GitHub Actions: Daily Backup & Deployment to AWS EC2 (Code and Environment Separated)

This GitHub Actions workflow handles **daily MySQL database backups** from an **AWS RDS** instance and **automated deployment** of the CSHub frontend and backend code to an EC2 instance using Docker. The code and Docker environments are maintained **separately** for modular management.

Workflow File

`.github/workflows/ CSHub-auto-deployment.yml`

Triggers

- **Scheduled**: Runs daily at `08:00` UTC (02:00 Central Time).
- **Manual**: Can also be triggered on demand using `workflow_dispatch`.

📦 Job 1: `dump_and_commit` — Backup Database

1. **Checkout Repository**
    a. Ensures the GitHub runner has access to the project files and previous backups.
2. **Configure Git User**
    a. Sets Git user credentials for automated commits.
3. **Install MySQL Client**
    a. Required to connect and export the MySQL database schema.
4. **Dump the RDS Database**
    a. Dumps the schema `cshub` to a `.sql` file using `mysqldump`.
    b. Output is timestamped and stored in the `db_backup/` directory.
5. **Clean Up Old Backups**
    a. Deletes backups older than 7 days automatically.
6. **Commit & Push Backups**
    a. Pushes the latest backup to the repo.

🚀 Job 2: `deploy-to-ec2` — Build & Deploy Code to EC2

Waits for the backup job to complete, then:

1. **Checkout Code**
    a. Pulls the latest frontend and backend source code.
2. **Setup Scala + sbt**
    a. Installs Scala and sbt (used to compile Play framework apps).
3. **Clean Ivy/Coursier Cache**
    a. Ensures fresh dependency resolution to avoid building conflicts.
4. **Compile & Package Apps**
    a. Runs `sbt clean compile stage dist` for both frontend and backend.
5. **Remove PID Files**
    a. Deletes stale `RUNNING_PID` files that may interfere with restarts.
6. **Rename and Upload ZIPs**
    a. Compresses compiled artifacts and uploads them to the EC2 instance via `scp`.
7. **SSH & Deploy via Docker**
    a. Logs into EC2 and deploys updated applications using “sudo apt-get update && sudo apt-get install -y unzip”
    b. Remove old files and unzips new frontend and backend package files in `/home/ubuntu/beichenh/scihub`.
    c. Stops and removes any existing Docker containers for the CSHub frontend and backend environments.
    d. Mounts the newly uploaded and extracted frontend/backend code directories into their respective Docker containers using pre-built CSHub environment images.
    e. Configures required environment variables, such as `PLAY_SECRET_KEY`, `DB_URL`, etc., to ensure proper runtime behavior.
    f. Cleans up unused Docker images to free up disk space on the EC2 instance.

🔐 Required Secrets

Secret Name	Description
`RDS_HOST`	Host address of the AWS RDS database
`DB_USERNAME`	RDS database username

DB_PASSWORD	RDS database password
DB_NAME	Name of the schema to be backed up
EC2_HOST	Public IP or domain of the EC2 instance
EC2_USER	SSH username for the EC2 instance
EC2_SSH_KEY	Private SSH key for connecting to the EC2 instance
DOCKER_USERNAME	Docker Hub username
DOCKER_PASSWORD	Docker Hub password
PLAY_FRONTEND_SECRET_KEY	Secret key for Play framework (frontend)
PLAY_BACKEND_SECRET_KEY	Secret key for Play framework (backend)

📌 Notes

- This setup assumes Docker, Java, and unzip are installed on the EC2 instance.
- Database dumps are lightweight and versioned via Git—ideal for lightweight RDS snapshot tracking.
- Secrets are managed securely using GitHub Secrets.
- Deployment is clean and repeatable using Docker and sbt-generated dist archives.