Repo link: [Zamshed87/Dockerized-Django-Application-Deployment-on-AWS-EC2-Kubernetes]

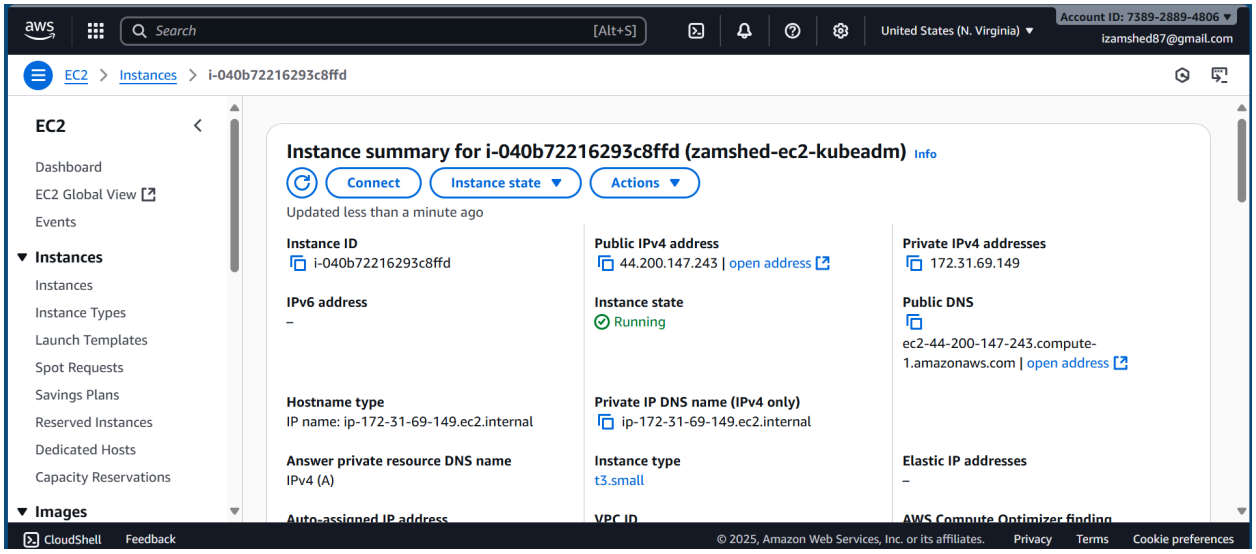
Docker hub image link : [zamshed/django-k8s-web general | Docker Hub]

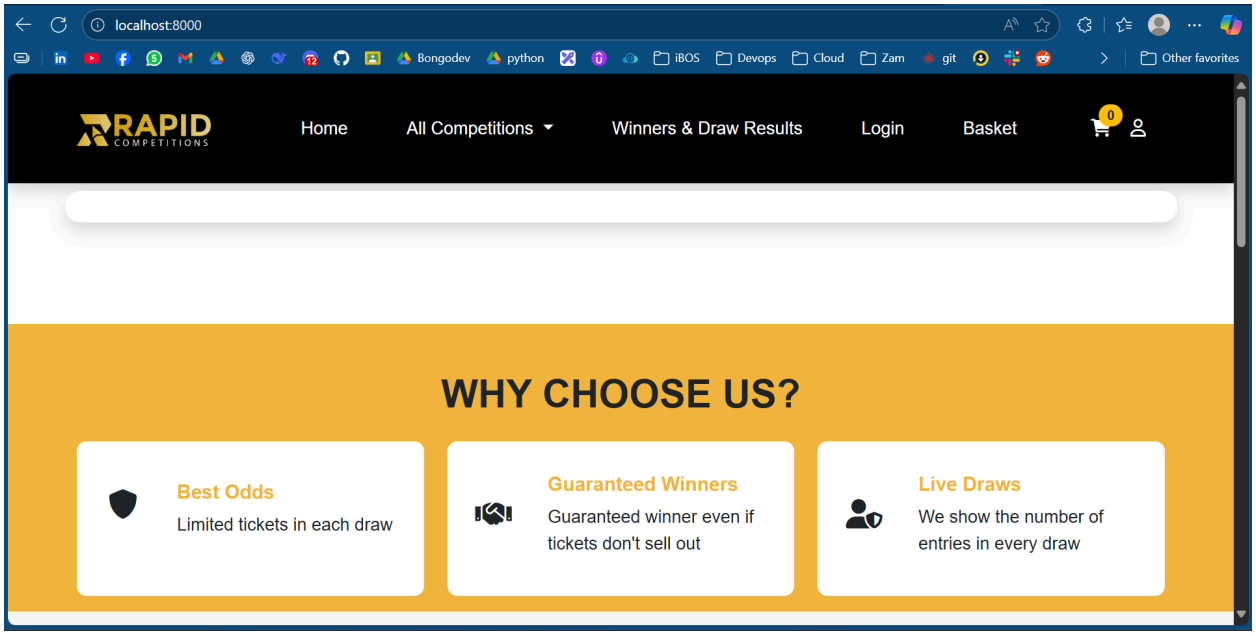
pip install dj-database-url python-decouple

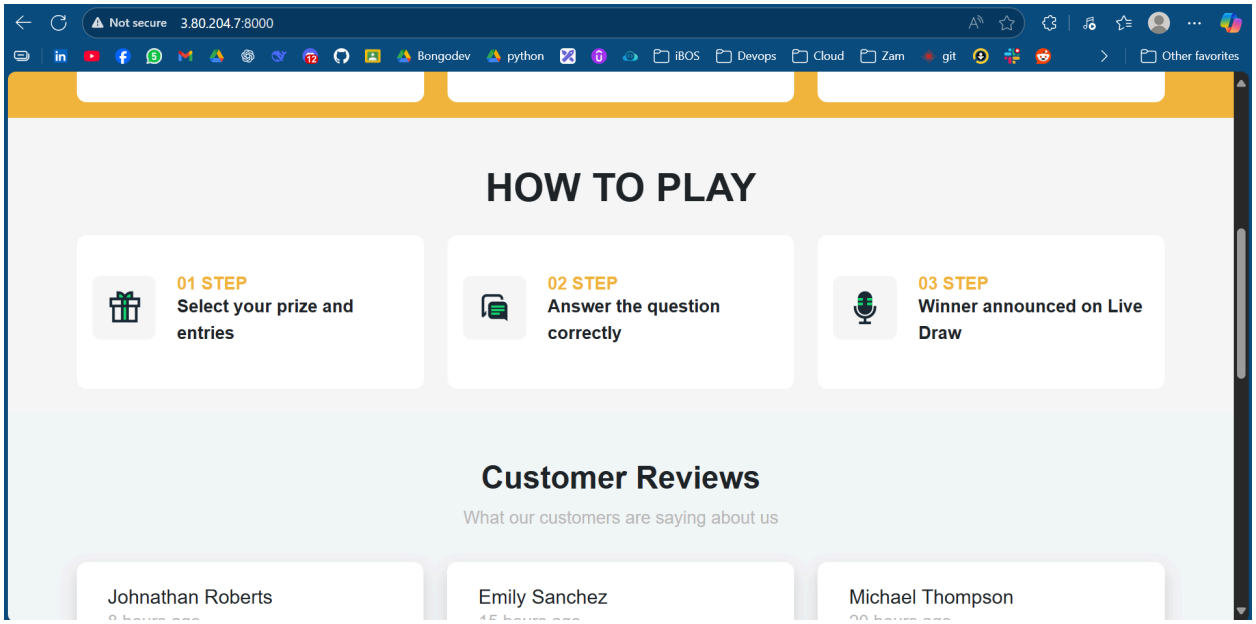
pip freeze | grep -E "dj-database-url|python-decouple" >> requirements.txt

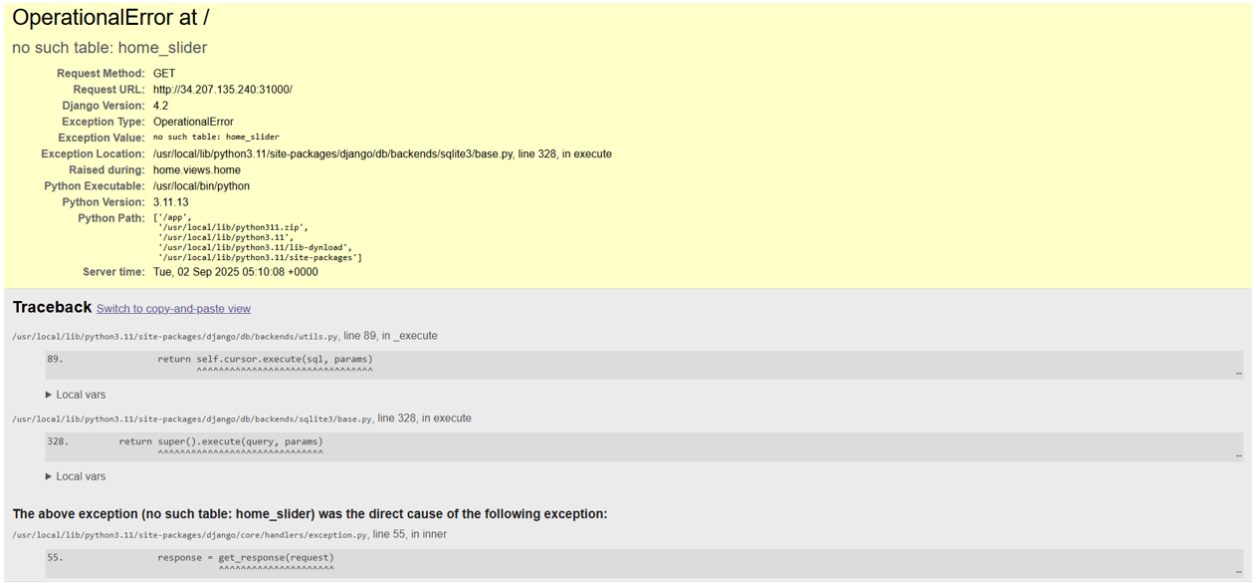
```
root@Zamshed-Dell:/mnt/d/Dockerized-Django-Application-Deployment-on-AWS-EC2-Kubernetes# docker compose up -d --build
WARN[0000] /mnt/d/Dockerized-Django-Application-Deployment-on-AWS-EC2-Kubernetes/docker-compose.yml: the attribute `version` is obsol
ete, it will be ignored, please remove it to avoid potential confusion
#1 [internal] load local bake definitions
#1 reading from stdin 689B done
#1 DONE 0.0s

#2 [internal] load build definition from Dockerfile
#2 transferring dockerfile: 730B 0.0s done
#2 DONE 0.0s

#3 [internal] load metadata for docker.io/library/python:3.11-slim
#3 DONE 1.1s

#4 [internal] load .dockerignore
#4 transferring context: 153B 0.0s done
#4 DONE 0.0s

#5 [1/8] FROM docker.io/library/python:3.11-slim@sha256:1d6131b5d479888b43200645e03a78443c7157efbdb730e6b48129740727c312
#5 DONE 0.0s

#6 [internal] load build context
#6 transferring context: 9.99kB 0.7s done
#6 DONE 0.7s

#7 [2/8] WORKDIR /app
#7 CACHED
```

```
root@Zamshed-Dell: /mnt/d/

root@Zamshed-Dell:/mnt/d/Dockerized-Django-Application-Deployment-on-AWS-EC2-Kubernetes# docker ps
CONTAINER ID   IMAGE                                                                COMMAND                  CREATED              STA
TUS                  PORTS                                          NAMES
115f7d2c80c3   dockerized-django-application-deployment-on-aws-ec2-kubernetes-web   "python manage.py ru…"   About a minute ago   Up
38 seconds           0.0.0.0:8000->8000/tcp, [::]:8000->8000/tcp    dockerized-django-application-deployment-on-aws-ec2-kubernetes-
web-1
8c296163c26b   postgres:13                                                          "docker-entrypoint.s…"   About a minute ago   Up
38 seconds           5432/tcp                                       dockerized-django-application-deployment-on-aws-ec2-kubernetes-
db-1
b0938ddb6ba3   dockerizing-express-js-project-app                                   "docker-entrypoint.s…"   5 weeks ago          Up
5 minutes (unhealthy)   0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp   dockerizing-express-js-project-app-1
root@Zamshed-Dell:/mnt/d/Dockerized-Django-Application-Deployment-on-AWS-EC2-Kubernetes#
```


Search [Alt+S]
United States (N. Virginia)
Account ID: 7389-2889-4806
izamshed87@gmail.com
EC2 > Instances > i-040b72216293c8ffd
EC2
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
Instance summary for i-040b72216293c8ffd (zamshed-ec2-kubeadm) Info
Connect
Instance state
Actions
Updated less than a minute ago
Instance ID
i-040b72216293c8ffd
Public IPv4 address
44.200.147.243 | open address
Private IPv4 addresses
172.31.69.149
IPv6 address
–
Instance state
Running
Public DNS
ec2-44-200-147-243.compute-1.amazonaws.com | open address
Hostname type
IP name: ip-172-31-69-149.ec2.internal
Private IP DNS name (IPv4 only)
ip-172-31-69-149.ec2.internal
Answer private resource DNS name
IPv4 (A)
Instance type
t3.small
Elastic IP addresses
–
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


localhost:8000
Bongodev
python
iBOS
Devops
Cloud
Zam
git
Other favorites
RAPID COMPETITIONS
Home
All Competitions
Winners & Draw Results
Login
Basket
0
WHY CHOOSE US?
Best Odds
Limited tickets in each draw
Guaranteed Winners
Guaranteed winner even if tickets don't sell out
Live Draws
We show the number of entries in every draw





1 Apply Migrations

Run the following inside your EC2 container:

```
docker compose exec web bash
```

Then inside the container:

```
python manage.py makemigrations
python manage.py migrate
```

This will create all tables in your PostgreSQL database according to your models.

2️⃣ (Optional) Create a superuser

If you want to access the admin panel:

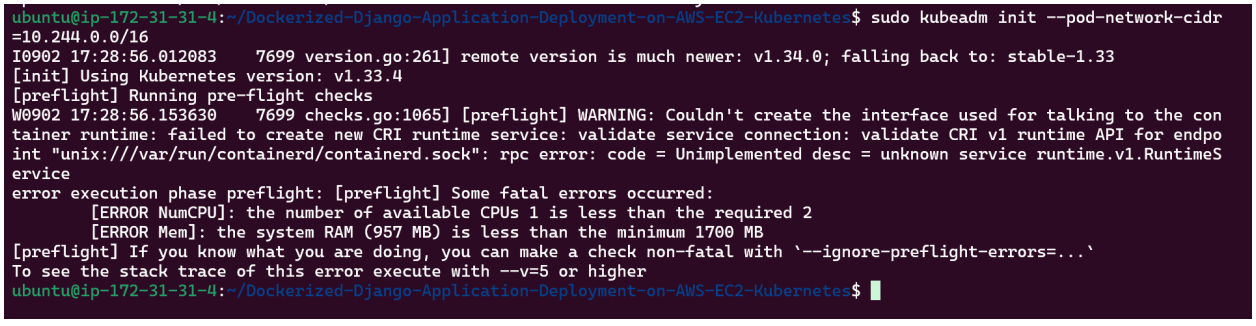
```
python manage.py createsuperuser
```

3️⃣ Collect static files (again if needed)

```
python manage.py collectstatic --noinput
```

4️⃣ Restart the container (if you want):

```
docker compose restart web
```



scp -i /mnt/d/Dockerized-Django-Application-Deployment-on-AWS-EC2-Kubernetes/1.pem -r /mnt/d/Dockerized-Django-Application-Deployment-on-AWS-EC2-Kubernetes ubuntu@34.207.196.215:~/Dockerized-Django-Application-Deployment-on-AWS-EC2-Kubernetes

sudo kubeadm init --pod-network-cidr=10.244.0.0/16

Install container runtime

```
# Install containerd
sudo apt update
sudo apt install -y containerd

# Create default config
sudo mkdir -p /etc/containerd
sudo containerd config default | sudo tee /etc/containerd/config.toml

# Restart containerd
sudo systemctl restart containerd
sudo systemctl enable containerd

# Enable IP forwarding
sudo sysctl net.ipv4.ip_forward=1
sudo sh -c 'echo "net.ipv4.ip_forward = 1" >> /etc/sysctl.conf'
sudo sysctl -p
```

sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=NumCPU,Mem

1️⃣ Configure kubectl for your user

Run these commands **as your regular user (`ubuntu`)**:

```
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Alternatively, if you want to use root:

```
export KUBECONFIG=/etc/kubernetes/admin.conf
```

2️⃣ Deploy a Pod network

Without a network, pods won't communicate. For example, using **Flannel**:

```
kubectl apply -f
https://raw.githubusercontent.com/flannel-io/flannel/master/Documentat
ion/kube-flannel.yml
```

- Wait a minute or two, then check:

```
kubectl get pods -n kube-system
```

You should see `coredns`, `kube-proxy`, and `flannel` pods running.

3️⃣ Verify the cluster

```
kubectl get nodes
kubectl get pods -A
```

- Your node should show `Ready` and have the `control-plane` label.

4️⃣ Deploy your Django app

1. **Namespace** (optional):

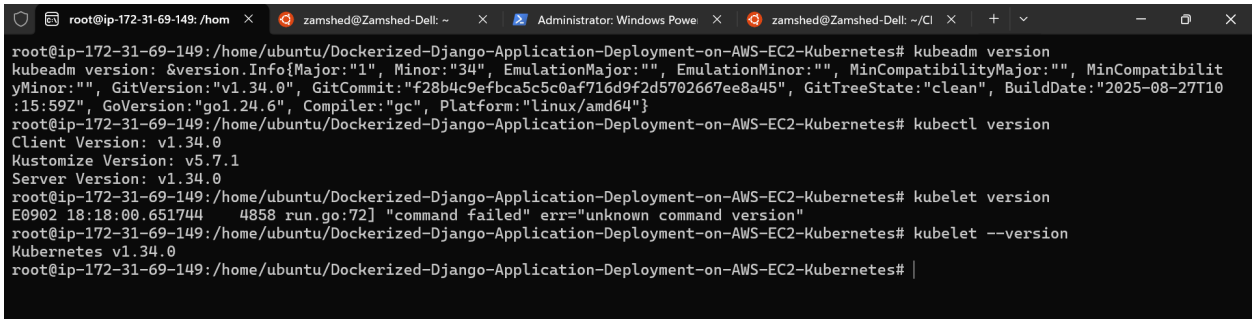
```
kubectl create namespace production
```

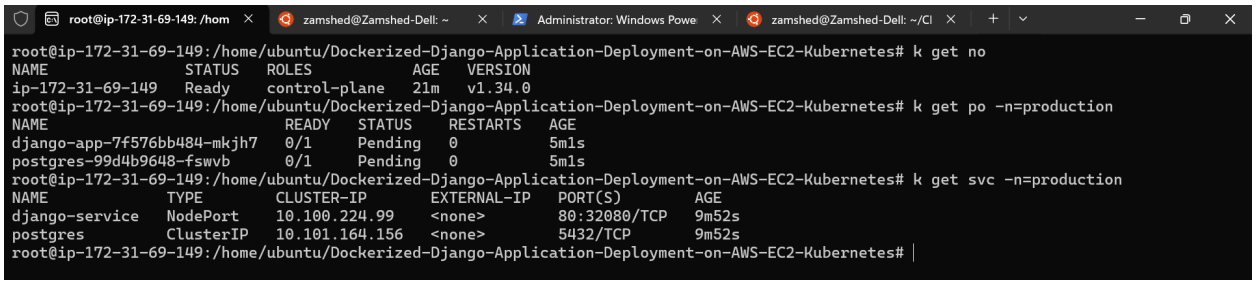
2. **Apply your Kubernetes manifests** (deployment + service):

```
kubectl apply -f k8s/deployment.yaml -n production
kubectl apply -f k8s/service.yaml -n production
```

3. **Verify pods**:

```
kubectl get pods -n production
kubectl get svc -n production
```





kubectl apply -f
https://raw.githubusercontent.com/flannel-io/flannel/master/Documentation/kube-flannel.yml