all             all             ::1/128                 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local   replication     all                                     md5
host    replication     all             127.0.0.1/32            md5
host    replication     all             ::1/128                 md5
-- INSERT --                                                          89,68         Bot
```



```
WARNINGS:
portfolio.Job: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.m
odels.AutoField'.
        HINT: Configure the DEFAULT_AUTO_FIELD setting or the PortfolioConfig.default_auto_field attribute to point to a
 subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, portfolio, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying portfolio.0001_initial... OK
  Applying portfolio.0002_auto_20200824_1033... OK
  Applying sessions.0001_initial... OK
[ec2-user@ip-172-31-81-89 django-tutorial]$ |
```



```
[ec2-user@ip-172-31-81-89 django-tutorial]$ psql -U postgres -d portfoliodb
Password for user postgres:
psql (10.21)
Type "help" for help.

portfoliodb=# \dt
                 List of relations
 Schema |              Name              | Type  |  Owner
--------+--------------------------------+-------+----------
 public | auth_group                     | table | postgres
 public | auth_group_permissions         | table | postgres
 public | auth_permission                | table | postgres
 public | auth_user                      | table | postgres
 public | auth_user_groups               | table | postgres
 public | auth_user_user_permissions     | table | postgres
 public | django_admin_log               | table | postgres
 public | django_content_type            | table | postgres
 public | django_migrations              | table | postgres
 public | django_session                 | table | postgres
 public | portfolio_job                  | table | postgres
(11 rows)

portfoliodb=# |
```


Connect to instance | EC2 Manag × | Log in | Django site admin ×

Not secure | ec2-54-204-229-64.compute-1.amazonaws.com:8000/admin/login/?next=/admin/

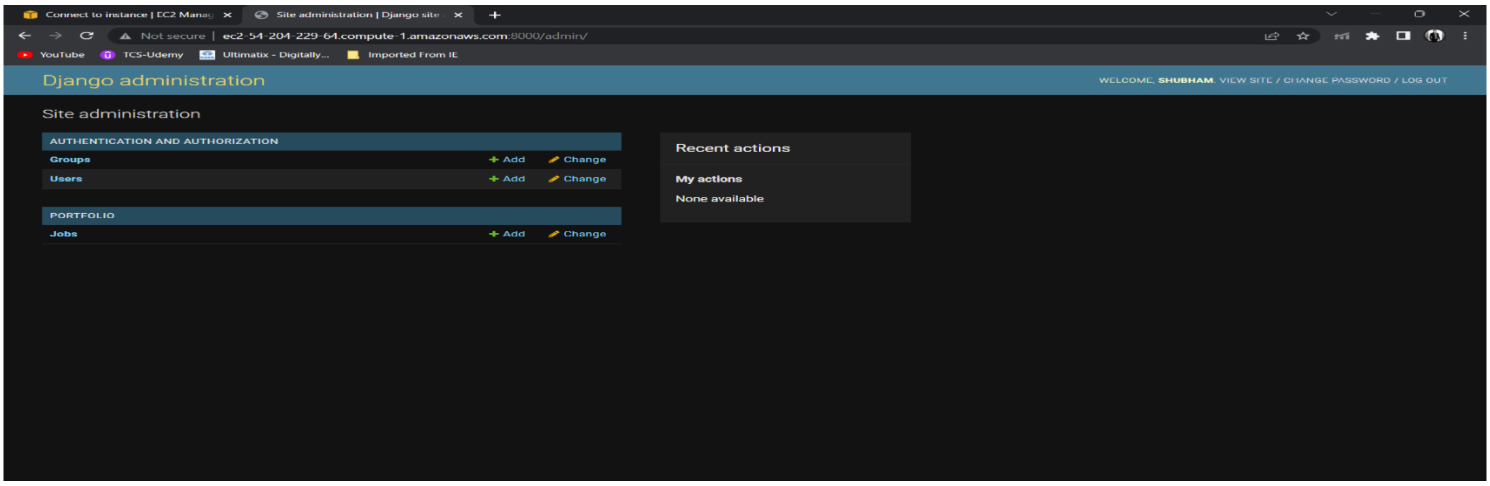
YouTube | TCS-Udemy | Ultimatix - Digitally... | Imported From IE

Django administration

Username:

Password:

Log in


Connect to instance | EC2 Manag × | Site administration | Django site ×

Not secure | ec2-54-204-229-64.compute-1.amazonaws.com:8000/admin/

Django administration

WELCOME, **SHUBHAM**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Site administration

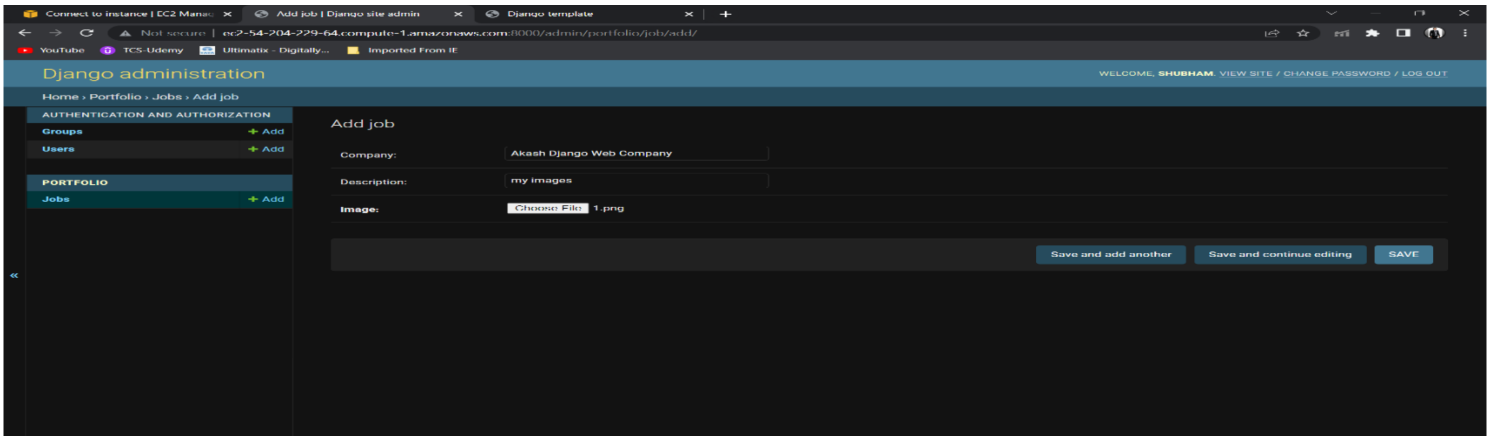
AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

PORTFOLIO		
Jobs	+ Add	Change

Recent actions

My actions

None available


Connect to instance | EC2 Manag × | Add job | Django site admin × | Django template ×

Not secure | ec2-54-204-229-64.compute-1.amazonaws.com:8000/admin/portfolio/job/add/

Django administration

WELCOME, **SHUBHAM**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Portfolio › Jobs › Add job

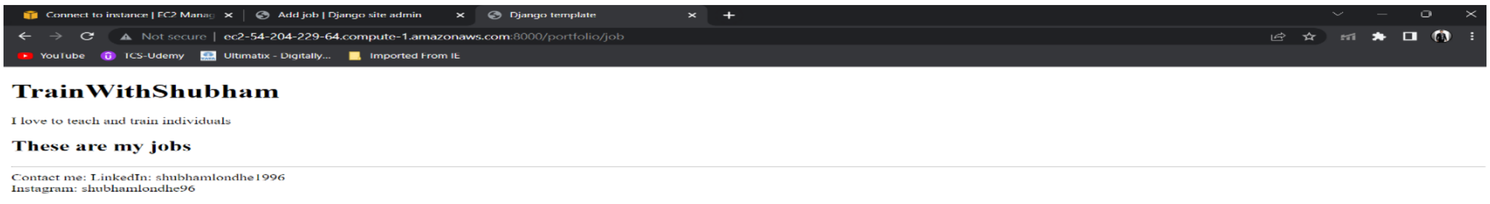
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add

PORTFOLIO
Jobs + Add

Add job

Company:	Akash Django Web Company
Description:	my images
Image:	Choose File 1.png

Save and add another | Save and continue editing | SAVE


Connect to instance | EC2 Manag × | Add job | Django site admin × | Django template ×

Not secure | ec2-54-204-229-64.compute-1.amazonaws.com:8000/portfolio/job

TrainWithShubham

I love to teach and train individuals

These are my jobs

Contact me: LinkedIn: shubhamlondhe1996
Instagram: shubhamlondhe96

Successfully Completed.