IkeaDocuScan V3 - Deployment Plan

Target Environment: Windows Server with IIS and SQL Server **Framework:** .NET 10.0 **Application Type:** ASP.NET Core with Blazor WebAssembly **Organization:** IKEA (ikea.com)

Table of Contents

Prerequisites

Server Requirements

- Windows Server 2019 or later
- IIS 10.0 or later with ASP.NET Core Hosting Bundle
- SQL Server 2017 or later
- .NET 10.0 Runtime and Hosting Bundle installed

Required IIS Features

- Windows Authentication
- Application Development → ASP.NET 4.x (for compatibility)
- Application Development → WebSocket Protocol (for SignalR)
- Security → Request Filtering

Required Software on Deployment Machine

- Visual Studio 2022 (v17.8 or later)
- SQL Server Management Studio (SSMS)
- Access to target server (RDP or file share)

Access Requirements

- SQL Server sysadmin or db_owner role
- IIS Administrator permissions
- File system write permissions on deployment directory
- Active Directory read access (for Windows Authentication)

Pre-Deployment Checklist

- [ ] Backup current production database (if upgrading)
- [ ] Backup current appsettings.Local.json (if exists)

- [ ] Backup current secrets.encrypted.json (if exists)
- [ ] Document current version number
- [ ] Verify .NET 10.0 Runtime installed on target server
- [ ] Verify ASP.NET Core Hosting Bundle 10.0 installed
- [ ] Schedule maintenance window (if required)
- [ ] Notify users of deployment (if downtime expected)

Version Management

Understanding the Version System

The application uses .NET's built-in versioning with two components:

In `IkeaDocuScan-Web.csproj`:

```
<VersionPrefix>3.0.*</VersionPrefix>
<VersionSuffix>beta.10Nov25</VersionSuffix>
```

Version Format: `{VersionPrefix}.{Auto-Build}.{Auto-Revision}-{VersionSuffix}` **Example:** `3.0.1234.5678-beta.10Nov25`

Setting the Version for Release

Step 1: Open `IkeaDocuScan-Web.csproj` in Visual Studio

Step 2: Update version properties:

```
<!-- For Production Release -->
<VersionPrefix>3.1.0</VersionPrefix>
<VersionSuffix></VersionSuffix>

<!-- For Beta/RC Release -->
<VersionPrefix>3.1.0</VersionPrefix>
<VersionSuffix>rc1</VersionSuffix>

<!-- For Hotfix -->
<VersionPrefix>3.0.1</VersionPrefix>
<VersionSuffix></VersionSuffix>
```

Step 3: Save the file

Step 4: Rebuild solution to apply new version

Version Visibility: The version is displayed in: - Assembly metadata - Application logs - Can be added to UI footer (future enhancement)

Publishing from Visual Studio

Step 1: Clean and Rebuild Solution

1. In Visual Studio, right-click the Solution in Solution Explorer
2. Select **Clean Solution**
3. Wait for completion
4. Right-click the Solution again
5. Select **Rebuild Solution**
6. Verify no build errors in Output window

Step 2: Publish the Application

1. Right-click **IkeaDocuScan-Web** project (not the solution)
2. Select **Publish…**

Step 3: Create or Select Publish Profile

For First-Time Deployment: 1. Click **New** to create a new publish profile 2. Select **Folder** as target 3. Click **Next** 4. Choose folder location (e.g., `C:\Publish\IkeaDocuScan`) 5. Click **Finish** 6. Profile will be saved in `Properties\PublishProfiles\`

For Subsequent Deployments: 1. Select existing publish profile from dropdown 2. Click **Show all settings** to verify configuration

Step 4: Configure Publish Settings

Click **Show all settings** and verify:

Setting	Value
Configuration	Release
Target Framework	net9.0
Deployment Mode	Framework-dependent
Target Runtime	Portable
File Publish Options	☑ Delete existing files prior to publish
	☑ Exclude files from App_Data folder

Important: Use **Framework-dependent** deployment (not self-contained) to reduce publish size and rely on server's .NET runtime.

Step 5: Publish

1. Click **Publish** button
2. Monitor Output window for progress
3. Verify success message: "Publish succeeded"
4. Note the publish folder path

Step 6: Verify Published Files

Navigate to publish folder and verify these key files exist:

```
IkeaDocuScan-Web.dll
IkeaDocuScan-Web.deps.json
IkeaDocuScan-Web.runtimeconfig.json
appsettings.json
appsettings.Production.json
web.config
wwwroot/
  _framework/
```

Do NOT copy these files to server (will be created manually): - `appsettings.Local.json` - `secrets.encrypted.json`

Step 7: Include Database Migration Scripts

IMPORTANT: The database migration scripts must be included in the deployment package.

1. Navigate to solution root directory: `IkeaDocuScanV3\`
2. Verify the `DbMigration\db-scripts\` folder exists and contains all SQL scripts
3. Copy the entire folder to the publish directory:

```
xcopy /E /I "DbMigration\db-scripts" "C:\Publish\IkeaDocuScan\DbMigrati
```

4. **Verify ALL SQL scripts are present in the correct order:**

   The scripts must be executed in this exact sequence:

   ```
   DbMigration\db-scripts\
     ├── 00_Create_Database_And_User.sql
     ├── 00A_Restore_And_Migrate_Schema.sql
     ├── 02_Migrate_FK_Data.sql
     ├── 03_Finalize_FK_Constraints.sql
     ├── 04_Create_DocuScanUser_Table.sql
     ├── 05_Migrate_Users_To_DocuScanUser.sql
     ├── 06_Add_FK_Constraint_UserPermissions.sql
     ├── 07_Remove_AccountName_From_UserPermissions.sql
   ```

5. Create a checklist to verify all scripts are present:

   ```
   [ ] Script 1:  00_Create_Database_And_User.sql
   [ ] Script 2:  00A_Restore_And_Migrate_Schema.sql
   [ ] Script 4:  02_Migrate_FK_Data.sql
   [ ] Script 5:  03_Finalize_FK_Constraints.sql
   [ ] Script 6:  04_Create_DocuScanUser_Table.sql
   [ ] Script 7:  05_Migrate_Users_To_DocuScanUser.sql
   [ ] Script 8:  06_Add_FK_Constraint_UserPermissions.sql
   [ ] Script 9:  07_Remove_AccountName_From_UserPermissions.sql
   ```

Action Required Before Deployment: - Deployer must verify all 8 scripts are present before creating deployment ZIP

These scripts will be executed manually in SQL Server Management Studio during database setup.

Step 8: Copy ConfigEncryptionTool

1. Navigate to ConfigEncryptionTool publish output:

   ```
   cd ConfigEncryptionTool\bin\Release\net10.0
   ```

2. Copy ConfigEncryptionTool.exe and dependencies to deployment package:

   ```
   xcopy /E /I ConfigEncryptionTool.exe "C:\Publish\IkeaDocuScan\Tools\Cor
   xcopy /E /I *.dll "C:\Publish\IkeaDocuScan\Tools\ConfigEncryptionTool\"
   ```

Step 9: Create Deployment ZIP File

1. Navigate to publish folder parent directory:

   ```
   cd C:\Publish
   ```

2. Create ZIP file with timestamp:

   ```
   $timestamp = Get-Date -Format "yyyyMMdd_HHmmss"
   $zipName = "IkeaDocuScan_v3_$timestamp.zip"
   Compress-Archive -Path "IkeaDocuScan\*" -DestinationPath $zipName
   ```

3. Verify ZIP file contents:

   - Application DLLs and files
   - DbMigration-scripts*.sql
   - Tools*

4. Transfer ZIP file to target server using approved file transfer method

---

Database Setup

IMPORTANT: The production database will contain existing data from the current production system. The database upgrade is performed through SQL migration scripts, **NOT** Entity Framework migrations.

Step 1: Restore Production Database Backup

For Production/Dev/Test Deployment:

1. Obtain the latest backup of the current production IkeaDocuScan database (for dev/test could be an export that excludes tables Document and DocumentFile)
2. Open SQL Server Management Studio (SSMS)
3. Connect to the production SQL Server instance
4. Run the 8 dll scripts in DbMigration-scripts

Step 2: Execute Database Migration Scripts

CRITICAL: The migration scripts must be executed in the exact order specified below using SQL Server Management Studio.

DO NOT USE Entity Framework Migrations - This deployment uses custom SQL scripts only.

Script Location: `C:\inetpub\wwwroot\IkeaDocuScan\DbMigration\db-scripts\` (extracted from deployment ZIP)

Execution Order:

The scripts are prefixed with numbers to indicate execution order. Execute ALL 8 scripts sequentially:

Order	Script Name	Purpose
1	`00_Create_Database_And_User.sql`	Creates database and application user
2	`00A_Restore_And_Migrate_Schema.sql`	Restores schema from backup
3	`02_Migrate_FK_Data.sql`	Migrates data for foreign keys
4	`03_Finalize_FK_Constraints.sql`	Creates foreign key constraints
5	`04_Create_DocuScanUser_Table.sql`	Creates DocuScanUser table
6	`05_Migrate_Users_To_DocuScanUser.sql`	Migrates user data
7	`06_Add_FK_Constraint_UserPermissions.sql`	Adds FK to UserPermissions
8	`07_Remove_AccountName_From_UserPermissions.sql`	Removes obsolete column

Execution Steps:

1. Open SQL Server Management Studio (SSMS)
2. Connect to SQL Server instance
3. Ensure connected to **IkeaDocuScan** database (dropdown at top)
4. Navigate to script folder: `C:\inetpub\wwwroot\IkeaDocuScan\DbMigration\db-scripts\`
5. **Execute each script in order (1-8 as listed in table above):**

   For each script:

a. Click **File → Open → File**

b. Select the script file (in correct order)

c. **Verify** database dropdown shows: **IkeaDocuScan**

d. **Review** script contents briefly (do NOT modify)

e. Click **Execute** (F5) or press F5

f. **Wait** for completion - watch Messages tab

g. **Verify** success message: "Command(s) completed successfully"

h. **Check** Messages tab for:

- Row counts affected
- Any warnings (yellow)
- Any errors (red - STOP if errors occur)

i. **Document** completion:

```
Script: [name]
Execution Time: [duration]
Rows Affected: [count]
Warnings: [none/list]
Status: SUCCESS / FAILED
```

j. If **SUCCESS**, proceed to next script

k. If **FAILED**, STOP and contact development team immediately

6. **After ALL scripts complete successfully:**

   Verify database state:

```sql
-- Verify migration completed
SELECT @@SERVERNAME AS ServerName,
       DB_NAME() AS DatabaseName,
       GETDATE() AS MigrationCompletedAt;

-- Check table count
SELECT COUNT(*) AS TableCount
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_TYPE = 'BASE TABLE';

-- Sample data check
SELECT COUNT(*) AS DocumentCount FROM Document;
SELECT COUNT(*) AS DocumentTypeCount FROM DocumentType;
SELECT COUNT(*) AS UserCount FROM DocuScanUser;
```

   Expected results (approximate - actual counts may vary):

   - TableCount: 10-15 tables
   - DocumentCount: [varies based on production data]
   - DocumentTypeCount: 5-20 types
   - UserCount: [varies based on production data]

Important Notes:

- □ **Do NOT skip any scripts**
- □ **Do NOT change the execution order**
- □ **Do NOT modify script contents**
- □ **Execute ONE script at a time**
- □□ If a script fails:
  - Note the exact error message
  - Note which script failed
  - Note how far through the script execution progressed
  - STOP immediately - do NOT continue

    - Contact development team with error details
- □ Keep a detailed log of:
    - Each script name
    - Execution start/end time
    - Rows affected
    - Any warnings or messages
    - Final status (SUCCESS/FAILED)

Script Execution Log Template:

```
Database Migration Execution Log
Date: ____________________
Database Server: ____________________
Database Name: IkeaDocuScan
Executed By: ____________________

Script Execution Details:
------------------------

Script 1: 00_Create_Database_And_User.sql
Start Time: __________
End Time: __________
Duration: __________
Rows Affected: __________
Warnings: __________
Status: [ ] SUCCESS  [ ] FAILED
Notes: _____________________________________________

Script 2: 00A_Restore_And_Migrate_Schema.sql
Start Time: __________
End Time: __________
Duration: __________
Rows Affected: __________
Warnings: __________
Status: [ ] SUCCESS  [ ] FAILED
Notes: _____________________________________________

Script 3: 02_Migrate_FK_Data.sql
Start Time: __________
End Time: __________
Duration: __________
Rows Affected: __________
Warnings: __________
Status: [ ] SUCCESS  [ ] FAILED
Notes: _____________________________________________

Script 4: 03_Finalize_FK_Constraints.sql
Start Time: __________
End Time: __________
Duration: __________
Rows Affected: __________
Warnings: __________
Status: [ ] SUCCESS  [ ] FAILED
Notes: _____________________________________________

Script 5: 04_Create_DocuScanUser_Table.sql
Start Time: __________
End Time: __________
Duration: __________
Rows Affected: __________
Warnings: __________
Status: [ ] SUCCESS  [ ] FAILED
Notes: _____________________________________________

Script 6: 05_Migrate_Users_To_DocuScanUser.sql
Start Time: __________
End Time: __________
Duration: __________
Rows Affected: __________
Warnings: __________
Status: [ ] SUCCESS  [ ] FAILED
Notes: _____________________________________________
```

```
Script 7: 06_Add_FK_Constraint_UserPermissions.sql
Start Time: __________
End Time: __________
Duration: __________
Rows Affected: __________
Warnings: __________
Status: [ ] SUCCESS  [ ] FAILED
Notes: _____________________________________________

Script 8: 07_Remove_AccountName_From_UserPermissions.sql
Start Time: __________
End Time: __________
Duration: __________
Rows Affected: __________
Warnings: __________
Status: [ ] SUCCESS  [ ] FAILED
Notes: _____________________________________________


OVERALL STATUS: [ ] ALL SCRIPTS SUCCESSFUL  [ ] MIGRATION FAILED

Signature: ____________________ Date: __________
```

Step 3: Verify Database User (Created by Scripts)

The migration scripts automatically create the application database user.

Verify the user was created:

```sql
USE IkeaDocuScan;
GO

-- Check if login exists
SELECT name, type_desc, create_date
FROM sys.server_principals
WHERE name = 'docuscanch';

-- Check if database user exists
SELECT name, type_desc, create_date
FROM sys.database_principals
WHERE name = 'docuscanch';

-- Verify user permissions
EXEC sp_helpuser 'docuscanch';
```

Expected Result: - Login `docuscanch` exists at server level - User `docuscanch` exists in IkeaDocuScan database - User has db_datareader, db_datawriter roles

If user was NOT created by scripts:

Contact development team immediately. The migration scripts should handle user creation.

Step 4: Test Database Connection

Test connectivity with the application user:

```sql
-- Test as docuscanch user
EXECUTE AS USER = 'docuscanch';

-- Should succeed (read access)
SELECT COUNT(*) FROM Document;
SELECT COUNT(*) FROM DocumentType;

-- Revert to admin
REVERT;
```

Expected: Queries execute successfully without errors.

Step 5: Verify Database Configuration Tables

To Be Defined: Some application configuration settings are stored in database tables. The exact tables and initial configuration will be documented by the development team.

Placeholder for configuration verification:

```
-- Configuration tables to be verified
-- TODO: Add specific tables and validation queries
-- Example: SELECT * FROM SystemConfiguration;
```

Action Required: Development team to provide: - List of configuration tables - Required configuration records - Validation queries - Seed data scripts (if needed)

IIS Configuration

Step 1: Extract Deployment Package

1. Copy the deployment ZIP file to the server (e.g., `C:\Temp\`)
2. Extract ZIP file to deployment directory:

   ```
   Expand-Archive -Path "C:\Temp\IkeaDocuScan_v3_*.zip" -DestinationPath "
   ```

   OR using Windows Explorer:

   - Right-click ZIP file → **Extract All**
   - Target: `C:\inetpub\wwwroot\IkeaDocuScan`
3. Verify extracted contents:

   ```
   C:\inetpub\wwwroot\IkeaDocuScan\
     ├── IkeaDocuScan-Web.dll
     ├── web.config
     ├── wwwroot\
     ├── DbMigration\
     │   └── db-scripts\
     └── Tools\
         └── ConfigEncryptionTool\
   ```

Step 2: Create Application Pool

1. Open IIS Manager
2. Expand server node
3. Right-click **Application Pools** → **Add Application Pool**

Setting	Value
Name	IkeaDocuScan
.NET CLR version	No Managed Code
Managed pipeline mode	Integrated
Start application pool immediately	☑

4. Click **OK**

Step 3: Configure Application Pool

1. Select **IkeaDocuScan** application pool
2. Click **Advanced Settings** in right panel

Setting	Value	Notes
General → .NET CLR Version	No Managed Code	Required for .NET Core
General → Managed Pipeline Mode	Integrated	
General → Start Mode	AlwaysRunning	Faster startup
Process Model → Identity	ApplicationPoolIdentity	Most secure option
Process Model → Idle Time-out (minutes)	0	Disable idle timeout
Process Model → Load User Profile	True	Required for DPAPI encryption
Recycling → Regular Time Interval	1740 (29 hours)	Avoid daily recycling

3. Click **OK**

Important: If using Windows Authentication with Active Directory, you may need to use a domain service account instead of ApplicationPoolIdentity because the account must be known to the domain: - Identity: **Custom Account** → `DOMAIN\ServiceAccount`

Step 4: Create IIS Website

Option A: Create New Website

1. Right-click **Sites** → **Add Website**

Setting	Value
Site name	IkeaDocuScan
Application pool	IkeaDocuScan
Physical path	C:
Binding → Type	https
Binding → IP address	All Unassigned
Binding → Port	443
Binding → Host name	docuscan.company.com
Binding → SSL certificate	(Select appropriate certificate)

2. Click **OK**

Option B: Create as Application under Existing Site

1. Expand **Sites** → **Default Web Site**
2. Right-click **Default Web Site** → **Add Application**

Setting	Value
Alias	docuscan
Application pool	IkeaDocuScan
Physical path	C:

3. Click **OK**

Step 5: Configure Website Settings

1. Select the IkeaDocuScan site/application
2. Double-click **Authentication**

Authentication Type	Status
Anonymous Authentication	Disabled
Windows Authentication	**Enabled**

3. Click **Windows Authentication** → **Advanced Settings**
   - Extended Protection: **Accept**
   - Enable Kernel-mode authentication: ☑

Step 6: Configure Application Settings

1. Select IkeaDocuScan site/application
2. Click **Configuration Editor** in main panel
3. Select **system.webServer/aspNetCore** section
4. Verify settings:

Setting	Value
processPath	dotnet
arguments	.-Web.dll
stdoutLogEnabled	true
stdoutLogFile	.
hostingModel	inprocess

5. Click **Apply**

Step 7: Configure WebSocket Protocol (for SignalR)

1. Select IkeaDocuScan site/application
2. Double-click **WebSocket Protocol** feature
3. If not installed:
   - Open **Server Manager**
   - **Manage** → **Add Roles and Features**
   - Navigate to **Web Server (IIS)** → **Web Server** → **Application Development**
   - Check **WebSocket Protocol**
   - Install

Application Configuration

Step 1: Use ConfigEncryptionTool to Create Encrypted Configuration

IMPORTANT: The connection string must be encrypted using Windows DPAPI. Use the provided ConfigEncryptionTool.

Why Encrypt: The database connection string contains the `docuscanch` user password and should not be stored in plain text.

Running the Tool:

1. Navigate to the tools directory:

   ```
   cd C:\inetpub\wwwroot\IkeaDocuScan\Tools\ConfigEncryptionTool
   ```

2. Run as the IIS Application Pool identity:

   ```
   runas /user:"IIS APPPOOL\IkeaDocuScan" "cmd.exe"
   ```

3. In the new command prompt window:

   ```
   cd C:\inetpub\wwwroot\IkeaDocuScan\Tools\ConfigEncryptionTool
   ConfigEncryptionTool.exe
   ```

4. Follow the interactive prompts:

```
DATABASE CONFIGURATION:
SQL Server: [Enter SQL Server name, e.g., PROD-SQL-01]
Database Name: IkeaDocuScan
Use Windows Authentication? n
Username: docuscanch
Password: [Enter docuscanch password - characters will be masked]

APPLICATION CONFIGURATION:
Scanned Files Path: \\fileserver\ScannedDocuments
```

5. Tool will create `secrets.encrypted.json` with encrypted connection string
6. Verify encryption test succeeds
7. Copy `secrets.encrypted.json` to application root:

```
copy secrets.encrypted.json C:\inetpub\wwwroot\IkeaDocuScan\
```

Output File Format:

```
{
  "ConnectionStrings": {
    "DefaultConnection": "[ENCRYPTED_VALUE_USING_DPAPI]"
  },
  "IkeaDocuScan": {
    "ScannedFilesPath": "\\\\fileserver\\ScannedDocuments"
  }
}
```

Security Notes: - The encrypted file can ONLY be decrypted on this machine with the same user account - Backup this file as part of server backups - Do NOT commit to source control

Step 2: Create appsettings.Local.json

Create `appsettings.Local.json` in `C:\inetpub\wwwroot\IkeaDocuScan\` with IKEA-specific settings:

```
{
  "IkeaDocuScan": {
    "ContactEmail": "docuscan-support@ikea.com",

    "DomainName": "ikea.com",

    "UserEmail": {
      "LDAPRoot": "LDAP://DC=ikea,DC=com",
      "LDAPFilter": "(sAMAccountName={0})"
    },

    "EmailGroups": {
      "LDAPRoot": "LDAP://OU=Ikea,OU=Collab,DC=ikea,DC=com",
      "LDAPFilter": "(name=*Reminder*)"
    },

    "ADGroupReader": "IKEA\\UG-DocScanningReaders-CG@WAL-FIN-CH-GEL",
    "ADGroupPublisher": "IKEA\\UG-DocScanningPublishers-CG@WAL-FIN-CH-GEL"
    "ADGroupSuperUser": "IKEA\\UG-DocScanningSuperUsers-CG@WAL-FIN-CH-GEL"
  },

  "Email": {
    "SmtpHost": "smtp-gw.ikea.com",
    "SmtpPort": 25,
    "UseSsl": false,
    "SmtpUsername": "",
    "SmtpPassword": "",
    "FromAddress": "noreply-docuscan@ikea.com",
    "FromDisplayName": "IKEA DocuScan System",
    "AdminEmail": "docuscan-admins@ikea.com",
    "ApplicationUrl": "https://docuscan.ikea.com"
  },

  "ExcelExport": {
    "ApplicationUrl": "https://docuscan.ikea.com"
  },

  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning",
      "Microsoft.EntityFrameworkCore": "Warning"
    }
  }
}
```

Configuration Priority (highest to lowest): 1. Environment Variables (IIS App Pool) 2. `secrets.encrypted.json` ← **Connection string (encrypted)** 3. `appsettings.Local.json` ← **Server-specific settings** 4. `appsettings.Production.json` ← **Deployed with app** 5. `appsettings.json` ← **Deployed with app (defaults)**

Configuration Sections Explained:

Section	Purpose	Notes
DomainName	IKEA Active Directory domain	Used for user authentication
UserEmail.LDAPRoot	LDAP path for user email lookup	Retrieves logged-in user's email
UserEmail.LDAPFilter	LDAP filter for user search	{0} = sAMAccountName
EmailGroups.LDAPRoot	LDAP path for email groups	Used for action reminders
EmailGroups.LDAPFilter	Filter for reminder groups	Finds groups with "Reminder" in name
ADGroupReader	AD group for read-only access	Full IKEA group path
ADGroupPublisher	AD group for read/write access	Full IKEA group path
ADGroupSuperUser	AD group for admin access	Full IKEA group path
SmtpHost	IKEA SMTP gateway	smtp-gw.ikea.com
SmtpPort	SMTP port	25 (no authentication)
SmtpUsername	SMTP user (not used)	Leave empty
SmtpPassword	SMTP password (not used)	Leave empty

Step 3: Verify Configuration File Permissions

Ensure configuration files have proper permissions:

```
icacls "C:\inetpub\wwwroot\IkeaDocuScan\appsettings.Local.json" /grant "II
icacls "C:\inetpub\wwwroot\IkeaDocuScan\secrets.encrypted.json" /grant "II
```

Step 4: Configure Scanned Files Path

1. Verify the network share or local path exists:

   ```
   dir \\fileserver\ScannedDocuments
   ```

2. Test access as App Pool identity:

   ```
   runas /user:"IIS APPPOOL\IkeaDocuScan" "cmd.exe"
   dir \\fileserver\ScannedDocuments
   ```

3. If access denied, grant permissions on the file server

Step 5: Review and Adjust Settings

Review these settings in `appsettings.Local.json`:

Setting	Description	Production Value
IkeaDocuScan:MaxFileSizeBytes	Max upload size	52428800 (50MB)
DocumentSearch:MaxResults	Max search results	1000
ExcelExport:MaximumRowCount	Max Excel rows	50000
ExcelExport:WarningRowCount	Warning threshold	10000
Logging:LogLevel:Default	Logging level	Information
Email:EnableEmailNotifications	Enable emails	true

File Permissions

Step 1: Set Application Directory Permissions

```
cd C:\inetpub\wwwroot\IkeaDocuScan

REM Grant Read & Execute to App Pool
icacls . /grant "IIS APPPOOL\IkeaDocuScan:(OI)(CI)(RX)"

REM Grant Write to logs directory
icacls logs /grant "IIS APPPOOL\IkeaDocuScan:(OI)(CI)(M)"
```

Step 2: Create Logs Directory

```
cd C:\inetpub\wwwroot\IkeaDocuScan
mkdir logs

icacls logs /grant "IIS APPPOOL\IkeaDocuScan:(OI)(CI)(M)"
```

Step 3: Set Scanned Files Permissions

For Local Path:

```
icacls "D:\ScannedDocuments" /grant "IIS APPPOOL\IkeaDocuScan:(OI)(CI)(RX)
```

For Network Share:

On the file server: 1. Share Permissions: Grant **Read** to `IIS APPPOOL\IkeaDocuScan` (or domain service account) 2. NTFS Permissions: Grant **Read & Execute** to the same account

Test Access:

```
runas /user:"IIS APPPOOL\IkeaDocuScan" "cmd.exe"
dir \\fileserver\ScannedDocuments
```

Step 4: Restrict Configuration File Access

```
cd C:\inetpub\wwwroot\IkeaDocuScan

REM Remove inherited permissions
icacls appsettings.Local.json /inheritance:r

REM Grant specific access
icacls appsettings.Local.json /grant "Administrators:(F)"
icacls appsettings.Local.json /grant "IIS APPPOOL\IkeaDocuScan:(R)"

REM Same for secrets file if it exists
icacls secrets.encrypted.json /inheritance:r
icacls secrets.encrypted.json /grant "Administrators:(F)"
icacls secrets.encrypted.json /grant "IIS APPPOOL\IkeaDocuScan:(R)"
```

Windows Authentication Setup

Step 1: Verify IKEA Active Directory Groups

Ensure the following IKEA AD groups exist and users are properly assigned:

AD Group	Purpose	Access Level
UG-DocScanningReaders-CG@WAL-FIN-CH-GEL	View documents	Read-only
UG-DocScanningPublishers-CG@WAL-FIN-CH-GEL	Create/edit documents	Read/Write
UG-DocScanningSuperUsers-CG@WAL-FIN-CH-GEL	Full admin access	SuperUser

Verify Group Membership:

Use Active Directory Users and Computers or PowerShell:

```
# Check if groups exist
Get-ADGroup -Filter "Name -like '*DocScanning*'" | Select-Object Name, Dis

# Check members of a group
Get-ADGroupMember -Identity "UG-DocScanningReaders-CG@WAL-FIN-CH-GEL"
```

Step 2: Verify LDAP Configuration

The LDAP settings are configured in `appsettings.Local.json` (Step 2 of Application Configuration).

Verify LDAP connectivity:

```
# Test LDAP connection
$domain = "ikea.com"
$ldapPath = "LDAP://DC=ikea,DC=com"

$searcher = New-Object System.DirectoryServices.DirectorySearcher
$searcher.SearchRoot = New-Object System.DirectoryServices.DirectoryEntry(
$searcher.Filter = "(objectClass=user)"
$searcher.PropertiesToLoad.Add("sAMAccountName") | Out-Null
$result = $searcher.FindOne()

if ($result) {
    Write-Host "LDAP connection successful" -ForegroundColor Green
} else {
    Write-Host "LDAP connection failed" -ForegroundColor Red
}
```

Step 3: Test Windows Authentication

1. Browse to application URL from domain-joined machine
2. Should automatically authenticate with current user
3. If prompted for credentials, check:
   - IIS Windows Authentication is enabled
   - Browser is configured for Integrated Windows Authentication
   - Site is in Intranet zone (IE/Edge)

Step 4: Grant Initial Admin Access

To Be Defined: The process for granting initial SuperUser access to the first administrator will be defined by the development team based on the final user permissions structure.

Expected Steps: 1. Identify initial admin user(s) from IKEA 2. Add user(s) to `UG-DocScanningSuperUsers-CG@WAL-FIN-CH-GEL` AD group 3. Verify user appears in application with SuperUser access after first login

If database seeding is required, development team will provide script.

Post-Deployment Verification

Step 1: Check Application Pool Status

1. Open IIS Manager
2. Navigate to **Application Pools**
3. Verify **IkeaDocuScan** pool is **Started**
4. If stopped, check Event Viewer for errors

Step 2: Check Application Logs

1. Navigate to logs directory:

   ```
   cd C:\inetpub\wwwroot\IkeaDocuScan\logs
   dir
   ```

2. Open most recent stdout log file

3. Look for:

   - □ "Now listening on: http://localhost:5000"
   - □ "Application started"
   - □ Any exception stack traces

Step 3: Check Windows Event Logs

1. Open **Event Viewer**
2. Navigate to **Windows Logs → Application**
3. Filter for:
   - Source: **IIS AspNetCore Module V2**
   - Source: **ASP.NET Core**
4. Check for errors or warnings

Step 4: Test Database Connection

1. Browse to application URL

2. Application should start without errors

3. Check logs for EF Core connection success:

   ```
   Entity Framework Core initialized
   Database connection successful
   ```

Step 5: Test Application Endpoints

Access these URLs and verify responses:

URL	Expected Result
https://docuscan.company.com	Home page loads
https://docuscan.company.com/health	Returns "Healthy"
https://docuscan.company.com/api/health	Returns JSON health status

Step 6: Test User Authentication

1. Browse to application from domain-joined machine
2. Verify automatic authentication with Windows credentials
3. Navigate to **Documents → Search Documents**

4. Verify access based on AD group membership

Step 7: Test Document Operations

1. **Search Documents:** Verify search functionality works
2. **View Document:** Click on a document to view details
3. **Check File Access:** Verify scanned files can be accessed
4. **Test Excel Export:** Generate an Excel export from a report

Step 8: Test Email Notifications (If Enabled)

1. Navigate to a feature that sends emails (e.g., Access Request)
2. Submit a test request
3. Verify email is received
4. Check SMTP logs if email not received

Step 9: Verify Version Number

Check application version is correct:

1. Browse to application
2. Check browser's Developer Tools → Network tab
3. Look for DLL version in response headers or assembly info

OR check file properties:

```
cd C:\inetpub\wwwroot\IkeaDocuScan
powershell "(Get-Item IkeaDocuScan-Web.dll).VersionInfo.FileVersion"
```

Step 10: Performance Check

Monitor application performance:

1. Open **Performance Monitor** (perfmon)
2. Add counters:
   - ASP.NET Core → Requests/Sec
   - .NET CLR Memory → # Bytes in all Heaps
   - Process → % Processor Time (w3wp)
3. Generate some load and verify metrics are reasonable

Troubleshooting Common Issues

Issue: Application Pool Stops Immediately

Symptoms: - Pool starts but stops after a few seconds - 502.5 error in browser

Resolution: 1. Check stdout logs in `logs` directory 2. Verify .NET 10.0 Runtime installed: `cmd dotnet --list-runtimes` 3. Should show: `Microsoft.AspNetCore.App 10.0.x` 4. Install ASP.NET Core Hosting Bundle 10.0 if missing

Issue: Database Connection Fails

Symptoms: - 500 error on any database operation - "Cannot open database" in logs - "Login failed for user 'docuscanch'" in logs

Resolution: 1. Verify connection string is encrypted in `secrets.encrypted.json` 2. Test SQL connection with docuscanch user: `cmd sqlcmd -S PROD-SQL-01 -d IkeaDocuScan -U docuscanch -P [password]` 3. Verify docuscanch user has database access: ```sql USE IkeaDocuScan;

– Check if login exists SELECT name, type_desc FROM sys.server_principals WHERE

name = ‘docuscanch’;

– Check if user exists in database SELECT name, type_desc FROM sys.database_principals WHERE name = ‘docuscanch’;

– Check user roles EXEC sp_helpuser ‘docuscanch’; ``` 4. If user missing, ensure migration scripts were executed completely 5. Verify password is correct in ConfigEncryptionTool output

Issue: Windows Authentication Not Working

Symptoms: - Prompted for credentials repeatedly - Anonymous user shown in logs

Resolution: 1. Verify Anonymous Authentication is **Disabled** in IIS 2. Verify Windows Authentication is **Enabled** 3. Check browser Intranet zone settings 4. Add site to Trusted Sites if needed

Issue: Cannot Access Scanned Files

Symptoms: - Files list is empty - “Access denied” errors

Resolution: 1. Verify path in appsettings.Local.json is correct 2. Test access as App Pool identity: `cmd runas /user:"IIS APPPOOL\IkeaDocuScan" "cmd.exe" dir \\fileserver\ScannedDocuments` 3. Grant permissions on file server if access denied

Issue: DPAPI Decryption Fails

Symptoms: - “Unable to decrypt configuration” errors - SMTP or database passwords not working

Resolution: 1. Verify secrets.encrypted.json was created using App Pool identity 2. Re-encrypt using ConfigEncryptionTool running as correct user 3. Alternative: Use environment variables instead

Issue: SignalR Not Working

Symptoms: - Real-time updates not working - WebSocket connection failures in browser console

Resolution: 1. Verify WebSocket Protocol feature installed in IIS 2. Check Application Request Routing (ARR): `IIS Manager → Server → Application Request Routing → Server Proxy Settings Enable proxy: OFF (unless needed)` 3. Verify firewall allows WebSocket connections

---

Rollback Procedure

If deployment fails and rollback is necessary:

Step 1: Stop Application Pool

```
%systemroot%\system32\inetsrv\appcmd stop apppool /apppool.name:"IkeaDocuS
```

Step 2: Restore Previous Files

1. Delete current deployment files
2. Copy previous version files from backup
3. Restore appsettings.Local.json from backup

Step 3: Rollback Database (If Necessary)

If migrations were applied that need reversal:

```
dotnet ef database update <PreviousMigrationName>
```

OR restore database backup:

```
USE master;
GO
ALTER DATABASE IkeaDocuScan SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
GO
RESTORE DATABASE IkeaDocuScan FROM DISK = 'C:\Backups\IkeaDocuScan_Backup.
GO
ALTER DATABASE IkeaDocuScan SET MULTI_USER;
GO
```

Step 4: Start Application Pool

```
%systemroot%\system32\inetsrv\appcmd start apppool /apppool.name:"IkeaDocu
```

Step 5: Verify Rollback Success

Test application functionality as per Step 5-8 in Post-Deployment Verification.

Appendix A: Configuration Reference

Required Configuration Sections

Section	Purpose	Required
ConnectionStrings	Database connection	□ Yes
IkeaDocuScan	Application settings	□ Yes
Email	SMTP configuration	□ Yes (if notifications enabled)
ExcelExport	Excel generation settings	□□ Optional (has defaults)
Logging	Logging configuration	□□ Optional (has defaults)

Sensitive Settings Checklist

These should be in `appsettings.Local.json` or `secrets.encrypted.json`:

In secrets.encrypted.json (DPAPI-encrypted): - [ ] ConnectionStrings:DefaultConnection (includes docuscanch password) - [ ] IkeaDocuScan:ScannedFilesPath

In appsettings.Local.json (server-specific): - [ ] IkeaDocuScan:DomainName (ikea.com) - [ ] IkeaDocuScan:UserEmail:LDAPRoot - [ ] IkeaDocuScan:UserEmail:LDAPFilter - [ ] IkeaDocuScan:EmailGroups:LDAPRoot - [ ] IkeaDocuScan:EmailGroups:LDAPFilter - [ ] IkeaDocuScan:ADGroupReader (IKEA group) - [ ] IkeaDocuScan:ADGroupPublisher (IKEA group) - [ ] IkeaDocuScan:ADGroupSuperUser (IKEA group) - [ ] Email:SmtpHost (smtp-gw.ikea.com) - [ ] Email:ApplicationUrl (https://docuscan.ikea.com) - [ ] ExcelExport:ApplicationUrl (https://docuscan.ikea.com)

Appendix B: Health Check Endpoints

The application includes health check endpoints:

Endpoint	Purpose	Response
/health	Basic health check	“Healthy” (200 OK)
/health/ready	Readiness probe	JSON status
/health/live	Liveness probe	JSON status

Example Response:

```
{
  "status": "Healthy",
  "checks": [
    {
      "name": "Database",
      "status": "Healthy",
      "description": "Connection successful"
    },
    {
      "name": "FileSystem",
      "status": "Healthy",
      "description": "Scanned files accessible"
    }
  ],
  "totalDuration": "00:00:00.1234567"
}
```

Use these endpoints for monitoring and load balancer health checks.

Document Version

Version	Date	Author	Changes
1.0	2025-01-06	System	Initial deployment plan

End of Deployment Plan