Implement & Deploy — Phase II (LMS)

Document control

Version: 1.0

Date: 2025-08-28

Author: Sok Kim Thanh / Team

# 1. Scope & Audience

Tài liệu này hướng dẫn developer và DevOps triển khai LMS từ giai đoạn code → build → migrate → deploy → post-deploy, đồng bộ với SDD (kiến trúc, NFRs) và Runbook vận hành.

# 2. Quick reference (summary checklist)

- Merge to `main`. CI build and tests must pass.

- Backup production DB trước khi áp migration.

- Pull image / deploy (blue/green nếu có).

- Apply migrations: `dotnet ef database update --no-build --context LmsDbContext`.

- Verify `/health` và chạy smoke tests: login, create course, enroll, start quiz, upload file.

# 3. Architecture summary (ref SDD)

Layers: Presentation (Blazor) → Application (MediatR) → Domain → Infrastructure (EF Core, Identity, Blob Storage). Outbox + background processors xử lý job bất đồng bộ. Xem SDD để có sơ đồ chi tiết và các quyết định kiến trúc.

# 4. Repo & solution conventions

Solution: `Lms.sln` với projects: `Lms.Domain`, `Lms.Application`, `Lms.Infrastructure`, `Lms.Web` và các project test. Tuân thủ quy ước namespace và cấu trúc module trong Implement Phase.

# 5. Libraries & standard configuration

Thư viện chính: Serilog, MediatR, AutoMapper, FluentValidation, EF Core (SqlServer), Identity, Azure.Storage.Blobs, Swashbuckle. Luôn khoá phiên bản trong csproj theo Appendix D của SDD.

# 6. Environment variables / Secrets (minimum)

- ASPNETCORE\_ENVIRONMENT

- ConnectionStrings\_\_DefaultConnection

- BlobStorage\_\_ConnectionString

- SMTP\_\_Host, SMTP\_\_Port, SMTP\_\_User, SMTP\_\_Password

- Jwt\_\_Issuer, Jwt\_\_Key, Jwt\_\_ExpireMinutes

- DataProtection\_\_Keys (nếu multi-instance)

- Storage\_\_SASKey (nếu dùng signed urls)

- Admin\_\_InitialEmail (optional seed)

# 7. Database: migration & seed policy

Quy trình:

- Dev: `dotnet ef migrations add <Name>` trong môi trường dev; PR review migration code.

- Staging/Prod: CI job chạy `dotnet ef database update --no-build --context LmsDbContext`.

- Backup production DB BEFORE applying migrations.

- Seed policy: Roles (Admin, Instructor, Learner), admin user, sample course. Seed chỉ chạy tự động khi biến môi trường chỉ định để tránh ghi đè dữ liệu production.

# 8. Authentication (Identity + JWT)

Sử dụng ASP.NET Identity với IdentityDbContext<AppUser, AppRole, Guid>. API sử dụng JWT Bearer; Blazor Server có thể dùng cookie auth cho UI. Env vars: Jwt\_\_Issuer, Jwt\_\_Key, Jwt\_\_ExpireMinutes.

Sample Program.cs snippet (đăng ký Identity + JWT):

// Identity  
builder.Services.AddIdentity<AppUser, AppRole>()  
 .AddEntityFrameworkStores<LmsDbContext>()  
 .AddDefaultTokenProviders();  
  
// JWT  
var jwtKey = builder.Configuration["Jwt:Key"] ?? throw new InvalidOperationException("Missing Jwt:Key");  
var jwtIssuer = builder.Configuration["Jwt:Issuer"];  
  
builder.Services.AddAuthentication(options =>  
{  
 options.DefaultAuthenticateScheme = JwtBearerDefaults.AuthenticationScheme;  
 options.DefaultChallengeScheme = JwtBearerDefaults.AuthenticationScheme;  
})  
.AddJwtBearer(options =>  
{  
 options.RequireHttpsMetadata = true;  
 options.SaveToken = true;  
 options.TokenValidationParameters = new TokenValidationParameters  
 {  
 ValidateIssuer = true,  
 ValidIssuer = jwtIssuer,  
 ValidateAudience = false,  
 ValidateIssuerSigningKey = true,  
 IssuerSigningKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(jwtKey)),  
 ClockSkew = TimeSpan.FromMinutes(2)  
 };  
});

# 9. Logging & health checks

Sử dụng Serilog (structured JSON). Enrichers: traceId, correlationId, userId. Health endpoints: /health (readiness/liveness). Đặt alert cho: high 5xx rate, DB connection failures, Outbox backlog.

Sample Serilog appsettings.json fragment:

{  
 "Serilog": {  
 "Using": [ "Serilog.Sinks.Console", "Serilog.Sinks.File" ],  
 "MinimumLevel": { "Default": "Information", "Override": { "Microsoft": "Warning" } },  
 "Enrich": [ "FromLogContext", "WithMachineName", "WithThreadId" ],  
 "WriteTo": [  
 { "Name": "Console" },  
 { "Name": "File", "Args": { "path": "logs/log-.json", "rollingInterval": "Day", "formatter": "Serilog.Formatting.Json.JsonFormatter, Serilog" } }  
 ]  
 }  
}

# 10. CI/CD pipeline (high-level)

Stages đề xuất:

1. PR: build + unit tests + static analysis.

2. Merge → main: build + integration tests + produce artifact (container image).

3. Pre-deploy (staging): deploy to staging, run smoke tests, run migration.

4. Production: pull new image, apply migrations (nếu safe) hoặc use migration-runner job, start new instances, run smoke tests & health-checks.

5. Post-deploy: monitor logs 30 minutes.

Sample migration step in pipeline:

# Agent script  
dotnet ef database update --no-build --context LmsDbContext --connection "$SQL\_CONN"

# 11. Deployment runbook (step-by-step)

Pre-deploy:

- Đảm bảo CI passed trên main; thông báo stakeholders.

- Backup DB production.

Deploy:

1. Pull latest image.

2. Stop old service (hoặc blue/green).

3. Apply DB migrations: `dotnet ef database update --no-build --context LmsDbContext`.

4. Start new instances.

5. Verify /health OK.

6. Run smoke tests: login, create course, enroll, start quiz, upload file.

Rollback:

- Nếu lỗi nghiêm trọng: rollback về image trước đó và restore DB từ backup (nếu schema đã thay đổi).

- Luôn giữ migrations reversible hoặc có script rollback manual.

- Tạo incident ticket và thông báo nhóm.

Post-deploy:

- Xác nhận background worker (Outbox) đang xử lý.

- Quay vòng keys tạm thời nếu cần.

- Cập nhật release notes và theo dõi logs tối thiểu 30 phút.

# 12. Modules / Implementation notes (key modules)

Authentication: register Identity, JWT, refresh token management, lockout và rate-limiting.

Course / Quiz / Enrollment: đảm bảo repository implement theo Application Abstractions và Global Query Filter cho soft-delete.

Outbox & Events: ghi Domain Events vào Outbox cùng transaction; background processor publish và cần retry policy + dead-letter handling.

# 13. Appendices

Appendix A: sample Program.cs (Identity + JWT + Serilog)

// Example - main Program.cs (short)  
var builder = WebApplication.CreateBuilder(args);  
builder.Host.UseSerilog((ctx, cfg) => cfg.ReadFrom.Configuration(ctx.Configuration));  
builder.Services.AddDbContext<LmsDbContext>(...);  
builder.Services.AddIdentity<AppUser, AppRole>().AddEntityFrameworkStores<LmsDbContext>();  
// JWT snippet (see earlier)  
builder.Services.AddHealthChecks().AddDbContextCheck<LmsDbContext>();  
var app = builder.Build();  
app.UseSerilogRequestLogging();  
app.MapHealthChecks("/health");  
app.MapControllers();  
app.Run();

Appendix B: EF Core migration commands & DDL snippets (tham khảo SDD Appendix A)

Appendix C: Deployment Checklist (env vars, monitoring, rollback plan) - bản gốc được hợp nhất vào phần trên.

# Revision notes

Tài liệu này hợp nhất Implement Phase và Deployment Checklist, đồng bộ với SDD. Mọi thay đổi kiến trúc (schema, outbox, audit) phải được cập nhật trong SDD và phản ánh vào tài liệu này.