



Server Deployment Guide

Quick Deployment (5 minutes)

Server Information

- **Host:** 185.104.114.88
 - **User:** u2860854
 - **Password:** 7i0O2n9U7p
 - **Project Path:** ~/telegram-travel-csp72-bot
-



Option 1: Automated Deployment (Recommended)

Connect and Deploy

```
# Step 1: Connect to server
ssh u2860854@185.104.114.88

# Step 2: Navigate to project
cd ~/telegram-travel-csp72-bot

# Step 3: Run deployment script
./deploy.sh
```

That's it! The script will:

- Pull latest changes from GitHub
 - Stop the current bot
 - Rebuild Docker image
 - Start the bot
 - Verify it's running
 - Show logs
-

Option 2: Manual Deployment

If automated script doesn't work

```
# Step 1: Connect
ssh u2860854@185.104.114.88

# Step 2: Navigate to project
cd ~/telegram-travel-csp72-bot

# Step 3: Pull changes
git pull origin master

# Step 4: Restart bot
docker-compose down
docker-compose up -d

# Step 5: Check logs
docker-compose logs -f bot
```

Verification Commands

Check if bot is running

```
docker-compose ps
```

View logs

```
docker-compose logs -f bot
```

Check last 50 log lines

```
docker-compose logs --tail=50 bot
```

Restart bot


```
docker-compose restart bot
```

Full restart (rebuild)


```
docker-compose down
docker-compose build --no-cache
docker-compose up -d
```

Testing After Deployment



Test 1: Application Submission

1. Open bot: https://t.me/your_bot_name
2. Click “ Подать заявку”
3. Fill all fields
4. Submit application
5. **Expected:** Success message + email received

Test 2: Draft Saving

1. Start new application
2. Fill all fields
3. Click “ Сохранить черновик”
4. **Expected:** “Черновик сохранён!”

Test 3: Draft Loading

1. Click “ Мои черновики”
2. Click “ Продолжить” on any draft
3. **Expected:** Draft loads with all data

Test 4: Email Verification

- Check email: **csp-72@yandex.ru**
- Look for new application
- Open attached Excel file
- **Expected:** Clean, formatted Excel with all data



Troubleshooting

Bot not starting?

```
# Check Docker status
docker-compose ps

# View detailed logs
docker-compose logs bot

# Check if port is in use
netstat -tulpn | grep :8080
```

Git pull fails?

```
# Reset local changes
git reset --hard origin/master
git pull origin master
```

Database issues?

```
# Check database
docker-compose exec bot python -c "from bot.database import database; print('DB OK')"
```

```
# Restart database
docker-compose restart db
```

Excel generation fails?

```
# Test Excel generation
docker-compose exec bot python -c "
from bot.utils.excel_generator import generate_excel
data = {
    'sport_type': 'Test',
    'event_rank': 'Test',
    'country': 'Test',
    'city': 'Test',
    'participants': []
}
print(generate_excel(data))
"
```



What Was Fixed

Issue 1: Application Submission

- **Problem:** Applications failing with error message
- **Solution:** Simplified Excel generation (no more template dependencies)
- **Result:** Reliable application submission

Issue 2: Draft Management

- **Problem:** Couldn't load or continue drafts
- **Solution:** Full draft management system
- **Result:** Load, edit, delete drafts



Important Notes

1. **Always check logs** after deployment
2. **Test all features** before considering deployment complete
3. **Keep .env file** unchanged (contains sensitive credentials)
4. **Backup database** before major changes
5. **Monitor email delivery** in first 24 hours



Environment Variables (Already Configured)

Your `.env` file should contain:

```
TELEGRAM_BOT_TOKEN=8148484265:AAF...
TELEGRAM_ADMIN_IDS=317683765
EMAIL_TO_OVERRIDE=csp-72@yandex.ru
SMTP_HOST=smtp.mail.ru
SMTP_PORT=465
SMTP_USER=csp-72@mail.ru
SMTP_PASSWORD=your_password
DATABASE_URL=postgresql://...
```

⚠ **DO NOT share .env file or commit it to git**

📞 Need Help?

Check these first:

1. Bot logs: `docker-compose logs -f bot`
2. Container status: `docker-compose ps`
3. Disk space: `df -h`
4. Memory usage: `free -h`

Common Issues:

“Cannot connect to Docker daemon”

```
sudo service docker start
```

“Port already in use”

```
docker-compose down
# Wait 10 seconds
docker-compose up -d
```

“Database connection failed”

```
docker-compose restart db
sleep 5
docker-compose restart bot
```

✅ Success Indicators

After deployment, you should see:

- ✅ Bot container status: “Up”
- ✅ No error messages in logs
- ✅ Bot responds in Telegram
- ✅ Applications can be submitted
- ✅ Emails arrive with Excel attachments
- ✅ Drafts can be saved and loaded

You're Done!

If all tests pass, the bot is ready for use.

Deployment Time: ~5 minutes

Testing Time: ~10 minutes

Total Time: ~15 minutes

Questions? Check `FIXES_DEPLOYED.md` for detailed technical information.