Sequence Diagram: ZIP Upload & Deploy (UC-08)

Participants: User, Browser, Project Page, POST /api/upload/{slug}, MinIO, POST /api/projects/{slug}/deployments, Build Queue, MongoDB

User → Project Page: Select "Upload ZIP" tab

User → Project Page: Choose ZIP file

Project Page → POST /api/upload/{slug}: POST multipart/form-data {file: archive.zip}

POST /api/upload/{slug} → POST /api/upload/{slug}: requireAuth()

POST /api/upload/{slug} → MongoDB: Find project by slug + ownership check

MongoDB --> POST /api/upload/{slug}: Project

POST /api/upload/{slug} → POST /api/upload/{slug}: Parse multipart form data

POST /api/upload/{slug} → POST /api/upload/{slug}: Validate file exists

POST /api/upload/{slug} → MinIO: Upload to uploads/{slug}/{filename}

MinIO --> POST /api/upload/{slug}: Upload successful

POST /api/upload/{slug} → MongoDB: Update project: zipFileName = filename

MongoDB --> POST /api/upload/{slug}: Updated

POST /api/upload/{slug} --> Project Page: 200 {fileName, path}

Project Page --> User: Show "Upload complete"

User → Project Page: Click "Deploy"

Project Page → POST /api/projects/{slug}/deployments: POST /api/projects/{slug}/deployments

POST /api/projects/{slug}/deployments → MongoDB: Create deployment sourceSnapshot.type = 'zip' sourceSnapshot.zipPath = path

MongoDB --> POST /api/projects/{slug}/deployments: Deployment created

POST /api/projects/{slug}/deployments → Build Queue: enqueueBuild(deploymentId)

Build Queue --> POST /api/projects/{slug}/deployments: Job enqueued

POST /api/projects/{slug}/deployments --> Project Page: 201 {deployment}

Note: Worker downloads ZIP from MinIO instead of git clone, extracts and runs build commands