# Department Website Deployment Guide using Docker

This document provides a detailed step-by-step procedure for **backing up**, **migrating**, and **deploying** the Department Website using Docker.

---

## 1. Backup Live Database Collections

### Step 1.1: SSH into the Department Web Server

```
ssh iittp@10.21.24.115
```

### Step 1.2: Verify Running Containers

```
docker ps -a
```

Expected containers:

- `deptflow-web` (Frontend)
- `deptflow-backup` (Backup helper)
- `deptflow-mongodb` (MongoDB database)

### Step 1.3: Create Backup Directory

```
mkdir -p ~/mongo_backups
ls mongo_backups/
```

### Step 1.4: Run `mongodump` inside the MongoDB container

```
docker exec deptflow-mongodb mongodump --db cse --out /backup
```

### Step 1.5: Copy Backup to Host

```
docker cp deptflow-mongodb:/backup ~/mongo_backups/cse_backup
```

### Step 1.6: Verify Backup Files

```
ls mongo_backups/cse_backup/cse/
```

## 2. Backup Docker Images

### Step 2.1: Save Application Image

```
docker save -o deptflow2508.tar deptflow:latest
```

### Step 2.2: Save MongoDB Image

```
docker save -o mongo2508.tar mongo:6
```

## 3. Copy Backup Files to Remote Machine

### Step 3.1: Compress Backup

```
cd ~/mongo_backups
tar -czvf cse_backup.tar.gz cse_backup
```

### Step 3.2: Copy Backup via SCP

```
scp -r cse_backup.tar.gz gsr@10.23.66.147:/home/gsr/IITTP_Dept_Website
```

## 4. Restore Database on Host Machine

### Step 4.1: Navigate to Backup Directory

```
cd ~/mongo_backups
```

### Step 4.2: Extract Backup

```
tar -xzvf cse_backup.tar.gz -C ~/mongo_collections/
cd ~/mongo_collections/cse_backup/cse
```

### Step 4.3: Import into MongoDB

```
mongorestore --db cse .
```

### Step 4.4: Verification

```
mongosh
use cse
show collections
db.users.find().limit(5).pretty()
db.publications.find().limit(5).pretty()
```

---

## 5. Environment Configuration

### Step 5.1: Navigate to Website Directory

```
cd IITTP_Dept_Website
```

### Step 5.2: Edit `.env`

- Disable **local environment variables**
- Enable **production environment variables**

---

## 6. Export Latest Images

Create and run script:

```
./export-docker-images.sh
```

This will package the latest Docker images for deployment.

---

## 7. Deploy Client Directory

### Step 7.1: Compress Client Code

```
tar -czvf client.tar.gz client
```

### Step 7.2: Copy to Production Server

```
scp -r client.tar.gz iittp@10.21.24.115:/home/iittp/Downloads/
```

## 8. Import Latest Images on Production Server

### Step 8.1: Load Application Image

```
docker load < images/deptflow-latest.tar.gz
```

### Step 8.2: Load MongoDB Image

```
docker load < images/mongo-6.tar.gz
```

## 9. Start All Services on Production Server

### Step 9.1: Navigate to Deployment Directory

```
cd IITTP_Dept_Website
```

### Step 9.2: Start Services

```
docker-compose up -d
```

### Step 9.3: Verify Services

```
docker ps
```

Ensure containers for `deptflow-web` and `deptflow-mongodb` are running successfully.

## ✅ Deployment Completed

The Department Website is now live with the latest database backup and application images.

## Troubleshooting

- **Mongo container restarting**: Check logs

```
docker logs deptflow-mongodb
```

- **Database not restored**: Ensure correct path in `mongorestore`.
- **Web not accessible**: Verify port mapping in `docker-compose.yml`.

---