CS 623 - CLOUD COMPUTING **FINAL PROJECT REPORT**

USING CLOUD COMPUTING TO IMPLEMENT BIDDING WEBSITE APPLICATION

BY –

Shashiraj Waleshwar – ve8664

Rishabh Parekh – hr8687

Siddhant Shah – pc3460

Table of contents

CHAPTER 1: INTRODUCTION

1.1 Problem Summary

An organization or company wants an open platform for the bidding projects for their own organization. So, an open bidding platform is created where the organizations can come together and put a bidding requests for their projects. The reason why we created this is that these companies don't want to put upfront investment on resources like hardware, servers as they don't know about the company size in future.

The company does not want to rely on primitive technology such as manual storage or local server storage which creates many problems while the company grows and it's also very slow while retrieving the employee information from the database and they don't map according to all their databases. So, the company requirements are to provide a system where large data can be stored based on size and storing employee information and helping in retrieving the data with the help of the employee ID which we can use as the primary key where this overall process is done in a fast and efficient manner.

1.2 Solution

To solve the above mentioned problem, our proposed system will eliminate all the above issues and provide efficient results by using cloud computing technology, this system is implemented as a web application on AWS.

The web application we developed is deployed on AWS cloud platform where EC2 is used for launching instances S3 bucket is used for storing for example images and Relational Database Services for connecting to databases, the whole idea is to deploy web application on an EC2 instance which acts as Infrastructure as a Service (IaaS). In this web application we are providing search functionality to retrieve stored information about employees very quickly and efficiently.

So, by this system the company or organization will be more benefited in numerous ways like companies don’t need to put upfront investment on purchasing resources as scaling up and scaling down is done whenever it’s needed automatically and pay as perusage of the resources only.

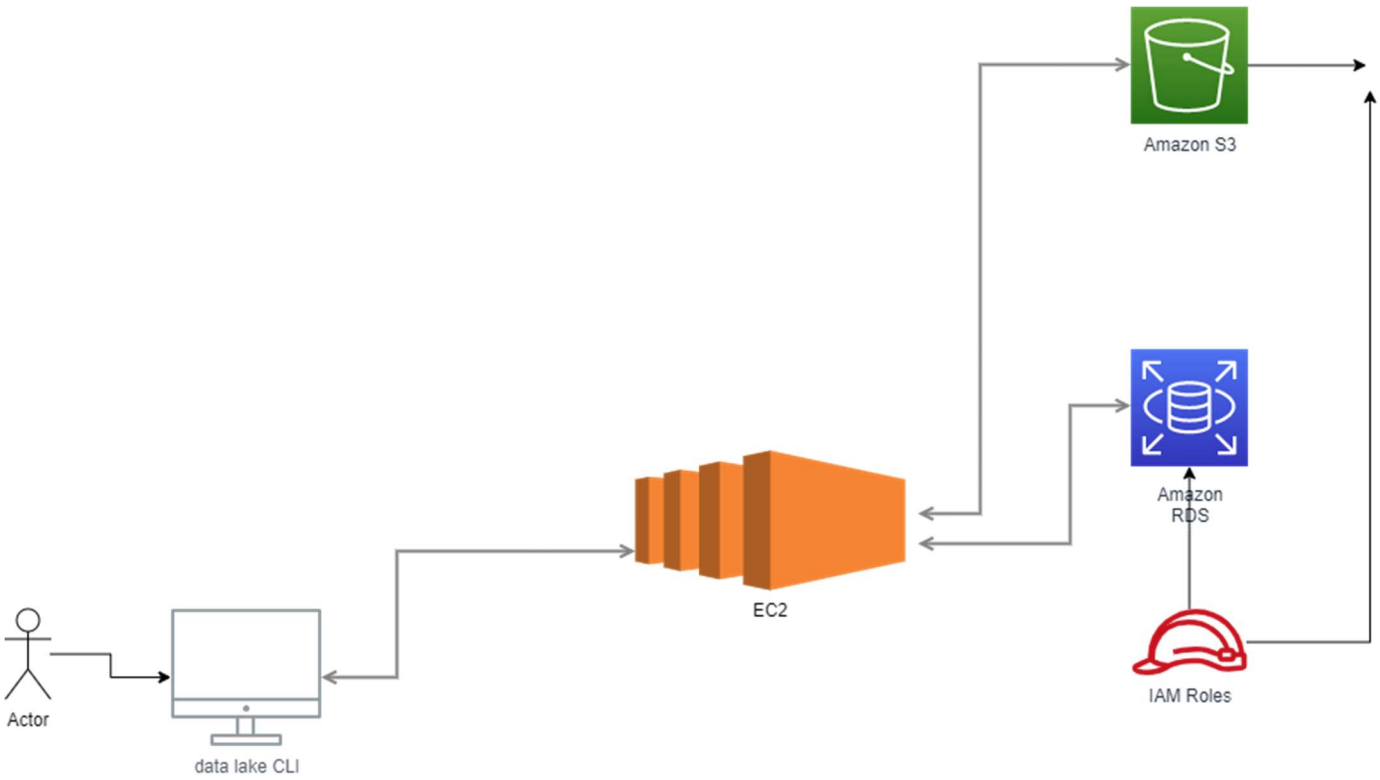
We are using many AWS services to implement this system like RDS, S3, EC2 and and python for programming. The cloud platform offers more storage space, so large data can be stored, the computation speed is very fast so that administrators can retrieve or analyze employee data very efficiently.

CHAPTER 2: IMPLEMENTATION WORKFLOW

2.1 Architectural Overview

Here we are going to discuss the Architecture of the system and Relation between services.

2.1.1 Architecture of the system



The overall architecture of the system can be roughly divided into three components:

1. Selecting and configuring Amazon services
2. Building client-side application
3. Connecting to S3 and RDS databases and IAM for giving permissions to our databases.

In selecting and configuring Amazon services we are using three AWS namely -

1. RDS
2. S3
3. EC2
4. IAM

We need to select these services one by one From AWS dashboard and click appropriate options to set it up.

In building client-side applications, we are using Python and Html for the coding part.

We then deployed our application on EC2 instance which is getting connected to RDS and S3 bucket.

2.1.2 Relation between services

- We are using Amazon **RDS** service, which is used to operate and scale a relational database in the cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks. In this we are using MySQL database and this is used to store the entries. This RDS service will be connected by the EC2 server, this relation is used to access the database.

- We are using Amazon **S3** service, which is a Storage for the internet. We can use it to store and retrieve any amount of data at any time, from anywhere on the web. We should give EC2 instance access to S3, basically to upload the data. To make this relation we will give an IAM role to our instance.

- We are using Amazon **EC2** service, it is a web service for launching and managing Linux/UNIX and Windows Server instances in Amazon's data centers. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment. Here EC2 should be connected to RDS, S3 as described above, and the website is alsodeployed in EC2 server.

- We are using an **IAM** service, which is used when we connect with our RDS. We should give authentication to our S3 service, so we need to provide an IAM role for our instance.

CHAPTER 3: DEVELOPMENT PROCESS

Here we are going to discuss the Description of the development process and challenges the team faced during this project building.

3.1 Description of the development process

1) The first phase is to select and configure the Amazon services.

We are using three Amazon service initially -

1. RDS
2. S3

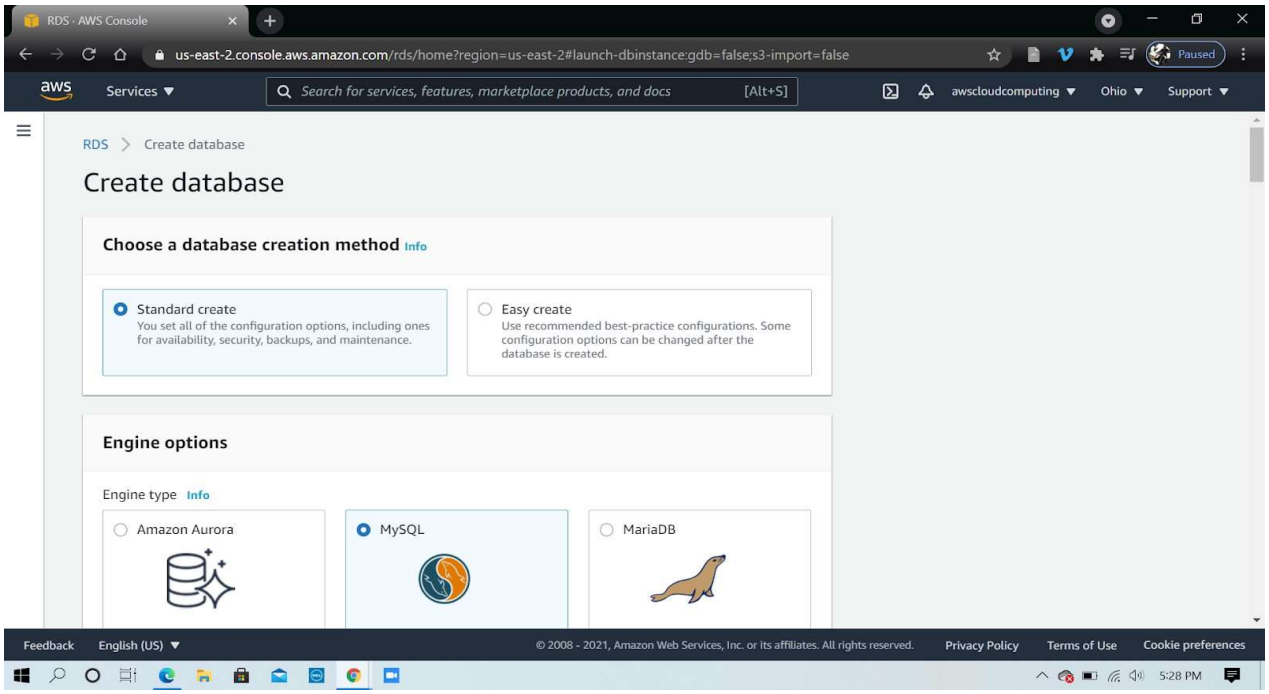
3. EC2

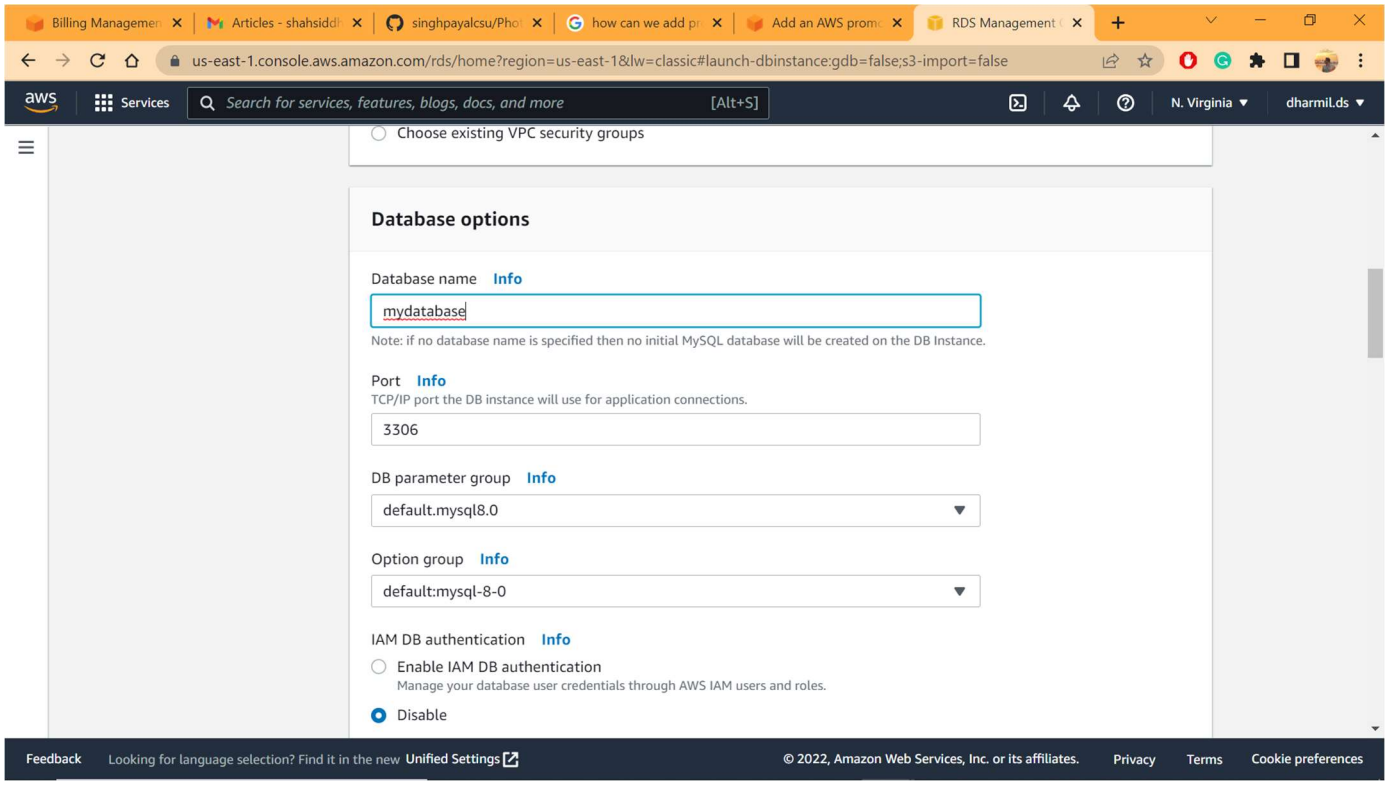
- Deploy MySQL database on AWS

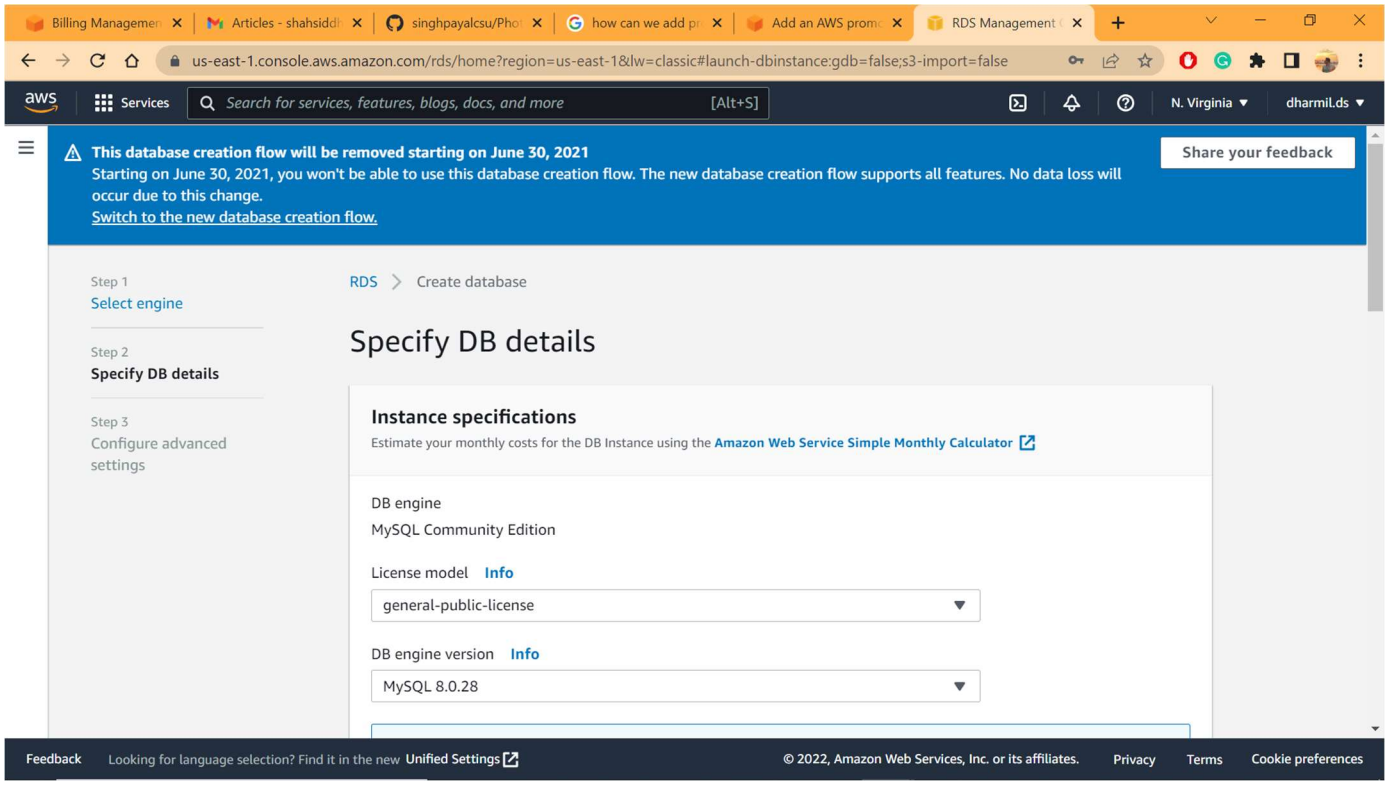
It is done by using Amazon RDS, for this we should select RDS service from AWS management console dashboard and then select MySQL database engine. There are many engines for this project. We are using MySQL engine.

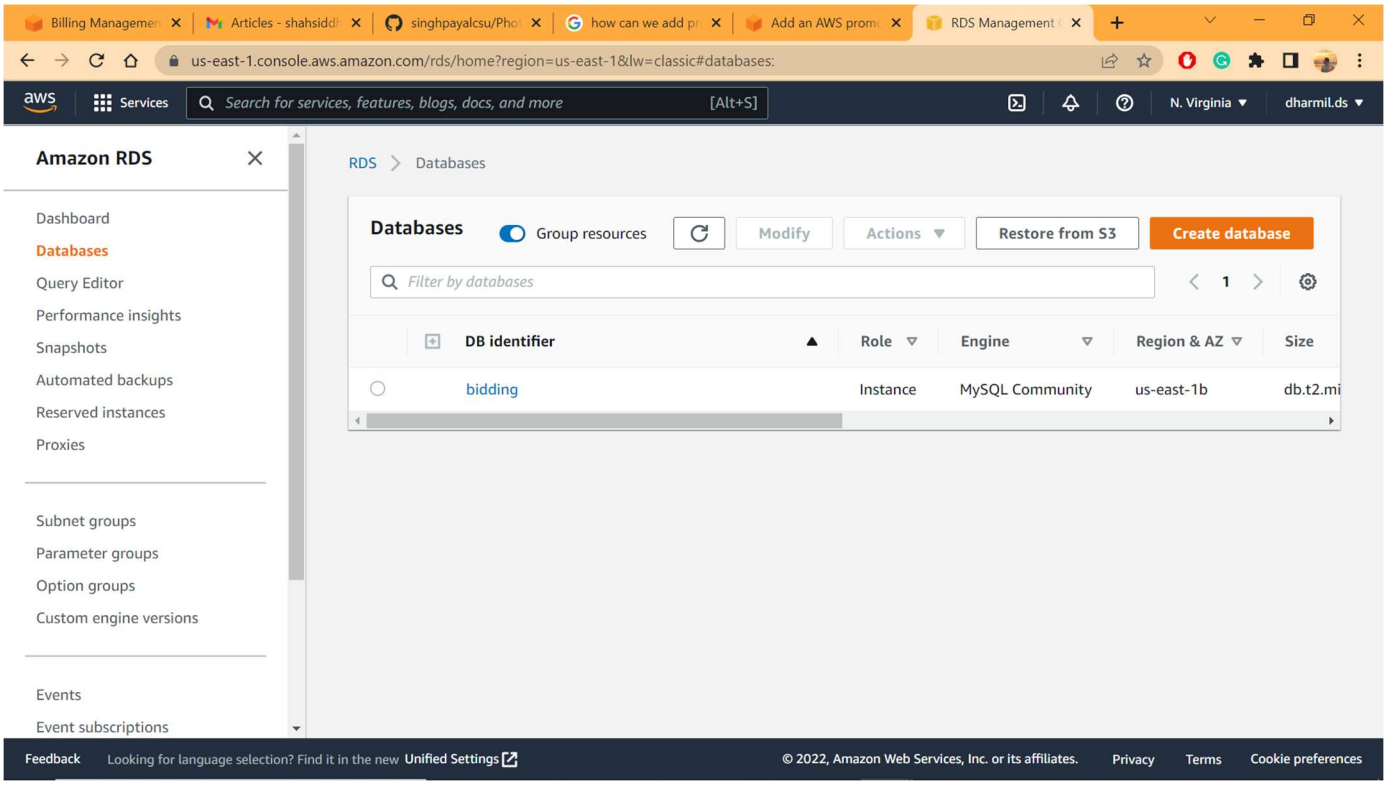
There are many advantages of using RDS, we don't need to launch a server, install MySQL on it and configure it manually instead of that we are just using AWS dashboard and select right options which will be managed automatically by AWS.

Some key advantages are easy to administer, highly scalable, available & durable, fast, secure and inexpensive.



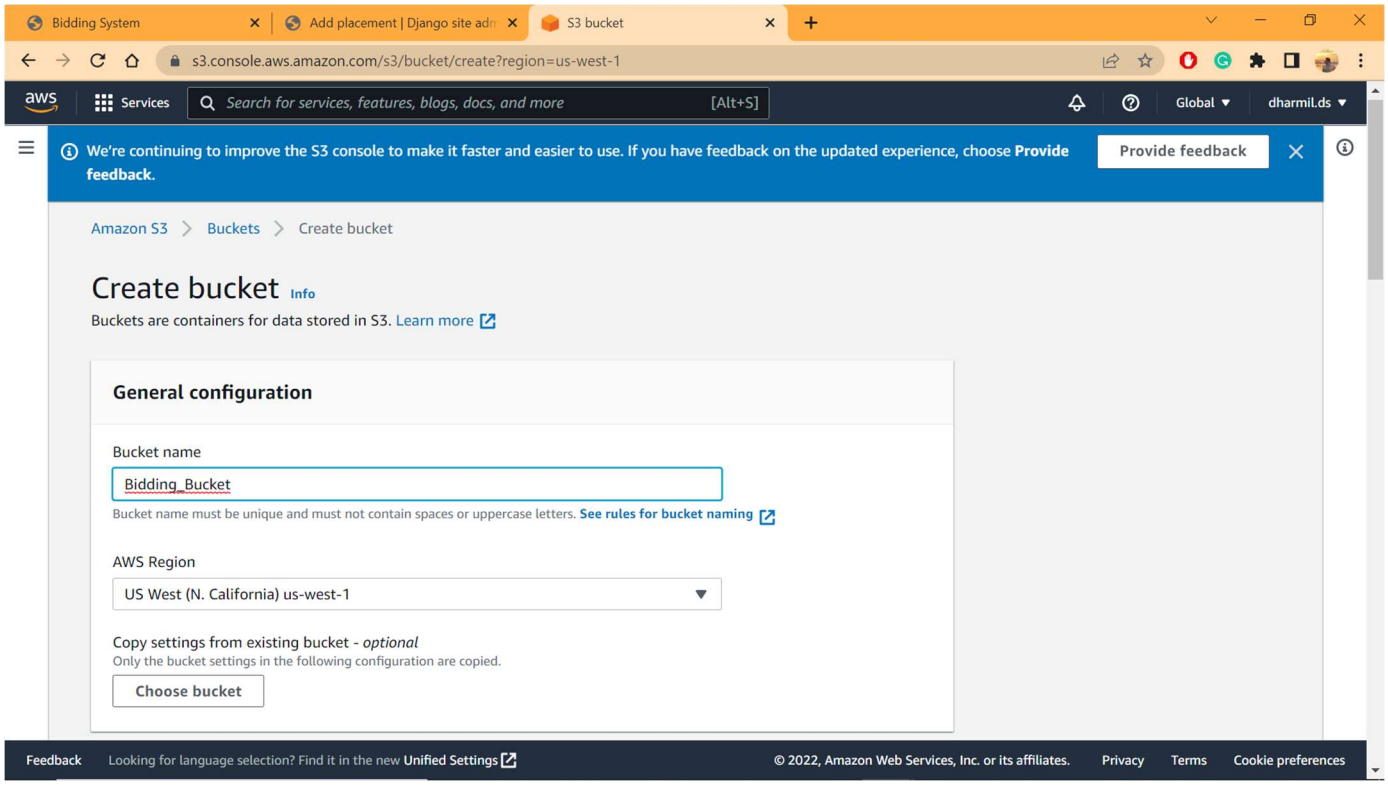

Billing Managemen
Articles - shahsiddi
singhpayalcsu/Phot
how can we add pr
Add an AWS promo
RDS Management
us-east-1.console.aws.amazon.com/rds/home?region=us-east-1&lw=classic#launch-dbinstance:gdb=false;s3-import=false
aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
N. Virginia
dharmil.ds
Choose existing VPC security groups
Database options
Database name Info
mydatabase
Note: if no database name is specified then no initial MySQL database will be created on the DB Instance.
Port Info
TCP/IP port the DB instance will use for application connections.
3306
DB parameter group Info
default.mysql8.0
Option group Info
default:mysql-8-0
IAM DB authentication Info
Enable IAM DB authentication
Manage your database user credentials through AWS IAM users and roles.
Disable
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

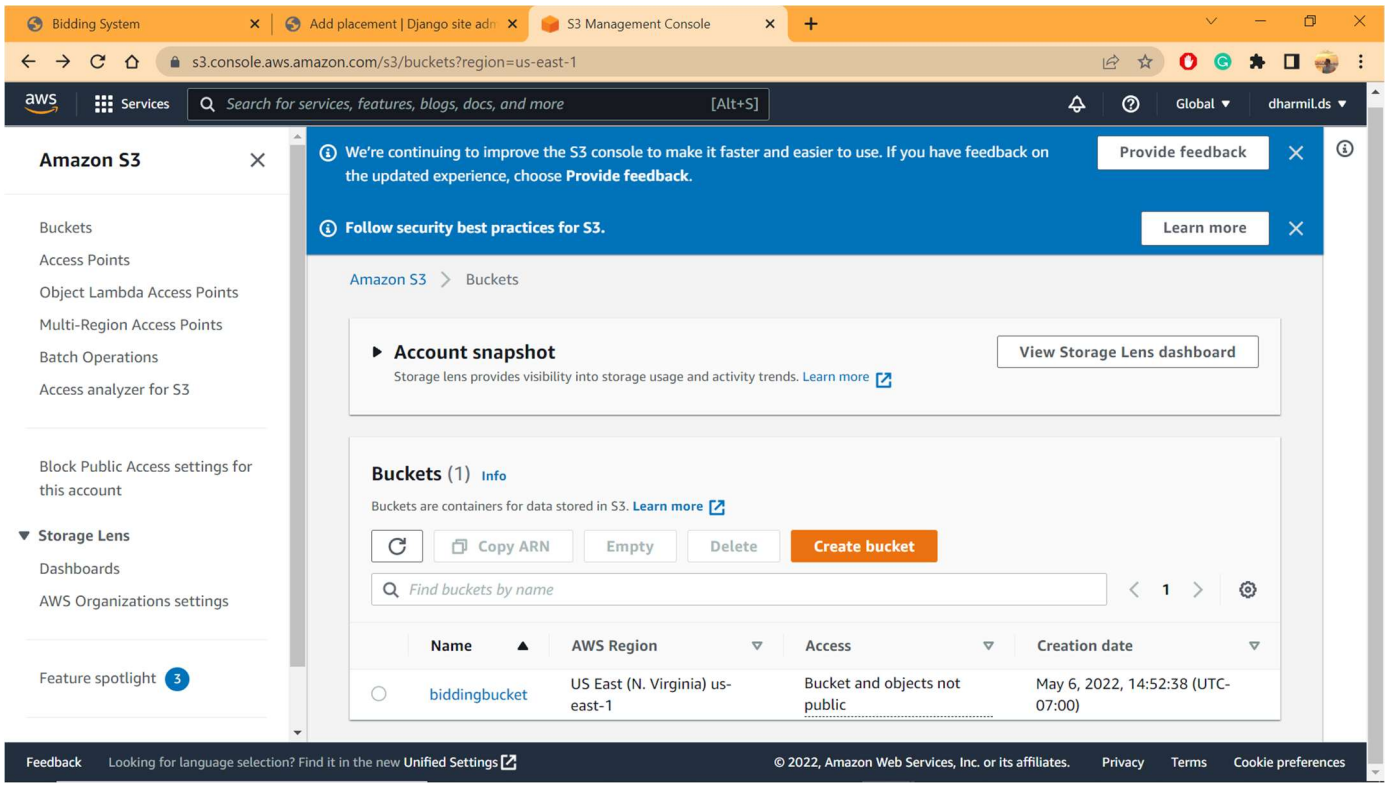

Billing Managemen
Articles - shahsiddi
singhpayalcsu/Phot
how can we add pr
Add an AWS promo
RDS Management
us-east-1.console.aws.amazon.com/rds/home?region=us-east-1&lw=classic#launch-dbinstance:gdb=false;s3-import=false
aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
N. Virginia
dharmil.ds
This database creation flow will be removed starting on June 30, 2021
Starting on June 30, 2021, you won't be able to use this database creation flow. The new database creation flow supports all features. No data loss will occur due to this change.
Switch to the new database creation flow.
Share your feedback
Step 1
Select engine
Step 2
Specify DB details
Step 3
Configure advanced settings
RDS > Create database
Specify DB details
Instance specifications
Estimate your monthly costs for the DB Instance using the Amazon Web Service Simple Monthly Calculator
DB engine
MySQL Community Edition
License model Info
general-public-license
DB engine version Info
MySQL 8.0.28
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

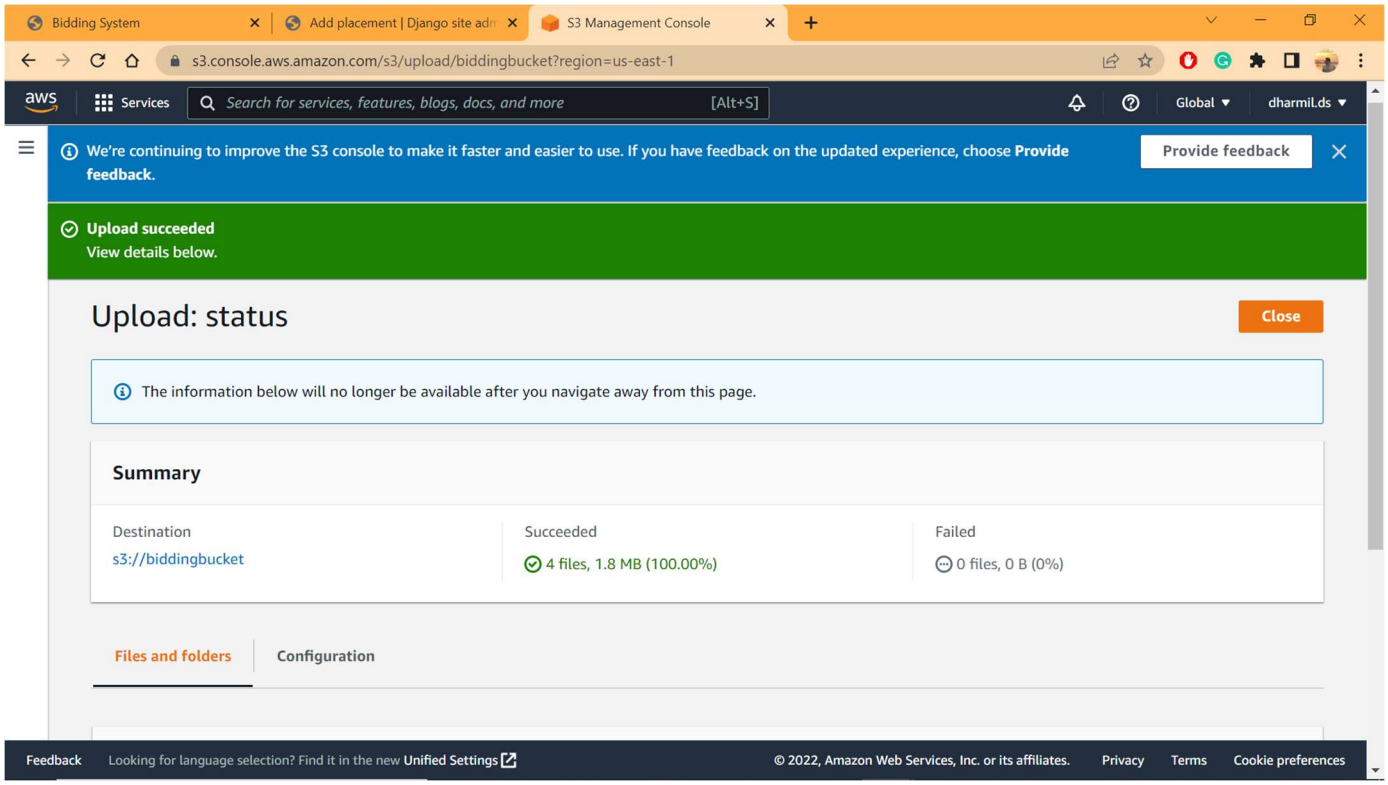


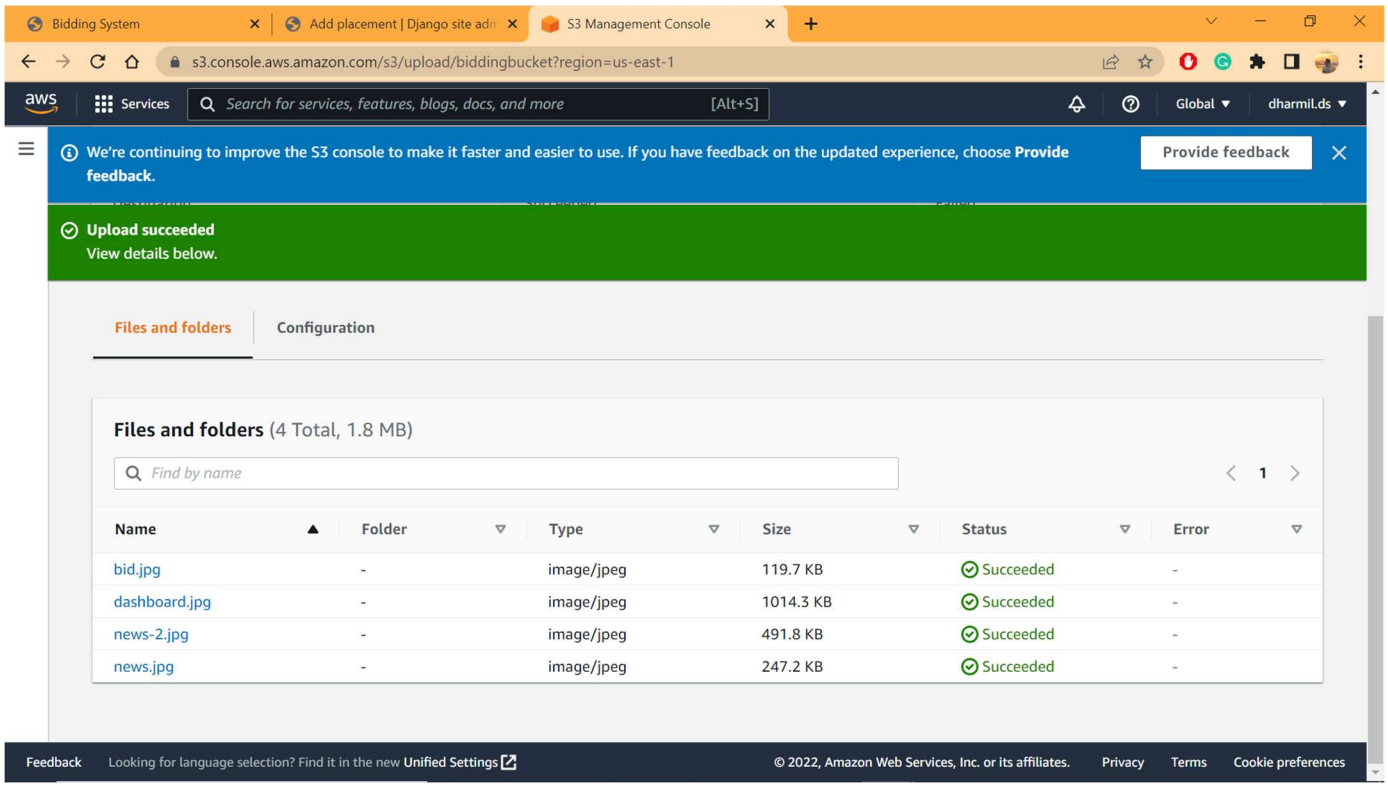
- Create S3 bucket

It is done by using Amazon S3, for this we should select S3 service from AWS management console dashboard and create it. Check whether your bucket has been deployed in the correct region. Below screenshot shows the configuration that we have done for the S3 bucket creation as we are fetching the images from the S3 bucket.

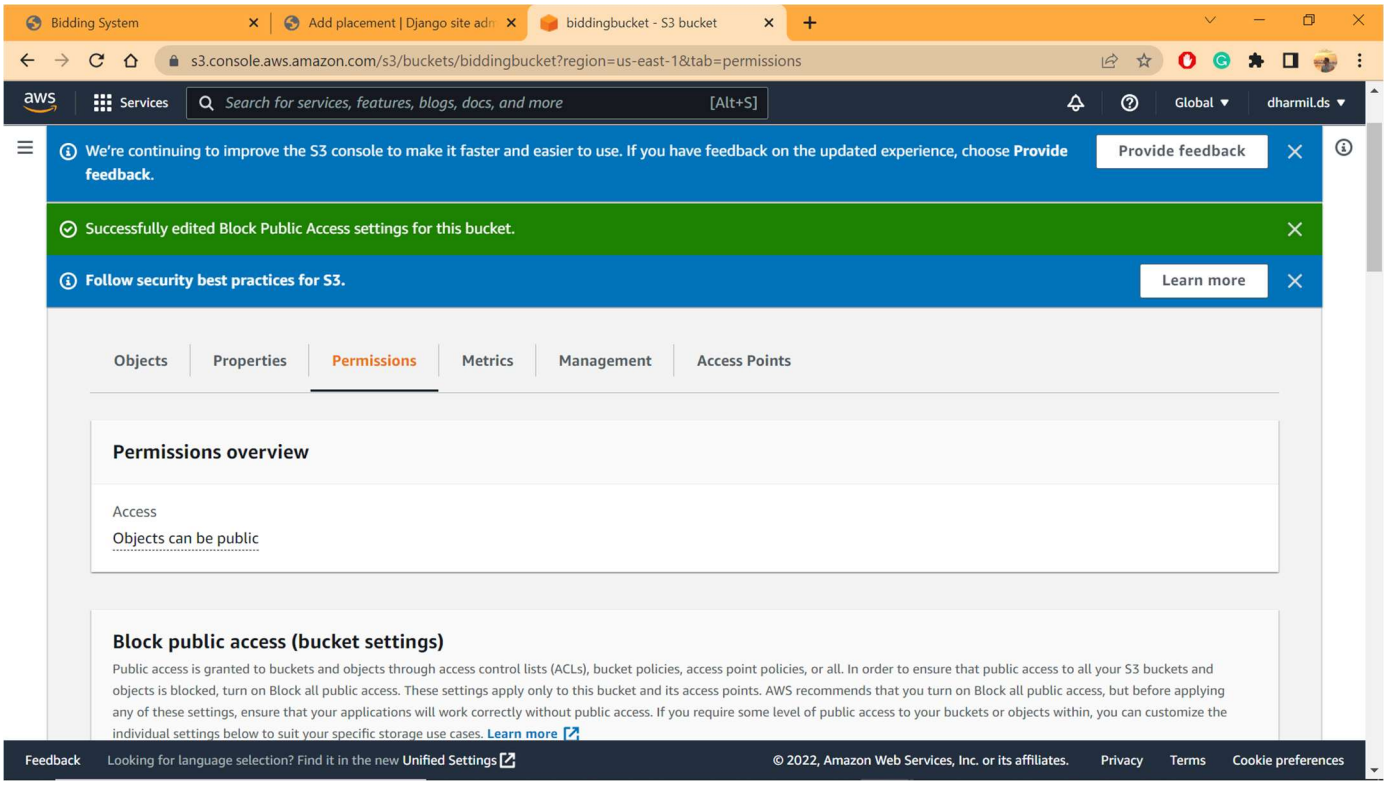

Bidding System
Add placement | Django site adm
S3 bucket
s3.console.aws.amazon.com/s3/bucket/create?region=us-west-1
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Global
dharmil.ds
We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose Provide feedback.
Provide feedback
Amazon S3 > Buckets > Create bucket
Create bucket Info
Buckets are containers for data stored in S3. Learn more
General configuration
Bucket name
Bidding_Bucket
Bucket name must be unique and must not contain spaces or uppercase letters. See rules for bucket naming
AWS Region
US West (N. California) us-west-1
Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.
Choose bucket
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

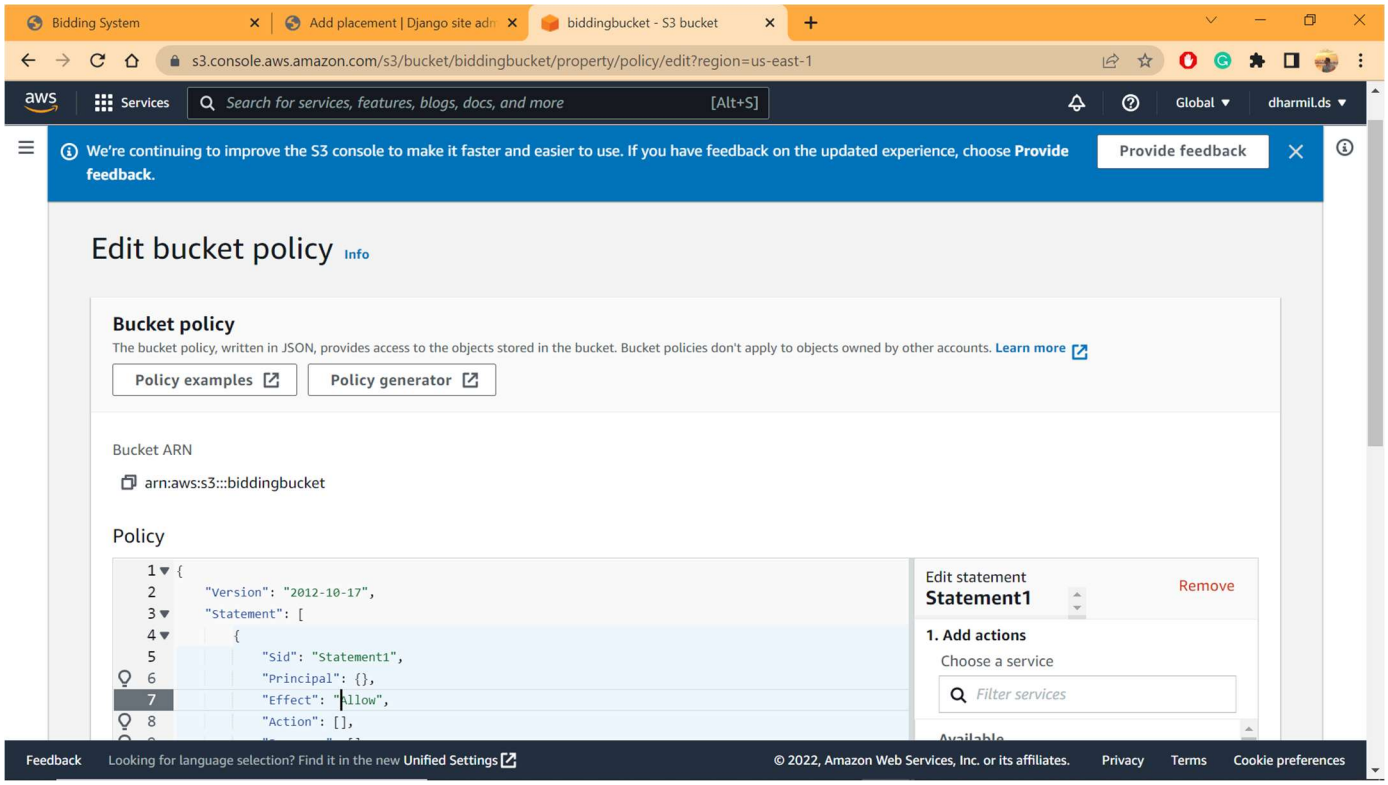

Bidding System
Add placement | Django site adm
S3 Management Console
s3.console.aws.amazon.com/s3/buckets?region=us-east-1
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Global
dharmil.ds
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
Access analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight 3
We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose Provide feedback.
Provide feedback
Follow security best practices for S3.
Learn more
Amazon S3 > Buckets
Account snapshot
Storage lens provides visibility into storage usage and activity trends. Learn more
View Storage Lens dashboard
Buckets (1) Info
Buckets are containers for data stored in S3. Learn more
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
1
Name
AWS Region
Access
Creation date
biddingbucket
US East (N. Virginia) us-east-1
Bucket and objects not public
May 6, 2022, 14:52:38 (UTC-07:00)
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

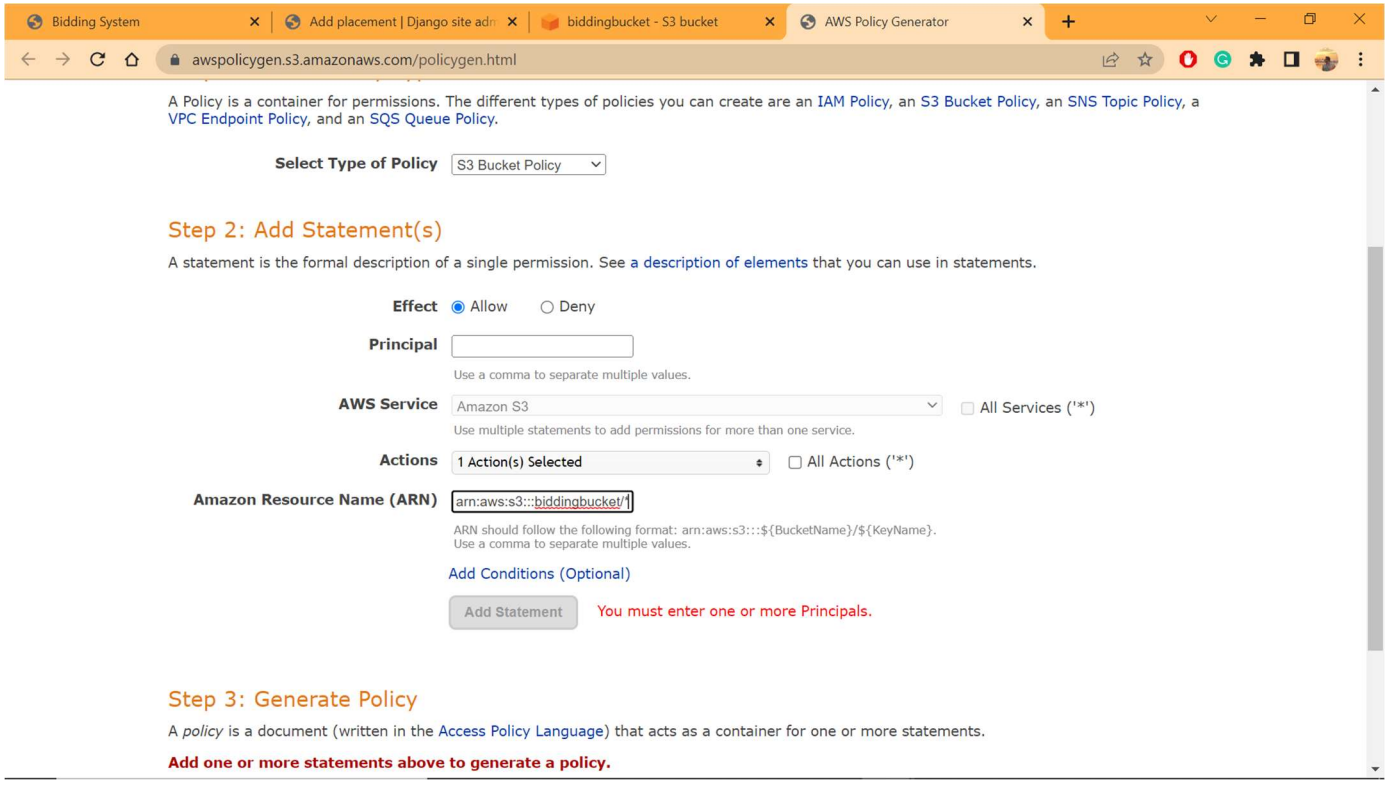

Bidding System
Add placement | Django site adm
S3 Management Console
s3.console.aws.amazon.com/s3/upload/biddingbucket?region=us-east-1
aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Global
dharmil.ds
We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose Provide feedback.
Provide feedback
Upload succeeded
View details below.
Upload: status
Close
The information below will no longer be available after you navigate away from this page.
Summary
Destination
s3://biddingbucket
Succeeded
4 files, 1.8 MB (100.00%)
Failed
0 files, 0 B (0%)
Files and folders
Configuration
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

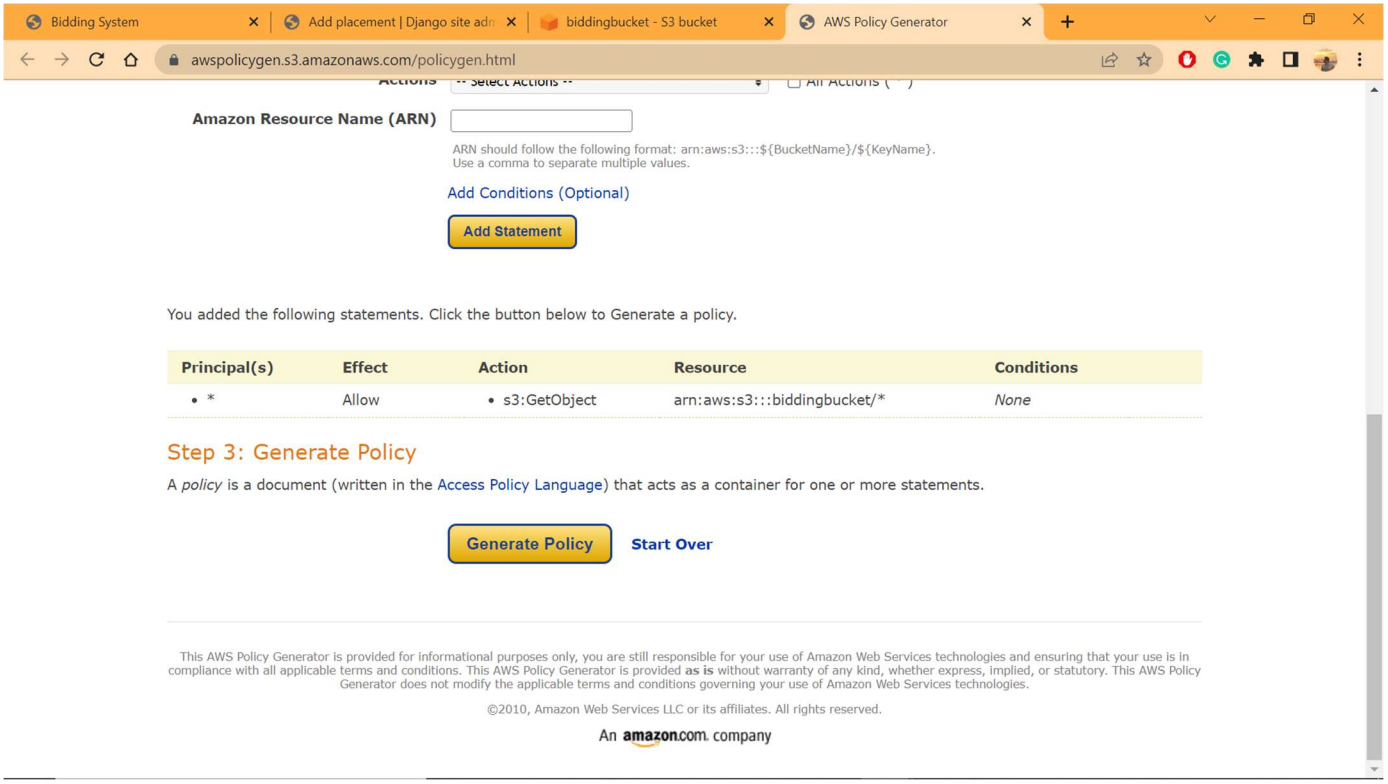

Bidding System
Add placement | Django site adm
S3 Management Console
s3.console.aws.amazon.com/s3/upload/biddingbucket?region=us-east-1
aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Global
dharmil.ds
We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose Provide feedback.
Provide feedback
Upload succeeded
View details below.
Files and folders
Configuration
Files and folders (4 Total, 1.8 MB)
Find by name
1
Name
Folder
Type
Size
Status
Error
bid.jpg
-
image/jpeg
119.7 KB
Succeeded
-
dashboard.jpg
-
image/jpeg
1014.3 KB
Succeeded
-
news-2.jpg
-
image/jpeg
491.8 KB
Succeeded
-
news.jpg
-
image/jpeg
247.2 KB
Succeeded
-
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

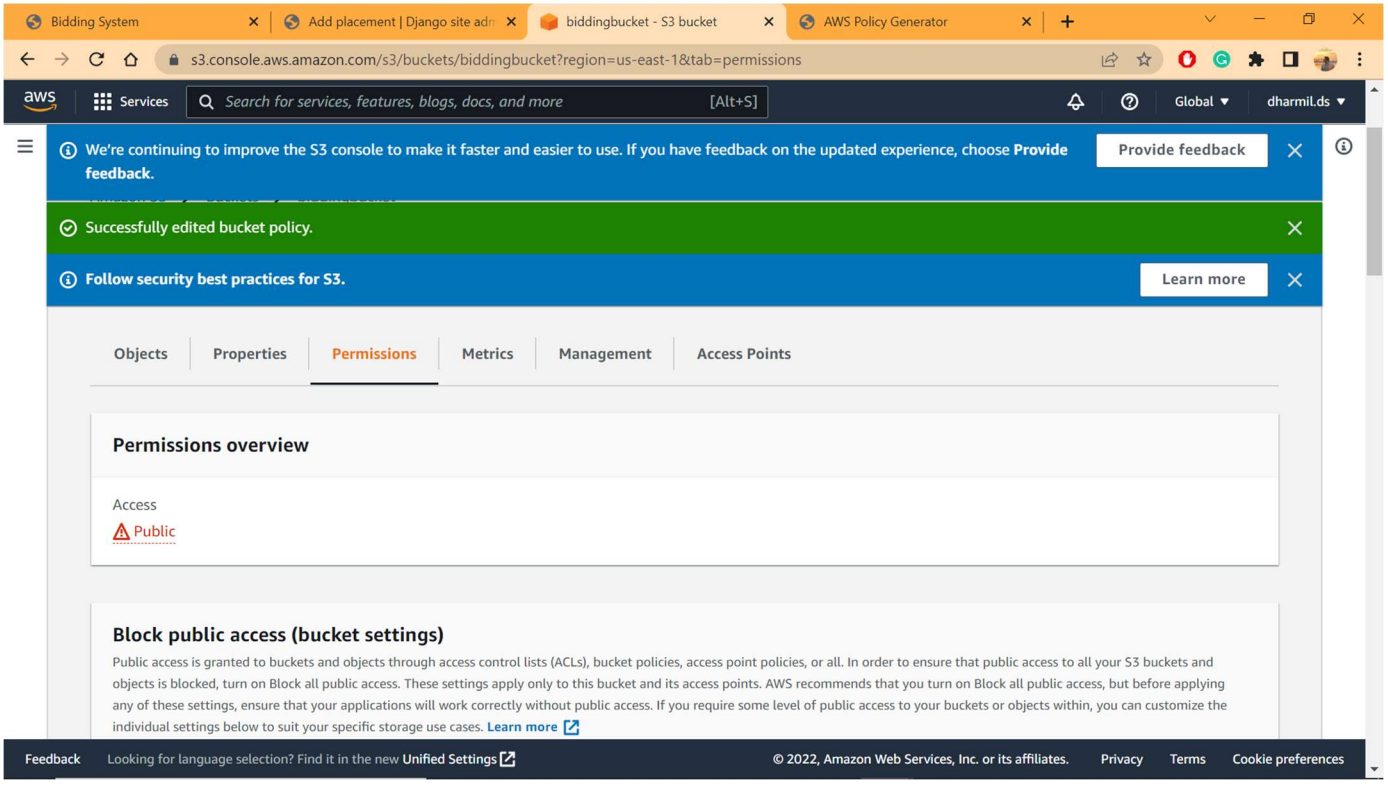
Some key advantages are Cost effective storage classes, easily managed data and access controls, query-in-place and process on-request, performance, scalability, availability and durability

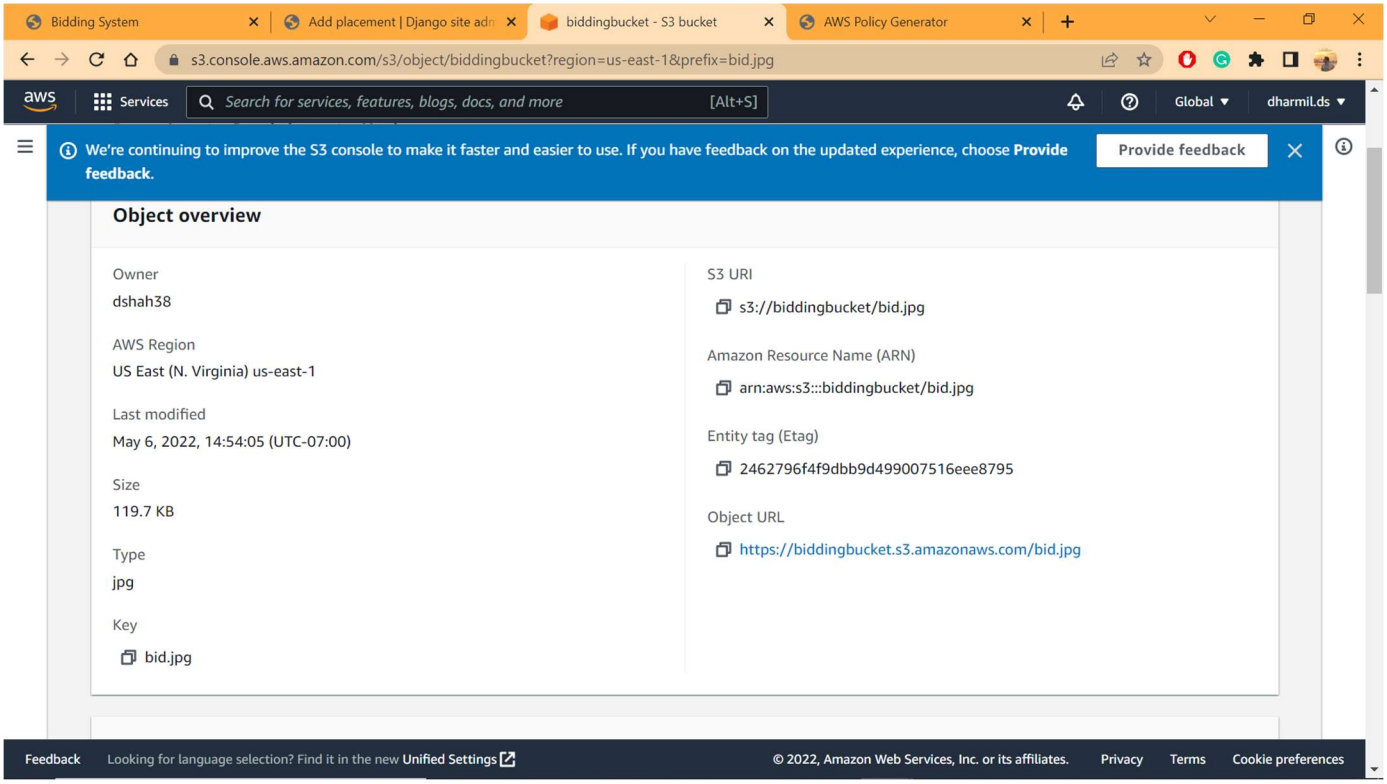



Bidding System
Add placement | Django site adm
biddingbucket - S3 bucket
s3.console.aws.amazon.com/s3/bucket/biddingbucket/property/policy/edit?region=us-east-1
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Global
dharmil.ds
We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose Provide feedback.
Provide feedback
Edit bucket policy
Info
Bucket policy
The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts.
Learn more
Policy examples
Policy generator
Bucket ARN
arn:aws:s3:::biddingbucket
Policy
{
"Version": "2012-10-17",
"Statement": [
{
"Sid": "Statement1",
"Principal": {},
"Effect": "Allow",
"Action": [],
Edit statement
Statement1
Remove
1. Add actions
Choose a service
Filter services
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Bidding System
Add placement | Django site adm
biddingbucket - S3 bucket
AWS Policy Generator
awspolicygen.s3.amazonaws.com/policygen.html
A Policy is a container for permissions. The different types of policies you can create are an IAM Policy, an S3 Bucket Policy, an SNS Topic Policy, a VPC Endpoint Policy, and an SQS Queue Policy.
Select Type of Policy
S3 Bucket Policy
Step 2: Add Statement(s)
A statement is the formal description of a single permission. See a description of elements that you can use in statements.
Effect
Allow
Deny
Principal
Use a comma to separate multiple values.
AWS Service
Amazon S3
All Services ('*')
Use multiple statements to add permissions for more than one service.
Actions
1 Action(s) Selected
All Actions ('*')
Amazon Resource Name (ARN)
arn:aws:s3:::biddingbucket/1
ARN should follow the following format: arn:aws:s3:::${BucketName}/${KeyName}.
Use a comma to separate multiple values.
Add Conditions (Optional)
Add Statement
You must enter one or more Principals.
Step 3: Generate Policy
A policy is a document (written in the Access Policy Language) that acts as a container for one or more statements.
Add one or more statements above to generate a policy.

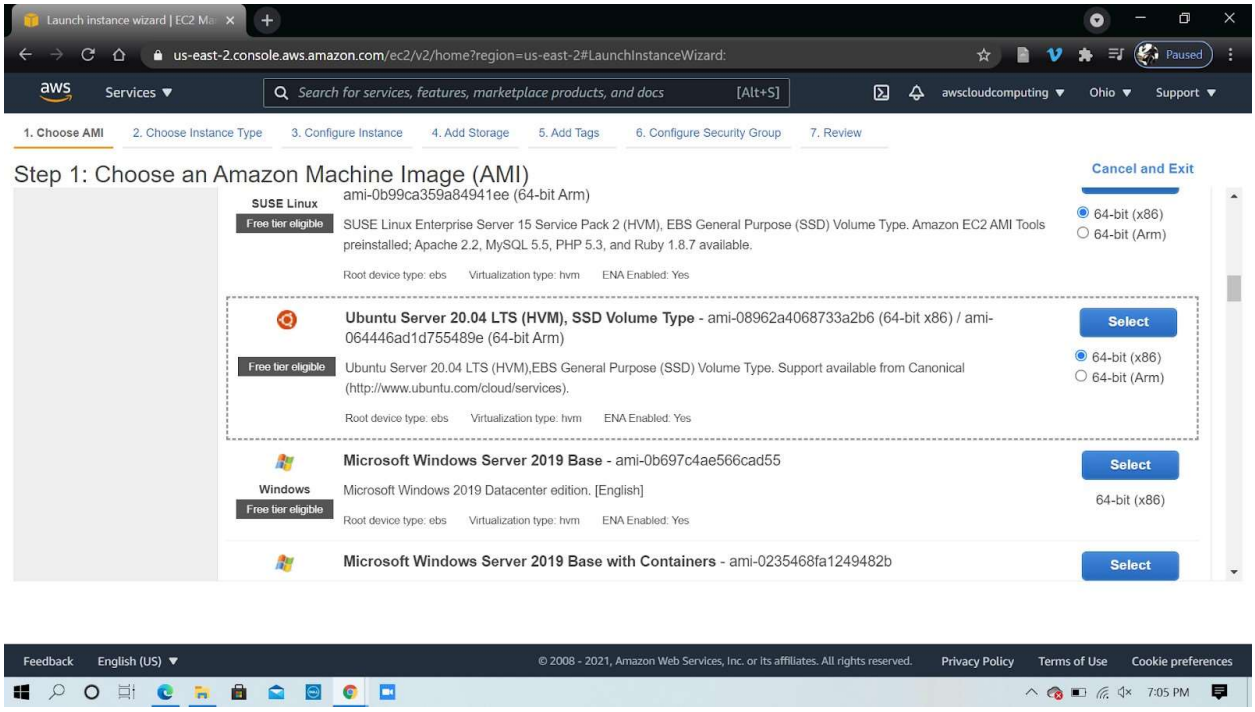

Bidding System
Add placement | Django site adm
biddingbucket - S3 bucket
AWS Policy Generator
awspolicygen.s3.amazonaws.com/policygen.html
Amazon Resource Name (ARN)
ARN should follow the following format: arn:aws:s3:::${BucketName}/${KeyName}.
Use a comma to separate multiple values.
Add Conditions (Optional)
Add Statement
You added the following statements. Click the button below to Generate a policy.
Principal(s) | Effect | Action | Resource | Conditions
* | Allow | s3:GetObject | arn:aws:s3:::biddingbucket/* | None
Step 3: Generate Policy
A policy is a document (written in the Access Policy Language) that acts as a container for one or more statements.
Generate Policy
Start Over
This AWS Policy Generator is provided for informational purposes only, you are still responsible for your use of Amazon Web Services technologies and ensuring that your use is in compliance with all applicable terms and conditions. This AWS Policy Generator is provided as is without warranty of any kind, whether express, implied, or statutory. This AWS Policy Generator does not modify the applicable terms and conditions governing your use of Amazon Web Services technologies.
©2010, Amazon Web Services LLC or its affiliates. All rights reserved.
An amazon.com company


Bidding System
Add placement | Django site adm
biddingbucket - S3 bucket
AWS Policy Generator
s3.console.aws.amazon.com/s3/buckets/biddingbucket?region=us-east-1&tab=permissions
Services
Search for services, features, blogs, docs, and more [Alt+S]
Global
dharmil.ds
We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose Provide feedback.
Provide feedback
Successfully edited bucket policy.
Follow security best practices for S3.
Learn more
Objects | Properties | Permissions | Metrics | Management | Access Points
Permissions overview
Access
Public
Block public access (bucket settings)
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

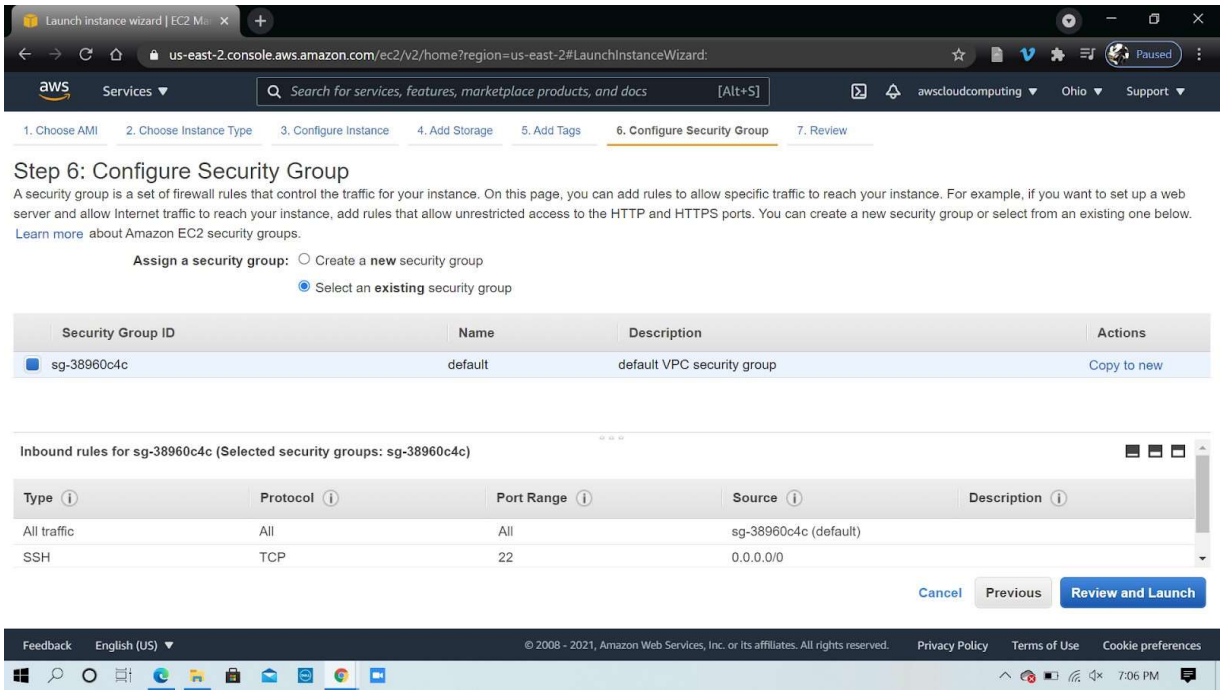


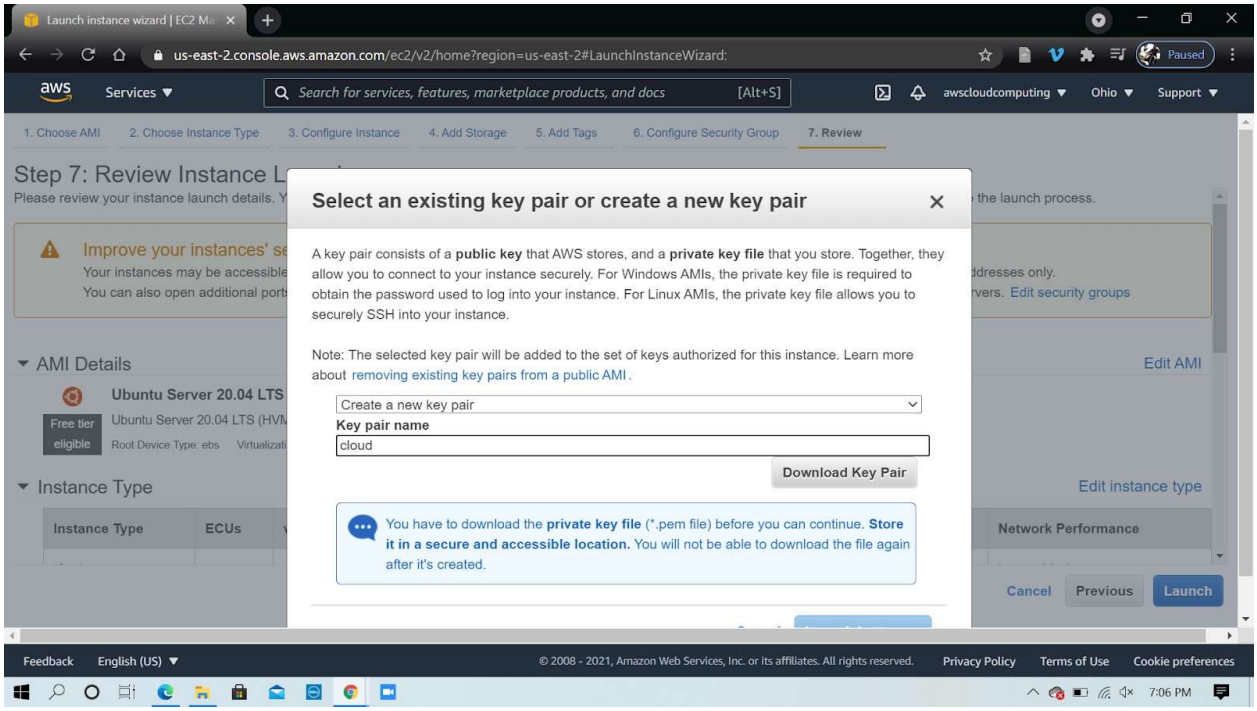
- Deploy a EC2 server

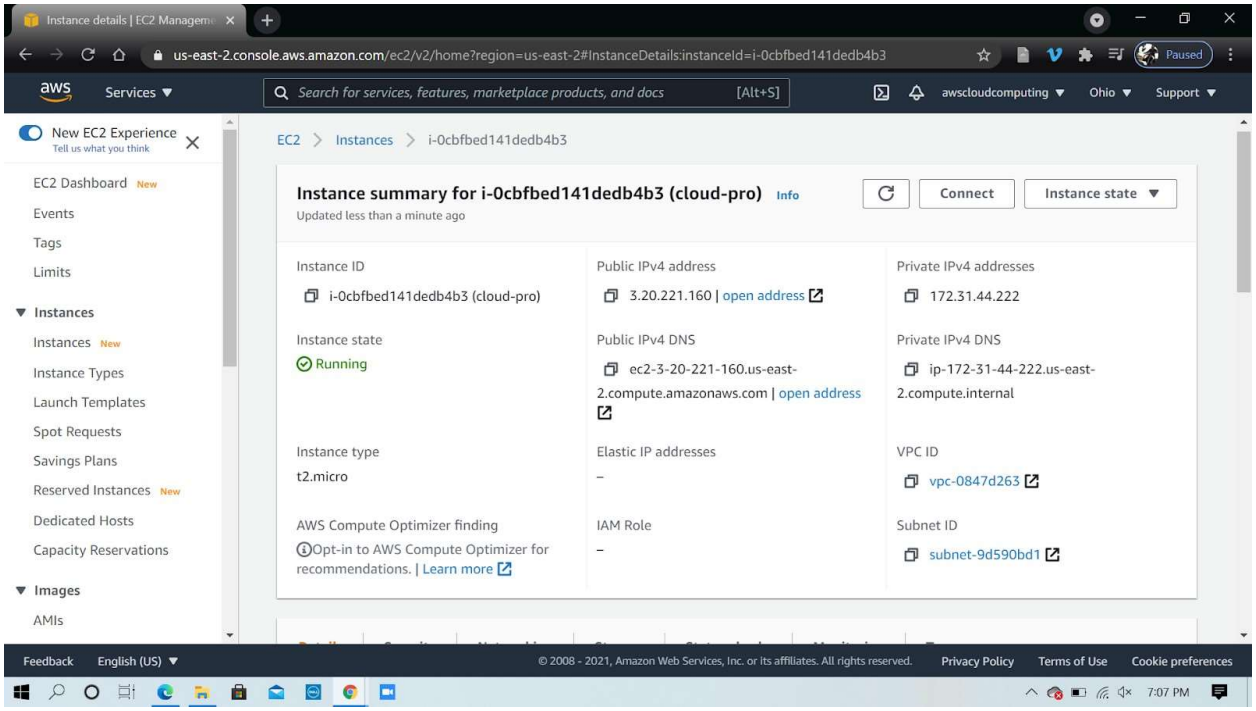
It is done by using Amazon EC2 service, for this we should select EC2 service from AWS management console dashboard. Then launch an instance, where the ubuntu server is used. Select all the desirable options, configure security groups, create a new key pair or select existing one if you have one, and then review and launch. Some key advantages are reliable, scalable and infrastructure on demand.


Launch instance wizard | EC2 Ma
us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard:
Paused
aws
Services
Search for services, features, marketplace products, and docs
[Alt+S]
awscloudcomputing
Ohio
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 1: Choose an Amazon Machine Image (AMI)
Cancel and Exit
SUSE Linux
Free tier eligible
ami-0b99ca359a84941ee (64-bit Arm)
SUSE Linux Enterprise Server 15 Service Pack 2 (HVM), EBS General Purpose (SSD) Volume Type. Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.5, PHP 5.3, and Ruby 1.8.7 available.
Root device type: ebs
Virtualization type: hvm
ENA Enabled: Yes
64-bit (x86)
64-bit (Arm)
Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-08962a4068733a2b6 (64-bit x86) / ami-064446ad1d755489e (64-bit Arm)
Select
Free tier eligible
Ubuntu Server 20.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).
Root device type: ebs
Virtualization type: hvm
ENA Enabled: Yes
64-bit (x86)
64-bit (Arm)
Microsoft Windows Server 2019 Base - ami-0b697c4ae566cad55
Select
Windows
Free tier eligible
Microsoft Windows 2019 Datacenter edition. [English]
Root device type: ebs
Virtualization type: hvm
ENA Enabled: Yes
64-bit (x86)
Microsoft Windows Server 2019 Base with Containers - ami-0235468fa1249482b
Select

Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences
7:05 PM


Launch instance wizard | EC2 Ma
us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard:
Paused
aws
Services
Search for services, features, marketplace products, and docs
[Alt+S]
awscloudcomputing
Ohio
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 6: Configure Security Group
A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below.
Learn more about Amazon EC2 security groups.
Assign a security group: Create a new security group
Select an existing security group
Security Group ID
Name
Description
Actions
sg-38960c4c
default
default VPC security group
Copy to new
Inbound rules for sg-38960c4c (Selected security groups: sg-38960c4c)
Type
Protocol
Port Range
Source
Description
All traffic
All
All
sg-38960c4c (default)
SSH
TCP
22
0.0.0.0/0
Cancel
Previous
Review and Launch
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences
7:06 PM


Launch instance wizard | EC2 Ma
us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard:
Paused
aws
Services
Search for services, features, marketplace products, and docs [Alt+S]
awscloudcomputing
Ohio
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 7: Review Instance L
Please review your instance launch details. Y
the launch process.
Improve your instances' se
Your instances may be accessible
You can also open additional ports
ddresses only.
rvers. Edit security groups
AMI Details
Edit AMI
Ubuntu Server 20.04 LTS
Free tier eligible
Ubuntu Server 20.04 LTS (HVM
Root Device Type: ebs
Virtualizati
Instance Type
Edit instance type
Instance Type
ECUs
Network Performance
Cancel
Previous
Launch
Select an existing key pair or create a new key pair
A key pair consists of a public key that AWS stores, and a private key file that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.
Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.
Create a new key pair
Key pair name
cloud
Download Key Pair
You have to download the private key file (*.pem file) before you can continue. Store it in a secure and accessible location. You will not be able to download the file again after it's created.
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences
7:06 PM

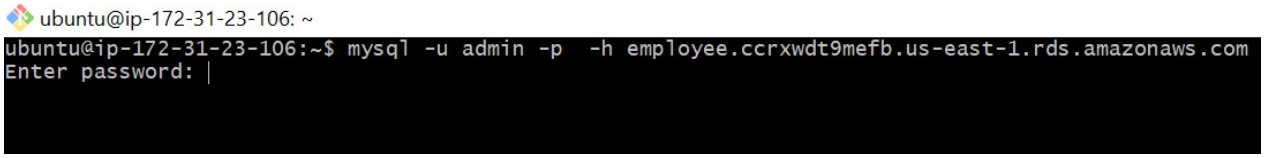

Instance details | EC2 Managem
us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#InstanceDetails:instanceId=i-0cbfbed141dedb4b3
Paused
aws
Services
Search for services, features, marketplace products, and docs [Alt+S]
awscloudcomputing
Ohio
Support
New EC2 Experience
Tell us what you think
EC2 Dashboard New
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Capacity Reservations
Images
AMIs
EC2 > Instances > i-0cbfbed141dedb4b3
Instance summary for i-0cbfbed141dedb4b3 (cloud-pro) Info
Updated less than a minute ago
Connect
Instance state
Instance ID
i-0cbfbed141dedb4b3 (cloud-pro)
Public IPv4 address
3.20.221.160 | open address
Private IPv4 addresses
172.31.44.222
Instance state
Running
Public IPv4 DNS
ec2-3-20-221-160.us-east-2.compute.amazonaws.com | open address
Private IPv4 DNS
ip-172-31-44-222.us-east-2.compute.internal
Instance type
t2.micro
Elastic IP addresses
-
VPC ID
vpc-0847d263
AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role
-
Subnet ID
subnet-9d590bd1
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences
7:07 PM

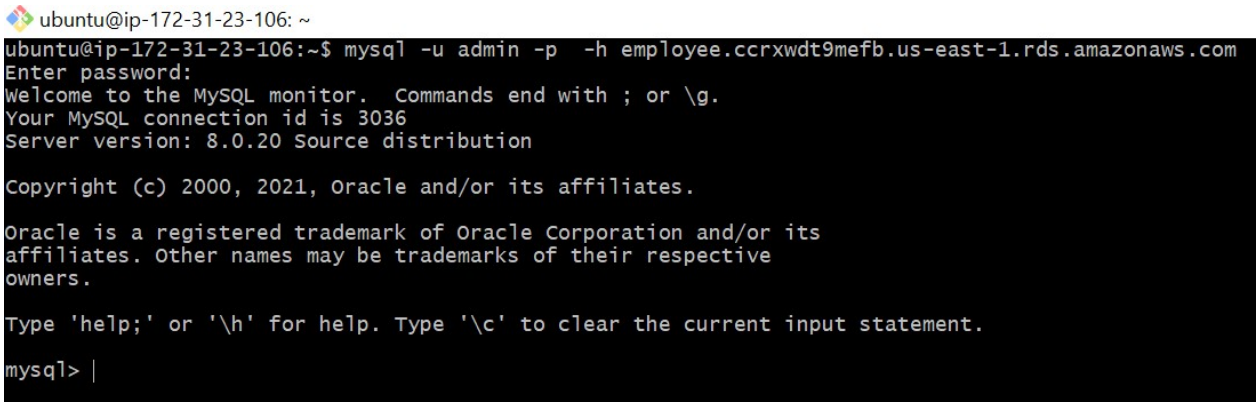
Next connect the server to our machine.

To connect from windows to server one can use Putty or Git bash or Super Putty which is a premium tool.
To connect from a mac one can use the terminal itself, this will connect to an instance deployed in AWS.

Next connect to our RDS

First deploy MySQL client on the server which helps to connect to our RDS machine, for this install MySQL client.





After installing MySQL client software now we connect to the RDS, so that now we will be in a MySQL shell.

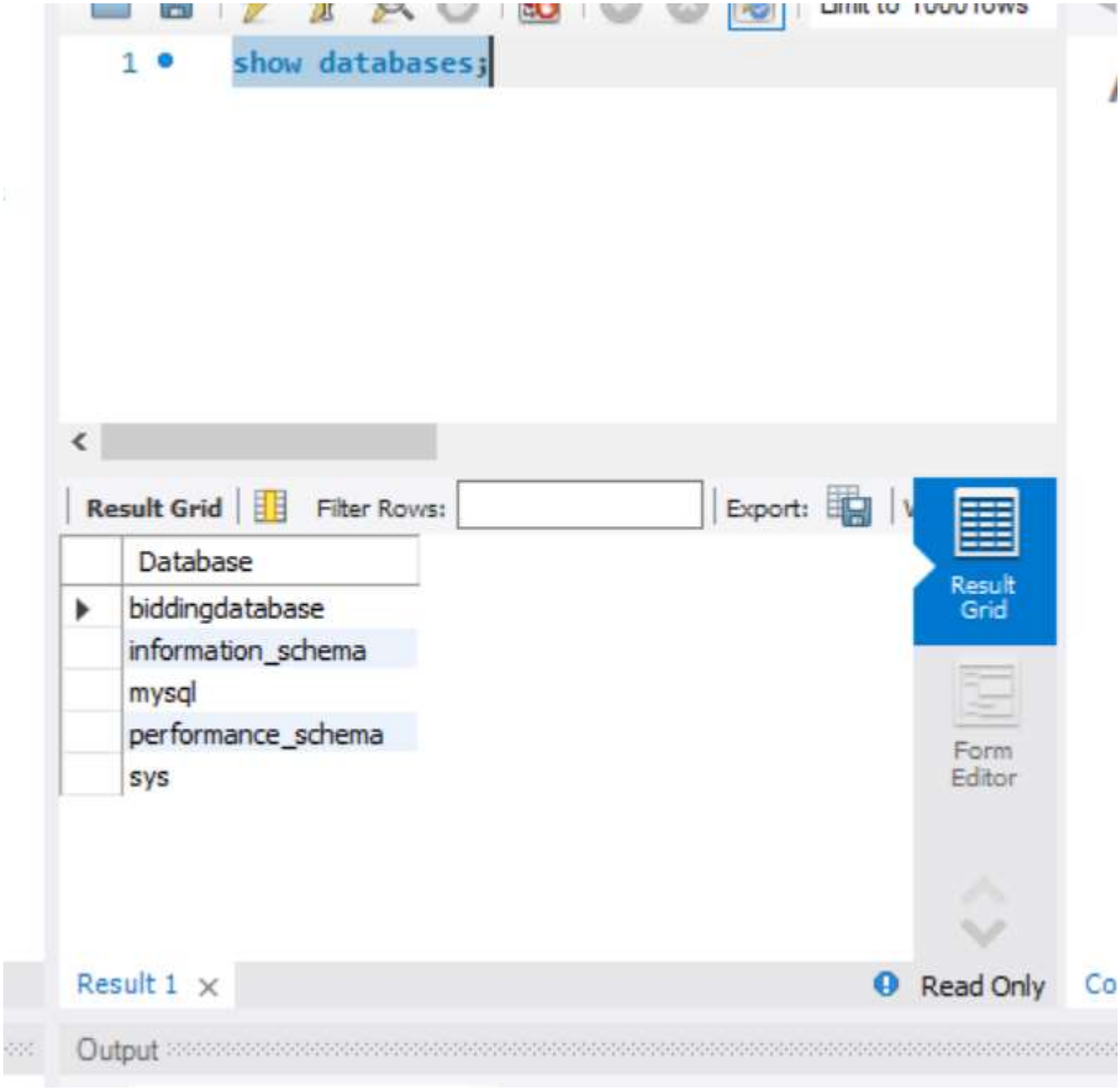
Next deploy database on server

For this first create a database and then create a table in that database for the information to be stored.

// commands

After logging into databases:

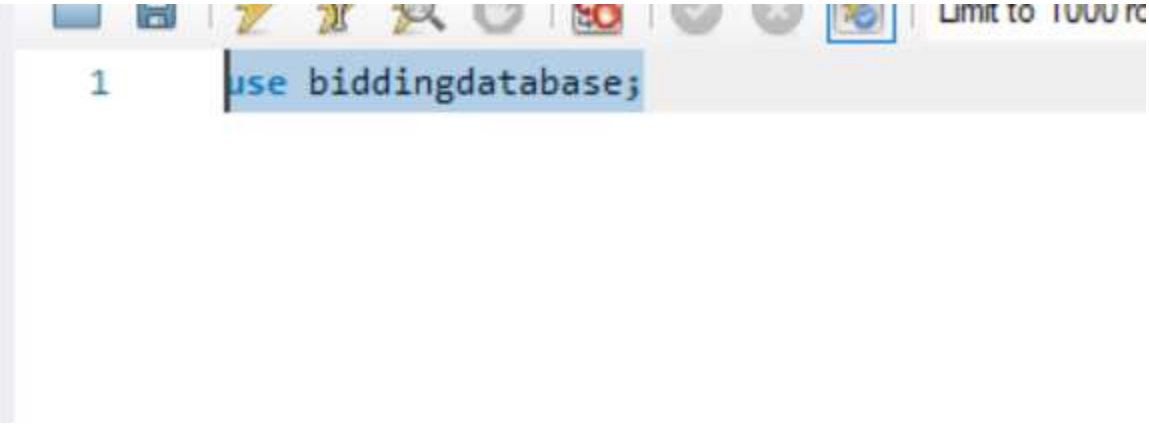
Show databases; // gives list of Databases present



create database biddingdatabase;// will create a database with the given name.

need to switch to created database)

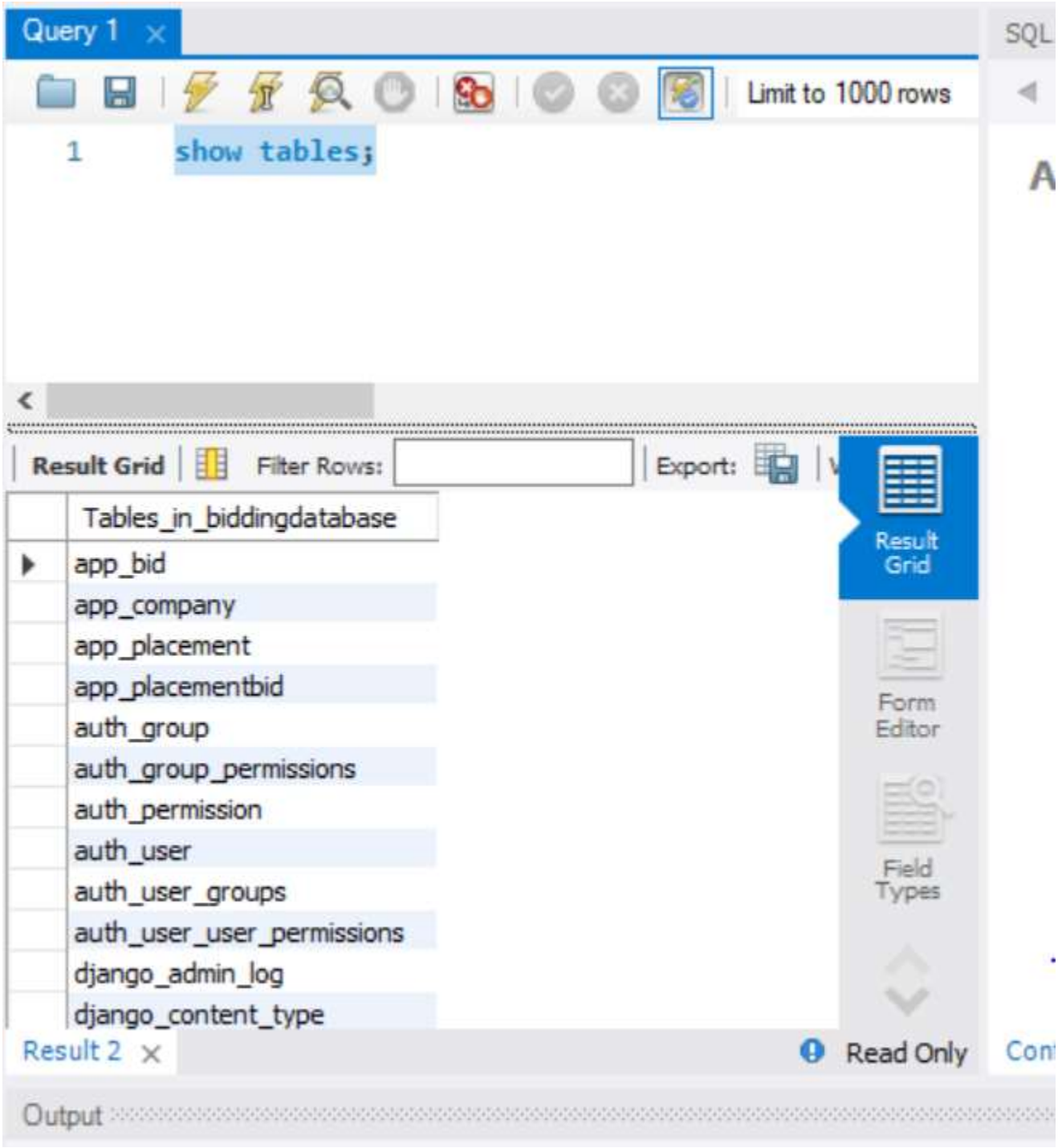
Use biddingdatabase;



(

(now we can create tables in the created database)

These are the tables that we created in our project :



2) The second phase is to build the website, configure the website codeand deploy the website on EC2 server.

- Build the website

Here to build the website we have used the Python Flask framework. DJango is known

asa simple but extensible framework.

- Configure the website code

1) Provide Host name of RDS in host region of config code.
2) Provide username, password and database name in config code.
3) Provide bucket name and region in the configuration code.

- Deploy website on EC2 server

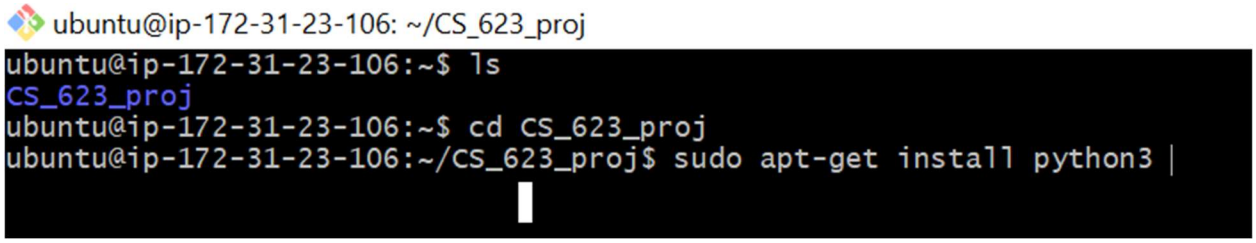
Make the website available on EC2 server.

After connecting to EC2 instance we need to clone a git repository:

Then navigate to the project folder and install the required packages:

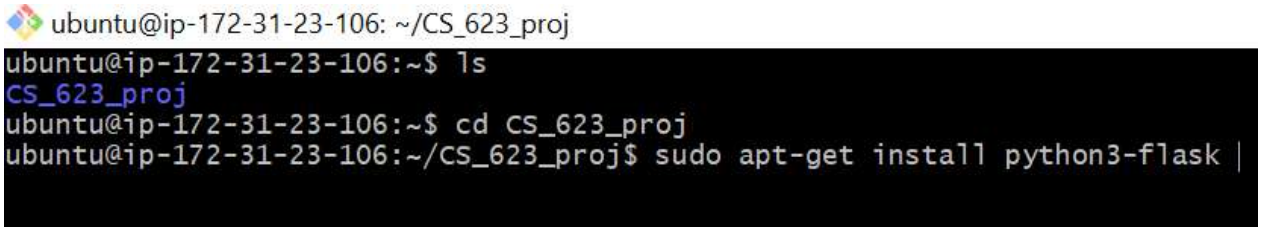
Execute all the below commands on the server to install them on server.

- sudo apt-get install python3



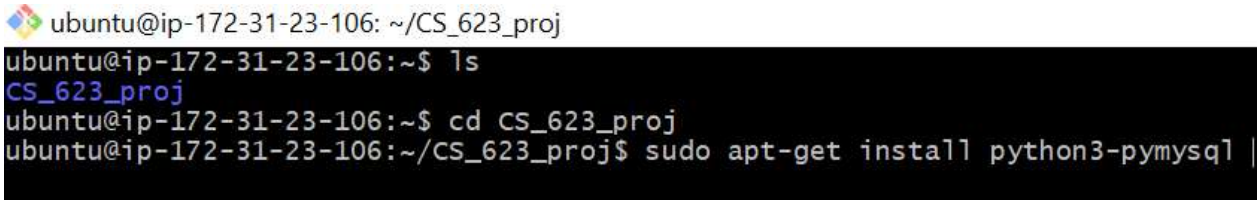
Python3 is the language that is coded for the website, we must install it on an existing system where we are running the website.

- sudo apt-get install python3-django



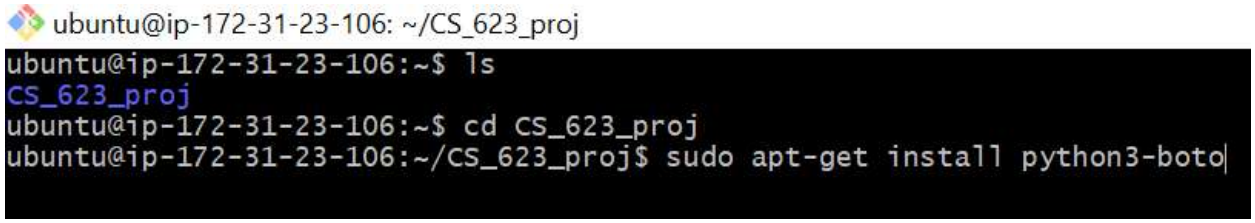
Flask helps to go ahead and publish a website on to that server.

- sudo apt-get install python3-pymysql



It is a library which helps the website to connect to the MySQL server.

- sudo apt-get install python3-boto



It is an SDK which helps to connect to S3 service where we upload files.

Then give EC2 instance or server access to basically upload data to S3. We need to do this because on the server we are running some code and that code is trying to interact with the AWS S3 service, when we connect with our RDS we should give authentication to our S3 service, so for that we must give the IAM role to our instance.

To give IAM role to our instance, follow the below steps -

- Select IAM service from AWS management console dashboard.
- Then create an admin role for our AWS account.
- Then select roles for our EC2 instance.

Then we should go to our EC2 service and click on the attach/replace IAM role and select the role which we created from it. Now EC2 is connected to RDS, EC2 is connected to S3, and website deployed in EC2 server.

3.1 Challenges team faced during building this project

- Different Amazon services being used in this project and connection to RDS from Git bash/ Putty is one of the challenges we faced.
- All the commands to install flask, boto and pymysql in the server.
- In the second phase initially we didn't give EC2 instance access to upload data to S3, after that we made a research on IAM roles and used IAM service to upload pictures to S3 bucket.
- Not able to decide whether to use elastic beanstalk or AWS EC2 for deploying our application.

CHAPTER 4: CODE & TECHNOLOGY

4.1 Programming language

1) Frontend

In frontend we used Python and Html languages.

2) Backend

In backend we used RDS (MySQL Database)

4.2 Cloud services

1) Amazon RDS:

We are using Amazon RDS service, which is used to operate and scale a relational database in the cloud. In this we are using MySQL database and this is used to store the entries. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.

2) Amazon S3:

We are using Amazon S3 service, which is a Storage for the internet. We can use it to store and retrieve any amount of data at any time, from anywhere on the web.

3) Amazon EC2 :

We are using Amazon EC2 service, it is a web service for launching and managing Linux/UNIX and Windows Server instances in Amazon's data centers.

4) Amazon IAM

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. When we connect with our RDS we should give authentication to our S3 service, so that we must give the IAM role to our instance.

CHAPTER 5: GIT ACCESS & SOURCE CODE & Youtube URL

Below is the GitHub Link for our project:

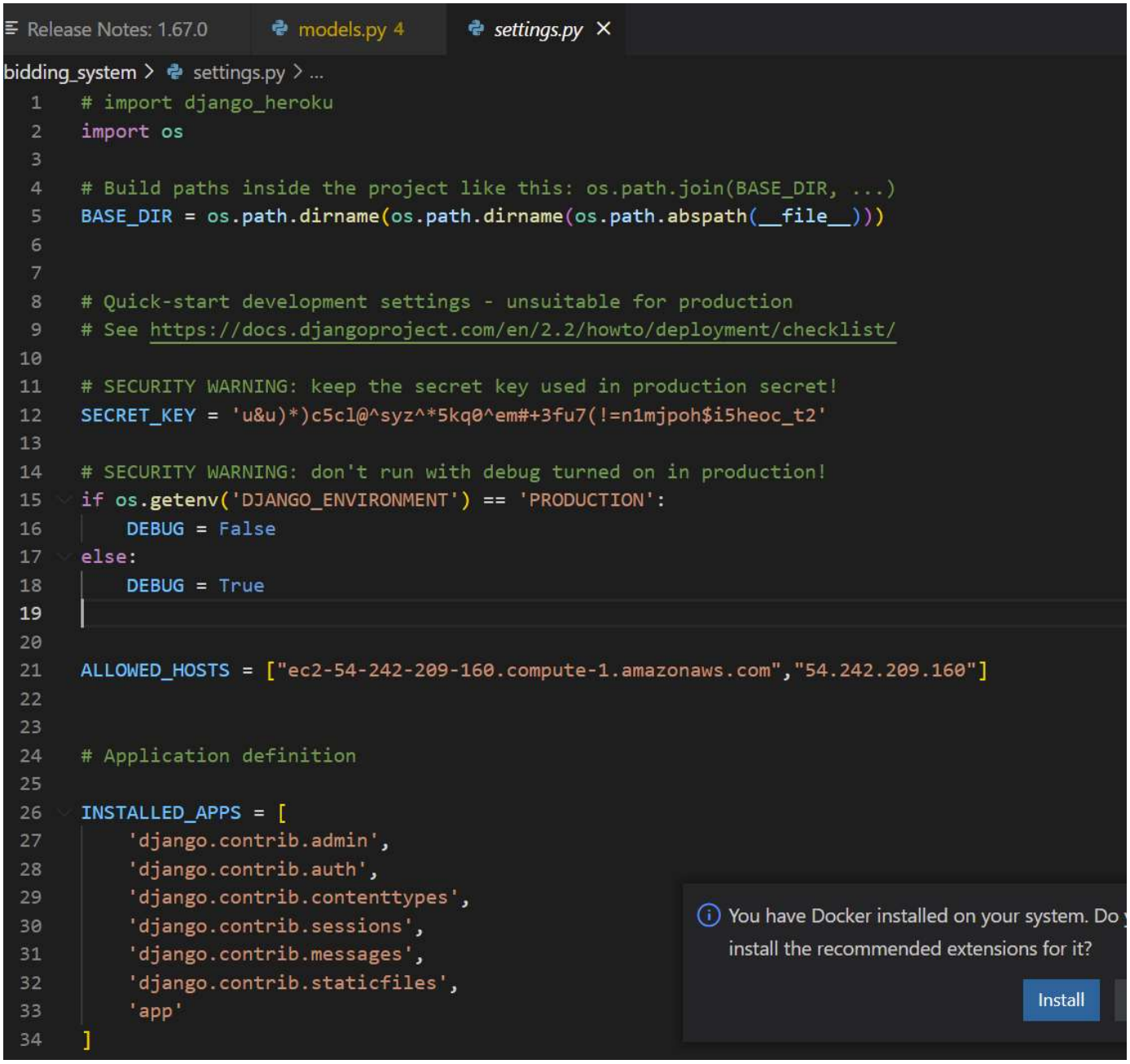
https://github.com/rishabparekh14/CS-632-CloudComputing-OnlineBiddingSystem

You tube URL :-

Please refer to below screenshot of our code for the modules of the page that we have shown below:

Python Code:

Settings.py :- It contains the code for connecting our application to any Database. May it be RDS or even local database


Release Notes: 1.67.0
models.py 4
settings.py
bidding_system > settings.py > ...
# import django_heroku
import os
# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/2.2/howto/deployment/checklist/
# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'u&u)*)c5cl@^syz^*5kq0^em#+3fu7(!=n1mjpoh$i5heoc_t2'
# SECURITY WARNING: don't run with debug turned on in production!
if os.getenv('DJANGO_ENVIRONMENT') == 'PRODUCTION':
DEBUG = False
else:
DEBUG = True
ALLOWED_HOSTS = ["ec2-54-242-209-160.compute-1.amazonaws.com","54.242.209.160"]
# Application definition
INSTALLED_APPS = [
'django.contrib.admin',
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles',
'app'
]
You have Docker installed on your system. Do
install the recommended extensions for it?
Install

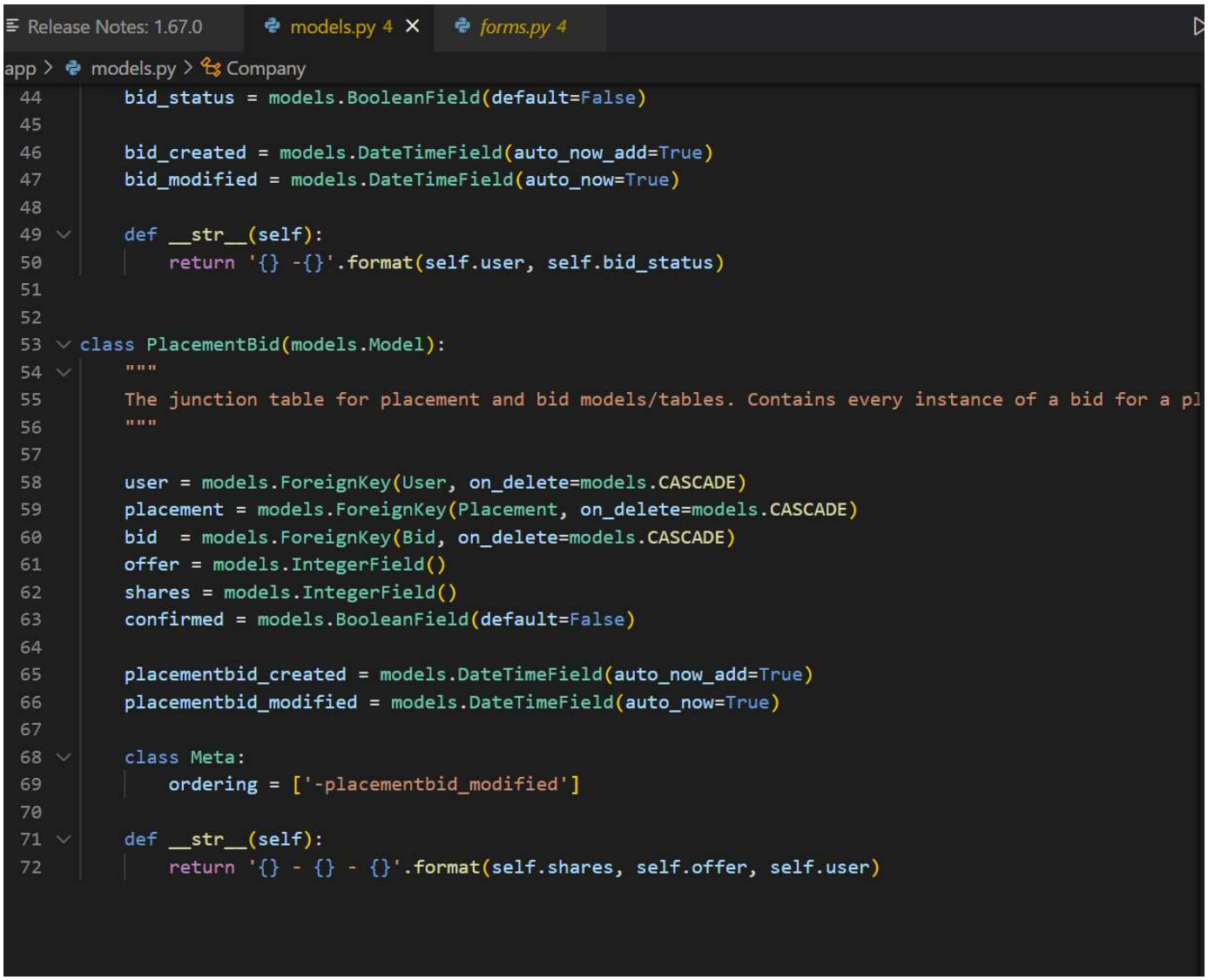
```
# DATABASES = {
#     'default': {
#     'ENGINE': 'django.db.backends.postgresql_psycopg2',
#     'NAME': 'my_db',
#     'USER' : 'hero',
#     'PASSWORD' : 'my_db@123',
#     'HOST' : 'localhost',
#     'PORT' : '5432',
# }
# }

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}


# Password validation
# https://docs.djangoproject.com/en/2.2/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
```

The above database shown in screenshot is dummy connection but we have actually connected to RDS. But, we have changed the database config and uploaded the code to EC2.

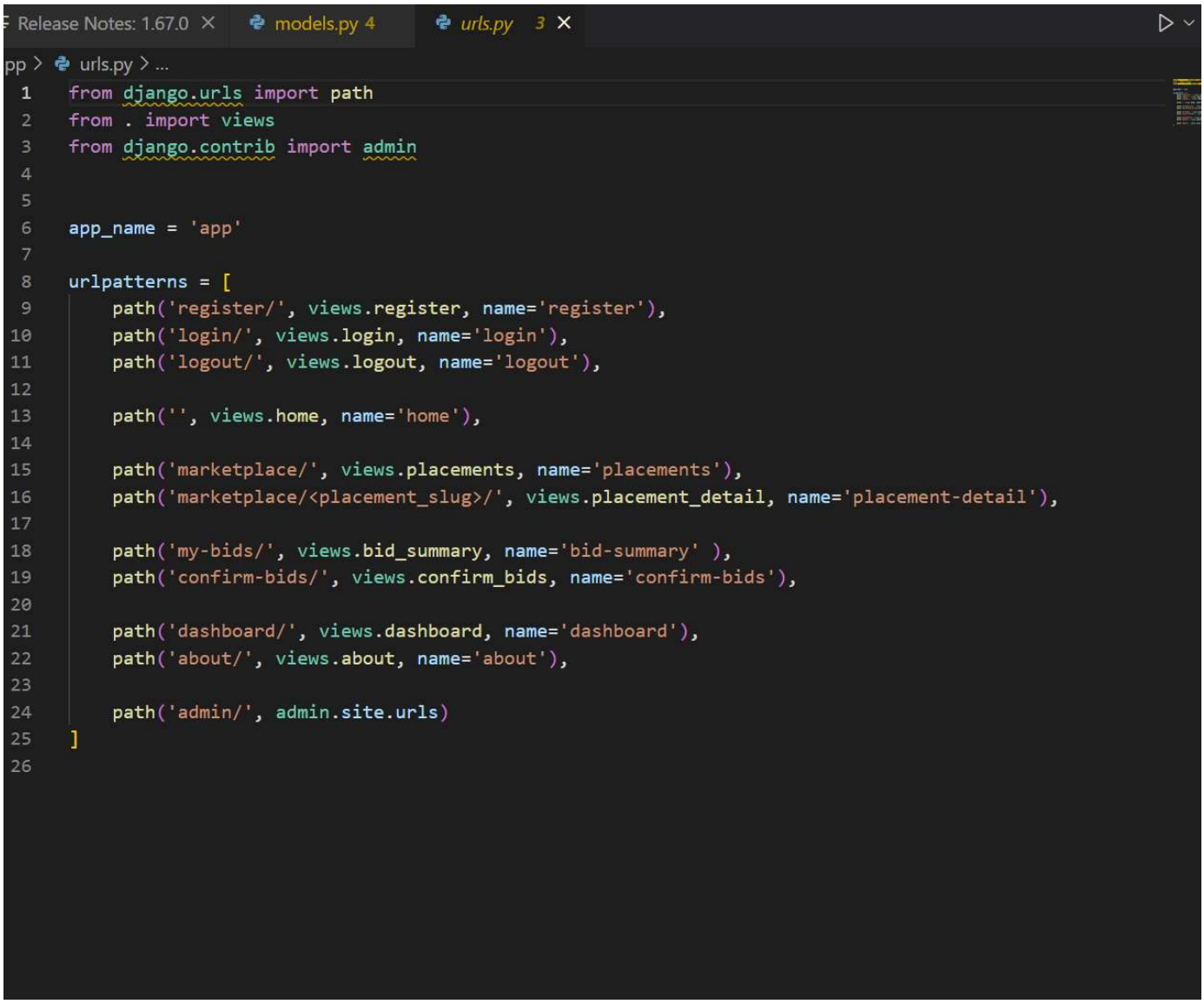
Models.py :- This contains information about the tables that we have created. The tables would be created based on the models mentioned in the below code.

app > models.py > Company

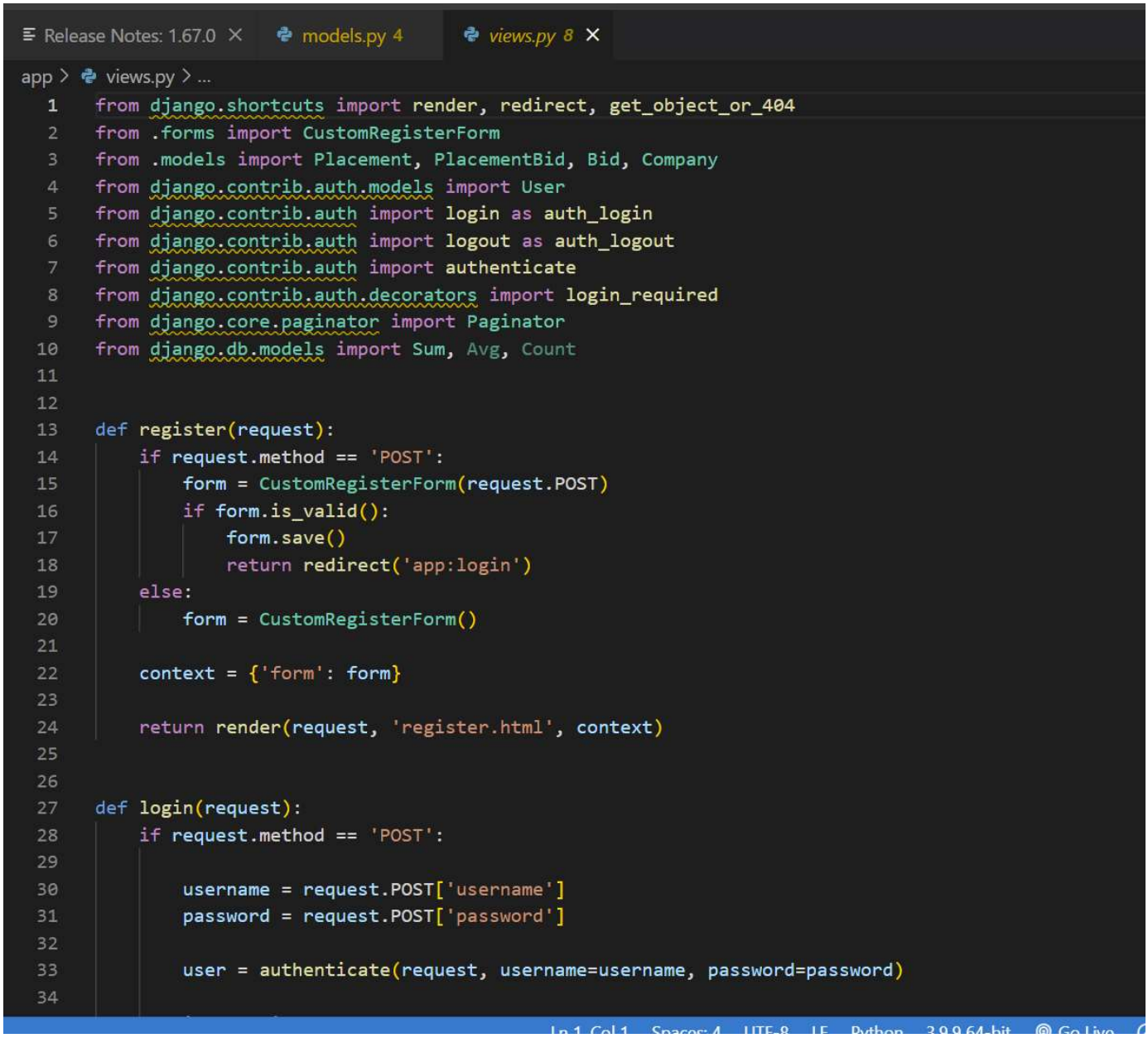
```python
from django.db import models
from django.conf import settings
from django.contrib.auth.models import User
# Create your models here.

class Company(models.Model):
    """
    Simply contains company details, referenced by Placement model
    """

    company_name = models.CharField(max_length=255)
    company_address = models.CharField(max_length=255)
    company_description = models.TextField(default="There is currently no description available for t

    company_created = models.DateTimeField(auto_now_add=True)
    company_modified = models.DateTimeField(auto_now=True)

    def __str__(self):
        return self.company_name


class Placement(models.Model):
    """
    A placement allows investors to bid on company capital raise
    """

    placement_title = models.CharField(max_length=255)
    placement_slug = models.SlugField()
    placement_company = models.ForeignKey(Company, on_delete=models.CASCADE)


    placement_created = models.DateTimeField(auto_now_add=True)
    placement_modified = models.DateTimeField(auto_now=True)

```



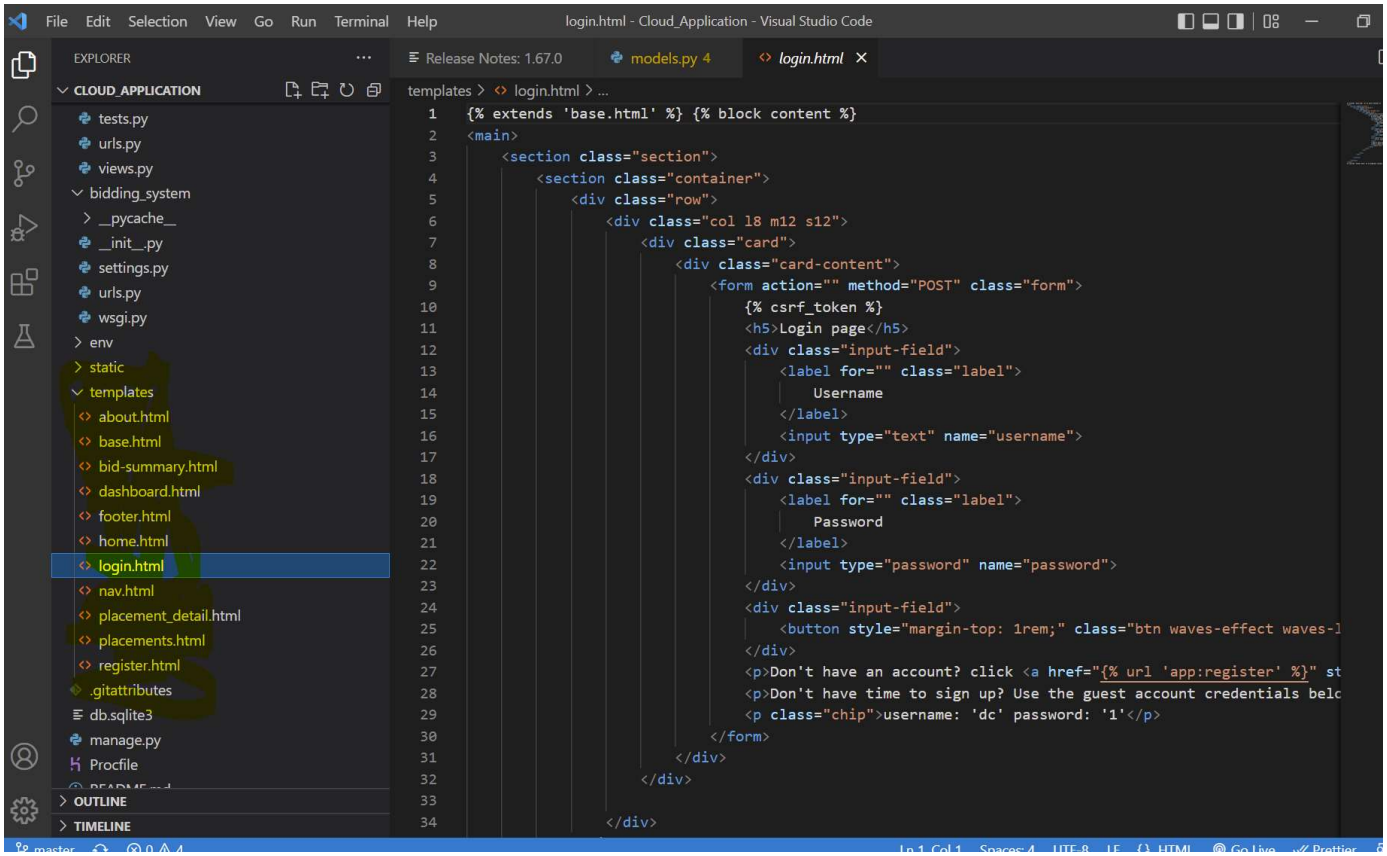
URLS.py :- This contains the url routes that are present in our application



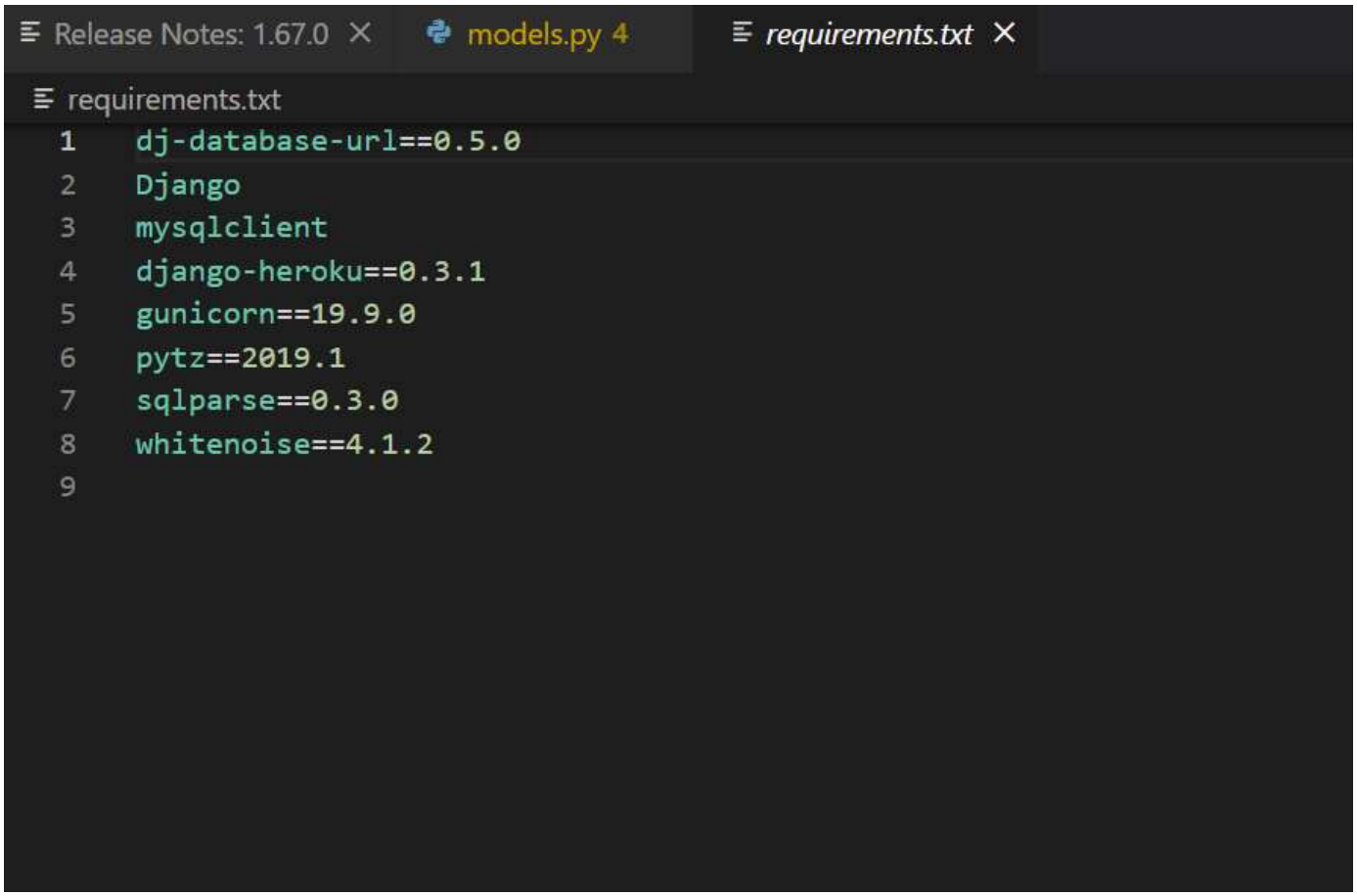
Views.py :- It contains the information of the API that we are using as well as contains the information about of the routing that what should be response after the page that we are passing. Also, it contains the logic of the html responses of the page i.e. it connects the html with the business logic as well as the database



Template folder:- The files highlighted on the left are all the html files which contains the code related to views of every page.

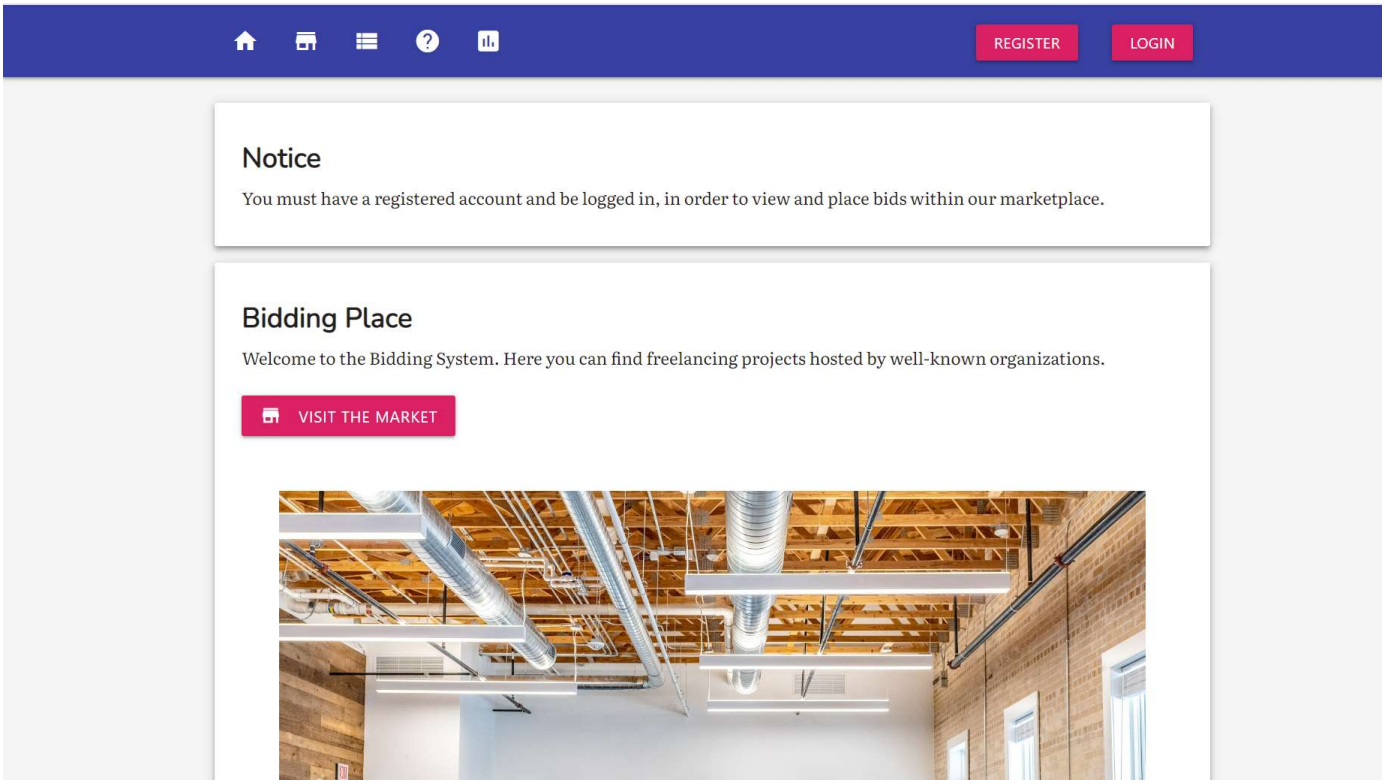


Requirement.txt : It contains the configuration needed for your application.

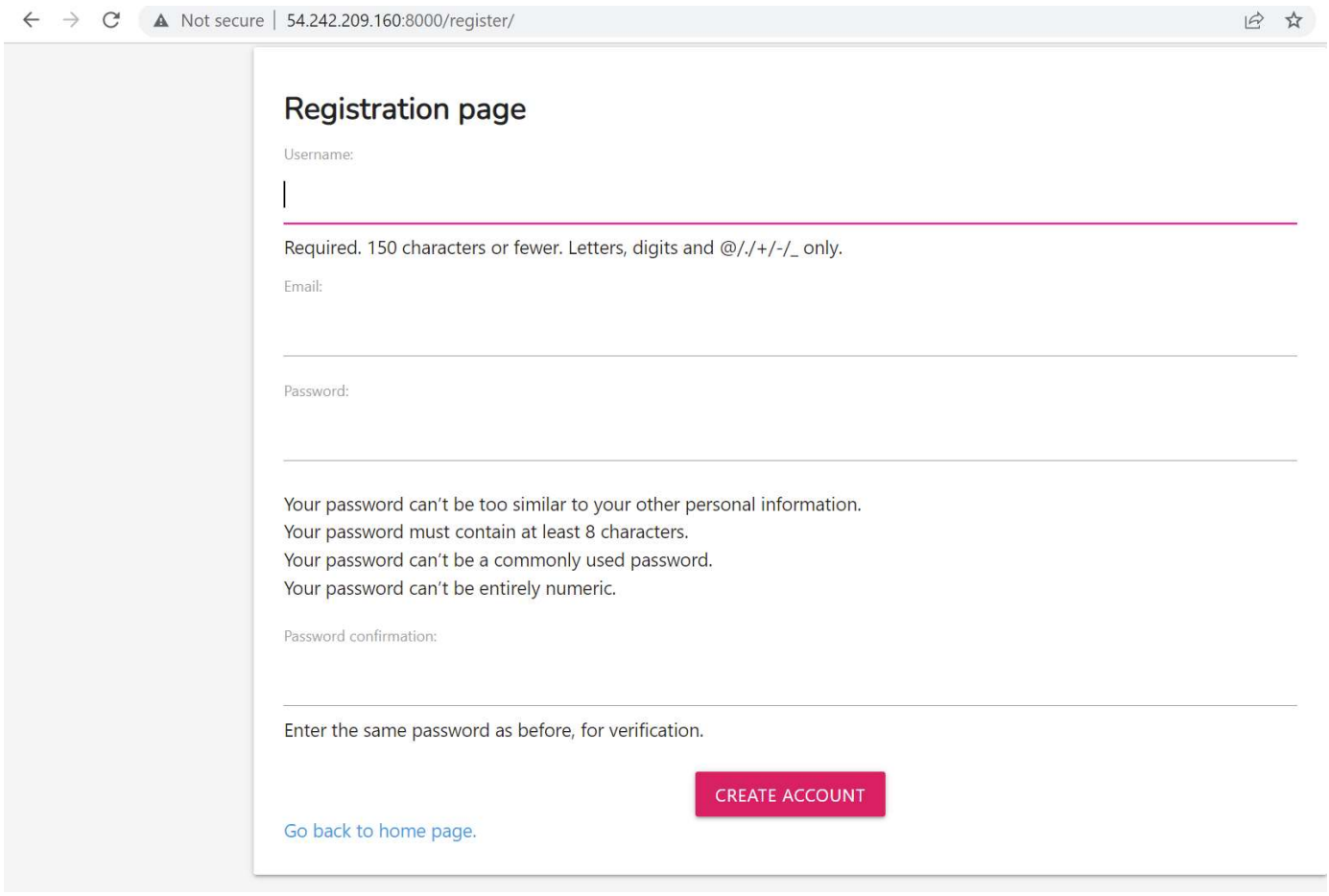

Release Notes: 1.67.0
models.py 4
requirements.txt
requirements.txt
1 dj-database-url==0.5.0
2 Django
3 mysqlclient
4 django-heroku==0.3.1
5 gunicorn==19.9.0
6 pytz==2019.1
7 sqlparse==0.3.0
8 whitenoise==4.1.2
9

CHAPTER 6: DEMO SCREENSHOTS FROM WEBSITE

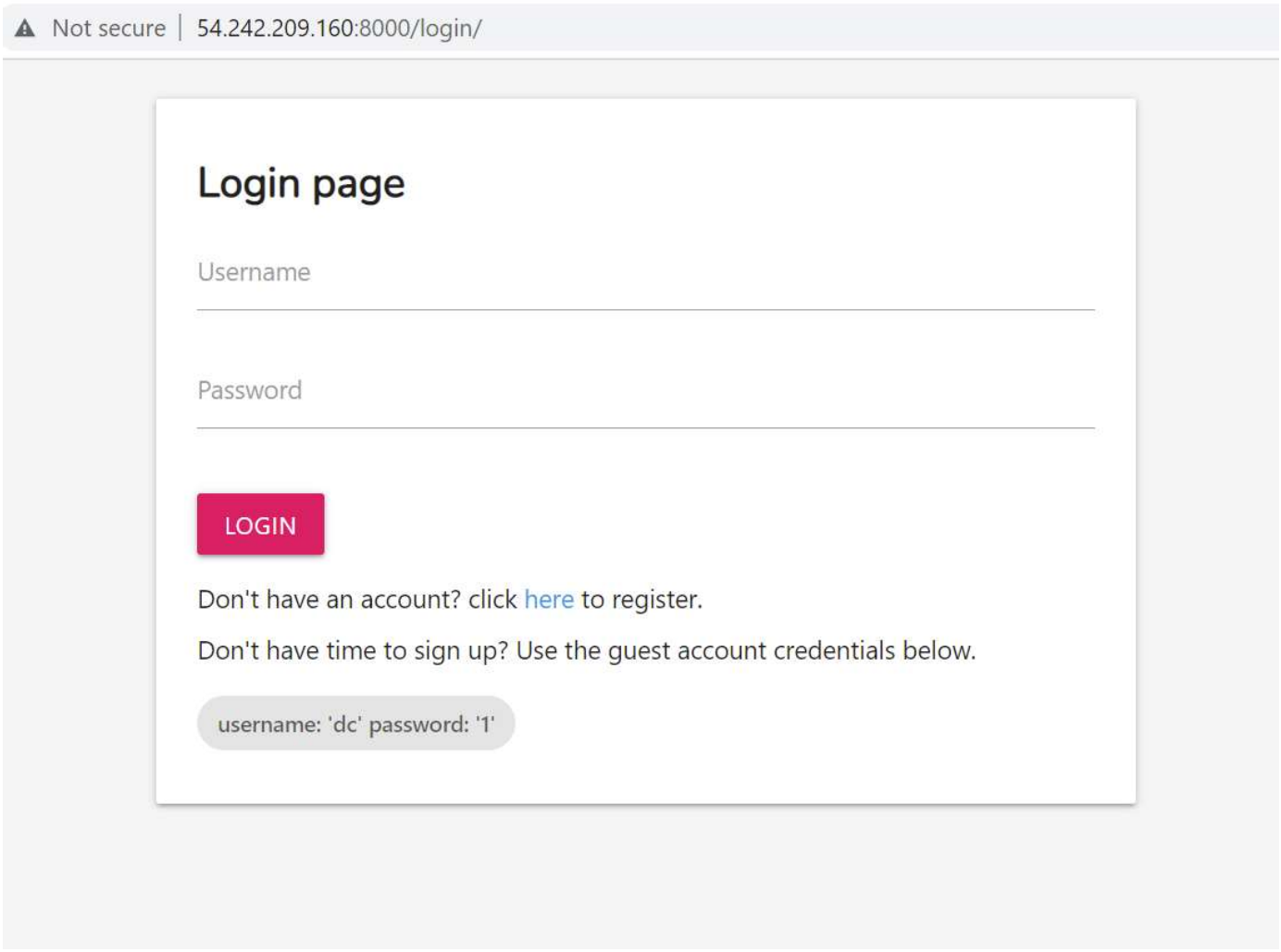
Home Page:



Register as a bidder Page:

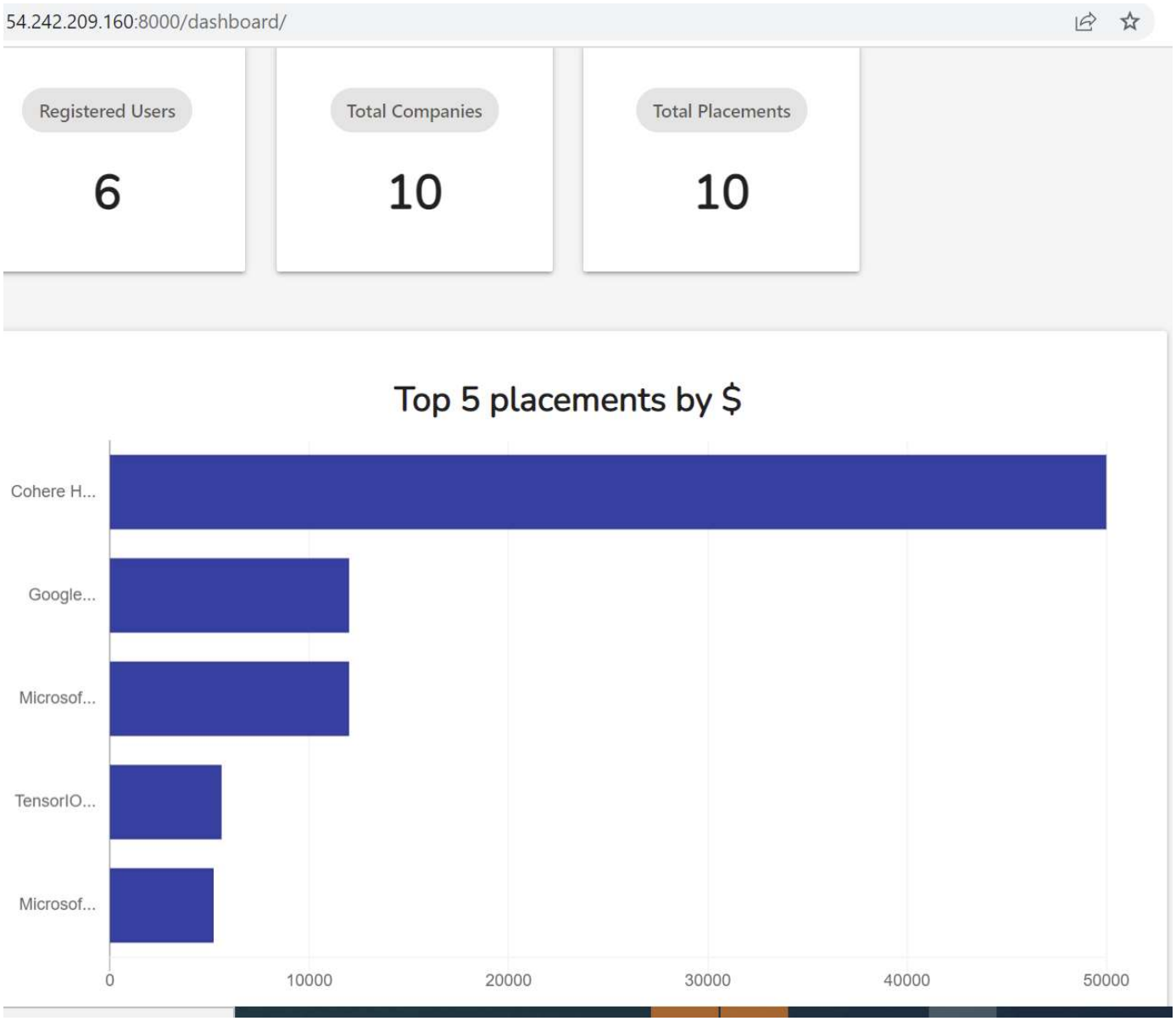

Not secure | 54.242.209.160:8000/register/
Registration page
Username:
Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.
Email:
Password:
Your password can't be too similar to your other personal information.
Your password must contain at least 8 characters.
Your password can't be a commonly used password.
Your password can't be entirely numeric.
Password confirmation:
Enter the same password as before, for verification.
CREATE ACCOUNT
Go back to home page.

Login Page:

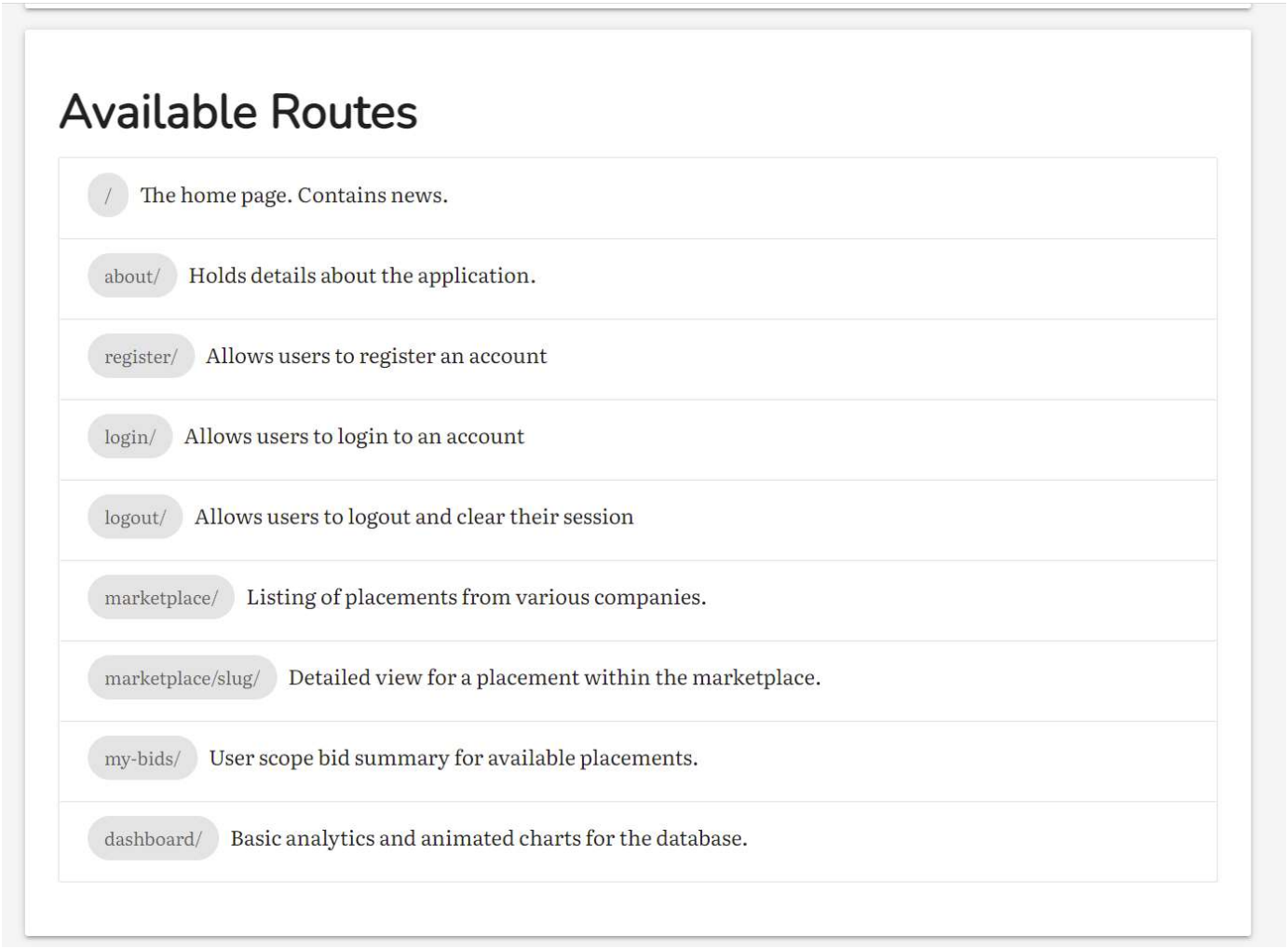


Statistics Page:-

This page shows the statistics of the amount of bids made by a certain Company.

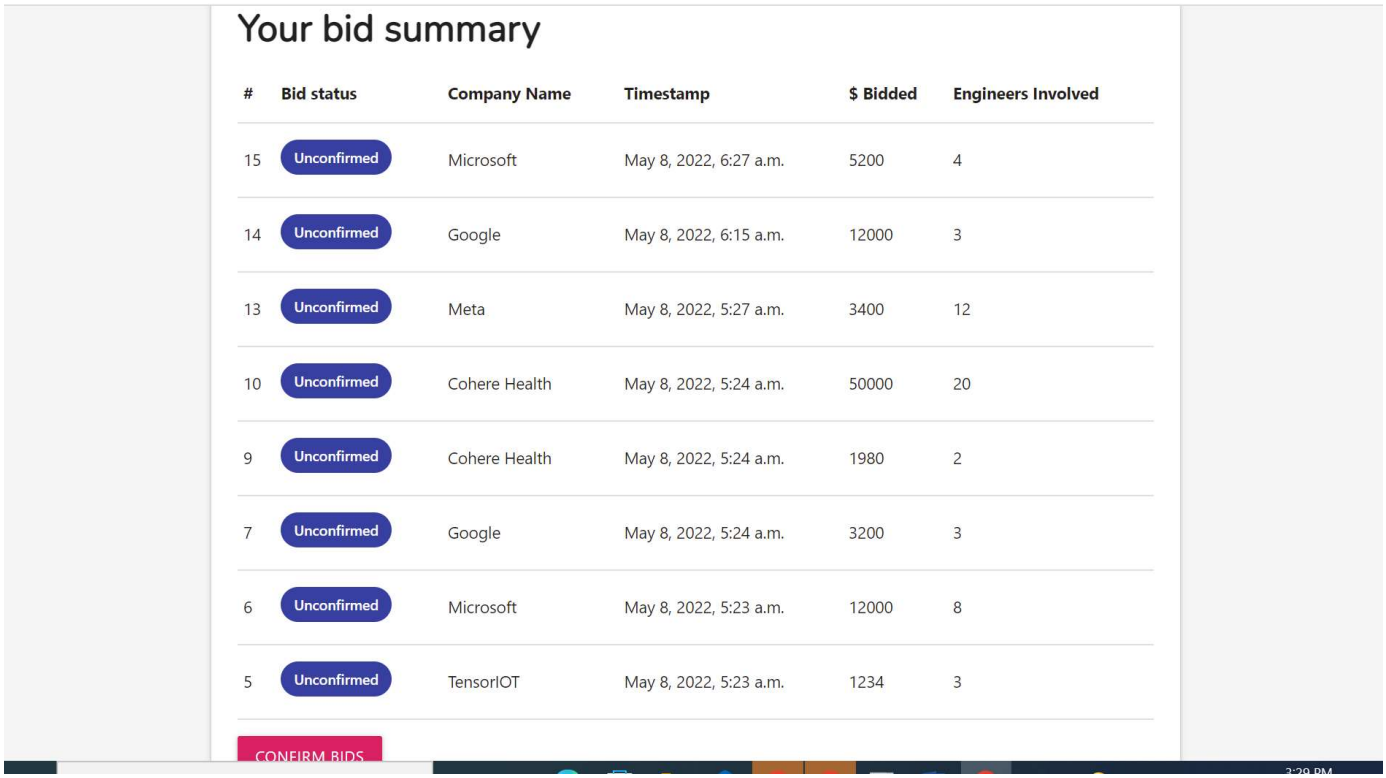


About us Page which contains the details of all the routes:



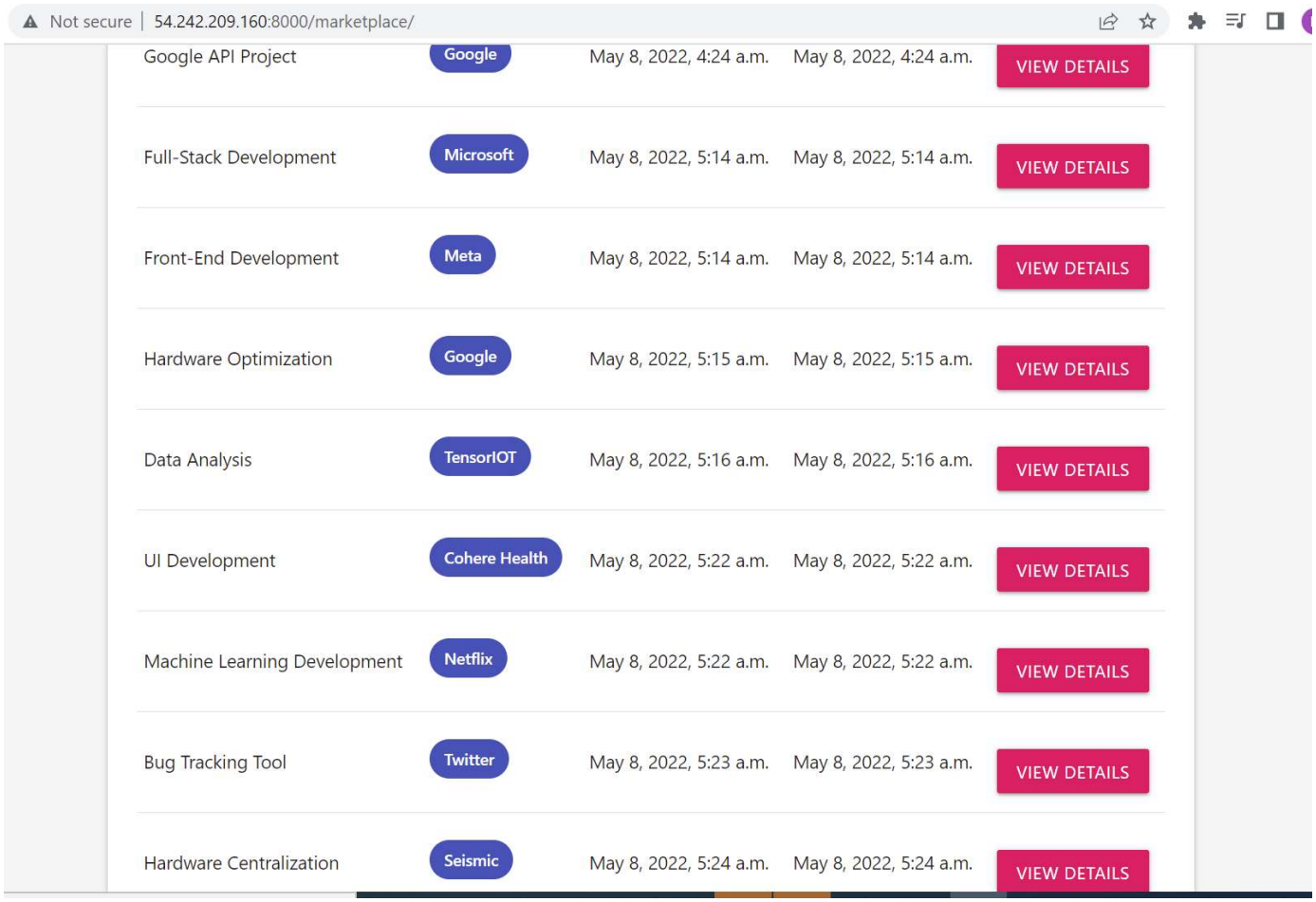
Bids page :

Bids made by us for the project posted by a company.



Bids Proposal Page:-

This page contains the bids placed by company which can looked by everyone who logs in.


Not secure | 54.242.209.160:8000/marketplace/
Google API Project
Google
May 8, 2022, 4:24 a.m.
May 8, 2022, 4:24 a.m.
VIEW DETAILS
Full-Stack Development
Microsoft
May 8, 2022, 5:14 a.m.
May 8, 2022, 5:14 a.m.
VIEW DETAILS
Front-End Development
Meta
May 8, 2022, 5:14 a.m.
May 8, 2022, 5:14 a.m.
VIEW DETAILS
Hardware Optimization
Google
May 8, 2022, 5:15 a.m.
May 8, 2022, 5:15 a.m.
VIEW DETAILS
Data Analysis
TensorIOT
May 8, 2022, 5:16 a.m.
May 8, 2022, 5:16 a.m.
VIEW DETAILS
UI Development
Cohere Health
May 8, 2022, 5:22 a.m.
May 8, 2022, 5:22 a.m.
VIEW DETAILS
Machine Learning Development
Netflix
May 8, 2022, 5:22 a.m.
May 8, 2022, 5:22 a.m.
VIEW DETAILS
Bug Tracking Tool
Twitter
May 8, 2022, 5:23 a.m.
May 8, 2022, 5:23 a.m.
VIEW DETAILS
Hardware Centralization
Seismic
May 8, 2022, 5:24 a.m.
May 8, 2022, 5:24 a.m.
VIEW DETAILS

CHAPTER 7: OUTSTANDING FEATURES OF THIS PROJECT

1. Usage of different cloud services viz, EC2, S3, RDS , IAM
2. Elimination of Infrastructure, and upfront investment.
3. Enhanced level of flexibility and scalability in comparison to traditional datacentres.
4. Storing and retrieving of data in a fast and efficient manner.
5. Simple Project management.
6. Accessible from anywhere in the world and Increased productivity.

CHAPTER 8: FUTURE ENHANCEMENTS

1. Implementation of AWS DynamoDB instead of MYSQL database, as AWS DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale.DynamoDB can handle more than 10 trillion requests per day and can support peaks of more than 20 million requests per second. So, AWS DynamoDB can be one of our future enhancements.

CHAPTER 9: REFERENCES

1. https://docs.aws.amazon.com/
2. https://docs.aws.amazon.com/s3/index.html
3. https://docs.aws.amazon.com/pythonsdk/
4. [2] "Introducing Python: modern computing in simple packages (Second edition.)", Bill Lubanovic, O'Reilly Media, November, 2019., Chapter 18. This is a book that discusses the basic usage of Python programming language within various topics, including the web systems. Bill Lubanovic is a senior software engineer currently working for Internet Archive. O'Reilly is a popular publisher that has published various books related to software development.