🔄 MongoDB Atlas Auto-Sync Setup Guide

📋 Overview

This guide will help you set up **automatic daily synchronization** between two MongoDB Atlas databases using GitHub Actions.

Flow:

```
Production Atlas DB (Source)
        ↓
GitHub Actions Runner (Every 24h at Midnight)
        ↓
Testing Atlas DB (Target - Fresh Copy)
```

Key Features:

- ✅ Fully automatic (zero manual intervention)
- ✅ Runs every night at 12:00 AM UTC
- ✅ Drops testing database before restore (fresh copy)
- ✅ Free (GitHub Actions Free tier)
- ✅ Secure (encrypted secrets)
- ✅ No server dependency (Render/Compass not needed)

🚀 Quick Setup (5 Minutes)

Step 1: Prepare MongoDB Connection Strings

You have two Atlas databases:

Production (Source):

```
mongodb+srv://engineeringservicesdeveloper_db_user:EsdDB@123@cluster0.qyouk3t.mongodb.net/e-pharmacy?appName=Cluster0
```

Testing (Target):

```
mongodb+srv://testing_db_user:<db_password>@cluster0.rdy6wsv.mongodb.net/e-pharmacy?appName=Cluster0
```

⚠️ **IMPORTANT:** Replace `<db_password>` with your actual testing database password.

Step 2: Add GitHub Secrets

GitHub Secrets encrypt sensitive data like database credentials. Here's how to add them:

2.1 Navigate to Repository Settings

1. Go to your GitHub repository:

   ```
   https://github.com/Aayushajs/PHRMA-PRODUCTION-APP-BACKEND-MAIN
   ```

2. Click **Settings** (top menu bar)

3. In the left sidebar, click:

   ```
   Security → Secrets and variables → Actions
   ```

4. Click **"New repository secret"** button

2.2 Add Production Database Secret

Secret Name:

```
PRODUCTION_MONGO_URI
```

Secret Value:

```
mongodb+srv://engineeringservicesdeveloper_db_user:EsdDB@123@cluster0.qyouk3t.m
ongodb.net/e-pharmacy?appName=Cluster0
```

Click **"Add secret"**

2.3 Add Testing Database Secret

Secret Name:

```
TESTING_MONGO_URI
```

Secret Value:

```

```

```
mongodb+srv://testing_db_user:YOUR_ACTUAL_PASSWORD@cluster0.rdy6wsv.mongodb.net
/e-pharmacy?appName=Cluster0
```

⚠ Replace `YOUR_ACTUAL_PASSWORD` with the real password!

Click **"Add secret"**

☑ Verification

You should now see two secrets:

```
✓ PRODUCTION_MONGO_URI
✓ TESTING_MONGO_URI
```

Step 3: Commit & Push Workflow File

The workflow file is already created at:

```
.github/workflows/mongodb-atlas-sync.yml
```

Push it to GitHub:

```
git add .github/workflows/mongodb-atlas-sync.yml
git commit -m "Add MongoDB Atlas daily sync workflow"
git push origin main
```

Step 4: Verify Workflow Installation

1. Go to your GitHub repository
2. Click **Actions** tab (top menu)
3. You should see:

   ```
   MongoDB Atlas Daily Sync (Production → Testing)
   ```

4. The workflow will automatically run:
   - **Every night at 12:00 AM UTC** (scheduled)
   - **Or manually** (you can trigger it anytime)

🧪 Manual Test Run (Recommended)

Before waiting 24 hours, test the workflow immediately:

Option 1: GitHub UI (Easy)

1. Go to **Actions** tab
2. Click **"MongoDB Atlas Daily Sync"** workflow
3. Click **"Run workflow"** dropdown button (right side)
4. Select:
   - Branch: `main`
   - Drop database: `true`
5. Click green **"Run workflow"** button

Option 2: Command Line

```
gh workflow run mongodb-atlas-sync.yml
```

(Requires GitHub CLI installed)

📊 Monitor Workflow Execution

View Real-Time Logs

1. Go to **Actions** tab
2. Click the running workflow
3. Click **"sync-mongodb-databases"** job
4. Watch live logs for each step:

   ```
   ✅ Install MongoDB Tools
   ✅ Dump Production Database
   ✅ Restore to Testing Database
   ✅ Cleanup & Summary
   ```

Check Sync Status

Look for these success indicators:

```
✅ Production database dumped successfully!
✅ Testing database restored successfully!
🎉 Sync completed at 2026-02-02 00:00:15 UTC
```

⏰ Cron Schedule Configuration

The workflow runs at **12:00 AM UTC** by default.

Change Schedule Time

Edit `.github/workflows/mongodb-atlas-sync.yml`:

```
schedule:
  - cron: '0 0 * * *'  # Current: Midnight UTC
```

Common Timezones:

Your Local Time	Cron Expression	Description
12:00 AM IST	`30 18 * * *`	6:30 PM UTC (IST = UTC+5:30)
12:00 AM PST	`8 0 * * *`	8:00 AM UTC (PST = UTC-8)
02:00 AM UTC	`0 2 * * *`	2:00 AM UTC
Every 12 hours	`0 */12 * * *`	Midnight + Noon UTC

Cron Format:

```
┌───────────── minute (0 - 59)
│ ┌───────────── hour (0 - 23)
│ │ ┌───────────── day of month (1 - 31)
│ │ │ ┌───────────── month (1 - 12)
│ │ │ │ ┌───────────── day of week (0 - 6) (Sunday = 0)
│ │ │ │ │
* * * * *
```

🔒 Security Best Practices

☑ DO's

1. **Always use GitHub Secrets** for connection strings
2. **Never commit credentials** to code
3. **Use read-only users** for production dumps (if possible)
4. **Enable 2FA** on GitHub account
5. **Restrict workflow permissions** in repo settings

❌ DON'Ts

1. **Never hardcode passwords** in YAML files
2. **Don't expose secrets** in logs (GitHub automatically masks them)

3. **Avoid public repositories** with production credentials

🛠️ Troubleshooting

Issue 1: Workflow Not Running

Problem: Workflow doesn't appear in Actions tab

Solution:

```
# Verify file path
ls .github/workflows/mongodb-atlas-sync.yml

# Push to GitHub
git push origin main
```

Issue 2: Authentication Error

Problem: `MongoServerError: bad auth`

Solution:

1. Double-check GitHub Secrets values
2. Ensure no extra spaces in connection string
3. Verify database user has correct permissions:
   - Production: Read access
   - Testing: Read+Write access

Issue 3: mongodump/mongorestore Fails

Problem: `mongodump: command not found`

Solution:
The workflow automatically installs tools. If this fails:

1. Check MongoDB Tools version in workflow file
2. Update `MONGODB_TOOLS_VERSION` env variable

Issue 4: Timeout Error

Problem: `Error: The operation was canceled.`

Solution:

1. Increase `timeout-minutes` in workflow file
2. Reduce database size (if >1GB)
3. Use `--numParallelCollections=8` for faster dumps

📈 Advanced Configuration

Option 1: Slack Notifications

Get notified on sync failures:

1. Create Slack webhook: https://api.slack.com/messaging/webhooks
2. Add GitHub Secret: `SLACK_WEBHOOK_URL`
3. Uncomment notification step in workflow

Option 2: Exclude Collections

Skip specific collections during sync:

```
# In mongodump step
mongodump \
  --uri="${PROD_MONGO_URI}" \
  --db="${{ env.DATABASE_NAME }}" \
  --excludeCollection=logs \
  --excludeCollection=temp_data \
  --out="${{ env.DUMP_DIR }}"
```

Option 3: Backup Before Restore

Keep safety backups of testing database:

```
# Add before restore step
- name: Backup Testing DB (Safety)
  run: |
    mongodump \
      --uri="${{ secrets.TESTING_MONGO_URI }}" \
      --db="${{ env.DATABASE_NAME }}" \
      --out="./backup_$(date +%Y%m%d)"
```

📊 Resource Usage

GitHub Actions Free Tier Limits

Resource	Free Tier	This Workflow
Minutes/Month	2,000	~15 min/month
Storage	500 MB	0 MB (temp files)
Concurrent Jobs	20	1

Estimated Cost: $0/month (well within free tier)

MongoDB Atlas Free Tier (M0)

Specification	Limit	Notes
Storage	512 MB	mongodump works fine
RAM	Shared	No performance impact
Bandwidth	Unlimited	No transfer fees
Backups	Manual only	This workflow is your backup!

🎯 Success Checklist

Before considering setup complete:

- □ GitHub Secrets added correctly (PRODUCTION_MONGO_URI, TESTING_MONGO_URI)
- □ Workflow file pushed to repository
- □ Workflow appears in Actions tab
- □ Manual test run completed successfully
- □ Testing database contains production data
- □ Cron schedule matches your timezone preference
- □ Email notifications enabled (GitHub account settings)

📞 Support & Maintenance

View Workflow History

```
GitHub Repo → Actions Tab → MongoDB Atlas Daily Sync
```

Each run shows:

- Execution time
- Success/Failure status
- Detailed logs
- Resource usage

Update MongoDB Tools Version

When new versions release:

```
# In mongodb-atlas-sync.yml
env:
  MONGODB_TOOLS_VERSION: '100.10.0'  # Update this
```

Disable Automatic Sync

Comment out the schedule section:

```
# schedule:
#   - cron: '0 0 * * *'
```

Workflow will only run manually via GitHub UI.

🎉 You're All Set!

Your MongoDB Atlas databases will now sync automatically every 24 hours.

Next Steps:

1. Wait for midnight UTC (or run manually)
2. Check Actions tab for first run
3. Verify testing database has fresh data
4. Monitor logs for any issues

Questions?

- Check GitHub Actions logs for detailed error messages
- Review MongoDB Atlas connection settings
- Verify network access (IP whitelist) in Atlas

📚 Additional Resources

- GitHub Actions Documentation
- MongoDB Database Tools
- mongodump Reference
- mongorestore Reference
- Cron Expression Generator

Last Updated: February 2, 2026
Workflow Version: 1.0
MongoDB Tools Version: 100.9.4