

FinanceDIPS2 — Complete Handover & Developer Guide

The single document for taking over FinanceDIPS2 end to end: connect, get the code, change it, and deploy it. No prior knowledge assumed.

How to use this document. This is everything you need. The **only** other item is the SSH key file `mytoyota_deploy`, sent to you separately over a secure channel (1Password / USB — never email). Save it, then follow **§1 Quick Start**. Everything after that is reference. Two `PLACEHOLDER` secrets (the key and the DB `sa` password) are noted where they're used — you retrieve the password yourself once you're on the server.

1 · Quick start — your first 15 minutes

```
# 1) Save the SSH key you were given, then lock it down (OpenSSH refuses a world-readable key):
# put it at %USERPROFILE%.ssh\mytoyota_deploy then:
icacls "%USERPROFILE%.ssh\mytoyota_deploy" /inheritance:r /grant:r "%USERNAME%:R"

# 2) Connect (ALWAYS write Administrator@... — ssh defaults to your local username otherwise):
ssh -i %USERPROFILE%.ssh\mytoyota_deploy Administrator@197.242.64.87
# refused / timed out? your IP needs whitelisting - see Troubleshooting (§13).

# 3) You're on the server. Source, git, the .NET 9 SDK and the live app are all here. Get latest:
cd C:\Sources\FinanceDIPS2
git pull

# 4) Build it:
cd FinanceDIPS2
dotnet build -c Release          # expect: 0 Error(s)

# 5) Make a change, then commit + push to the on-box repo:
git add -A
git commit -m "my change"
git push

# 6) Publish it live (full reversible recipe in §11).
```

That's the whole loop — **edit** → **build** → **commit** → **deploy**, all on one server (197.242.64.87). Sections 2–14 are the detail behind each step.

2 · What you are working on

ITEM	VALUE
Live URL	<code>https://financedips.ampleinsurance.co.za</code> (fronted by Cloudflare → origin below)
App / stack	FinanceDIPS2 — ASP.NET Core Blazor Server, .NET 9 , EF Core 9 + ASP.NET Identity, Syncfusion & iTextSharp. Runs in-process under IIS. Assembly <code>FinanceDIPS2.dll</code> .
The one server	AppsoluteIT-Server — public <code>197.242.64.87</code> , internal <code>10.50.20.3</code> = <code>host.appsoluteit.co.za</code> . Windows + IIS + OpenSSH. Has the .NET 9 SDK and git 2.55 (<code>C:\Git</code>) installed.
Git repo (self-hosted)	Central bare repo <code>C:\GitRepos\FinanceDIPS2.git</code> (= <code>origin</code>); working copy <code>C:\Sources\FinanceDIPS2</code> . No external remote.
IIS site	<code>Ample</code> - <code>financedips.ampleinsurance.co.za</code> (app pool same name); path <code>C:\inetpub\wwwroot\Ample\FinanceDips\<yyyy-MM-dd></code> (new dated folder per deploy)
Databases	AmpleFDIPS (main) + AMPLEREFFERAL_DB (referrals), SQL instance <code>AppsoluteIT-Server\SQL</code> reached as <code>host.appsoluteit.co.za</code> . Auth <code>sa</code> . Port 1433 not public.

Not to be confused with `AmpleFinanceSystem` (a different, older app, DB `AMPLE_DB`). It does **not** serve financedips. The live app is **FinanceDIPS2** only.

3 · Access you need

FOR	WHAT / HOW
The server (everything)	SSH as <code>Administrator@197.242.64.87</code> with the ED25519 key <code>mytoyota_deploy</code> (sent to you securely). Your PC's public IP may need whitelisting on port 22 — a one-time ask to Dirk van Tonder (AppsoluteIT infra). This is the only access required.
Database <code>sa</code> password	In the server's live <code>appsettings.json</code> (open it once you're on the box) and the Ample credentials workbook. Shown as <code><SA_PASSWORD></code> below.

4 · Source & the git workflow (all on the server)

The repo lives on the box. Work in `C:\Sources\FinanceDIPS2` — git is installed and `origin` is the local central repo `C:\GitRepos\FinanceDIPS2.git`. Everyday loop:

```
cd C:\Sources\FinanceDIPS2
git pull                # latest from the central repo on this box
# ... edit, build ($5) ...
git add -A
git commit -m "what you changed"
git push                # back to C:\GitRepos\FinanceDIPS2.git (this server)
```

Want a copy on your own PC too? (optional)

```
# pull the working folder down over SSH:
scp -i %USERPROFILE%\ssh\mytoyota_deploy -r Administrator@197.242.64.87:C:/Sources/FinanceDIPS2 .\FinanceDIPS2
# or clone the central repo over SSH:
git clone ssh://Administrator@197.242.64.87/C:/GitRepos/FinanceDIPS2.git
```

Solution: `FinanceDIPS2.sln` ; project: `FinanceDIPS2\FinanceDIPS2\FinanceDIPS2.csproj` .

5 · Build

```
cd C:\Sources\FinanceDIPS2\FinanceDIPS2
dotnet restore
dotnet build -c Release      # should say 0 Error(s)

# EF Core tools must match EF Core 9 (a global v10 tool FAILS on this net9 project).
# One-time, in the project folder:
dotnet new tool-manifest
dotnet tool install dotnet-ef --version 9.0.3
dotnet ef --version          # should print 9.0.3
```

The server already has the .NET 9 SDK and git 2.55, so cloning, building, committing and publishing all happen right here — nothing else to install.

Running the app locally needs DB access — point `appsettings.Development.json` at a copy of `AmpleFDIPS` , or tunnel to the live DB (§9).

That is the live database — prefer a restored copy for dev.

6 · How the code is organised

PATH	WHAT'S THERE
<code>Program.cs</code>	Startup / DI: Identity, EF <code>ApplicationDbContext</code> , services, controllers.
<code>Data/ApplicationDbContext.cs</code>	EF context (Identity + <code>Invoices</code> , <code>Referrals</code> , <code>AuthorizedUsers</code> , <code>NotificationRecipients</code>) + seed data.
<code>Data/Migrations/</code>	EF migrations — add one whenever you change the model.
<code>Models/</code>	Entities & DTOs.
<code>Services/</code>	<code>InvoiceService</code> , <code>EmailService</code> , <code>UserAccessService</code> , <code>NotificationRecipientService</code> , referral services.
<code>Components/Pages/</code>	Blazor routes: Home, InvoiceCreate, InvoiceDetails, InvoicePayments, Referrals, ManageUsers , ManageRecipients .
<code>Components/Layout/NavMenu.razor</code>	Sidebar; shows finance features only to authorised users.

7 · How permissions work

Access to the finance features is a **database allow-list**: table `AuthorizedUsers` , read via `IUserAccessService` (cached ~30s). A logged-in user whose email is an active row sees the gated features.

```
// gate a page or nav item:
@Inject FinanceDIPS2.Services.IUserAccessService UserAccess
private HashSet<string> _allowed = new(StringComparer.OrdinalIgnoreCase);
protected override async Task OnInitializedAsync() => _allowed = await UserAccess.GetAllowedEmailsAsync();
bool isAllowed = email != null && _allowed.Contains(email.ToLower());
```

Add/remove people in the app at **/manage-users** — no code change. **Don't** reintroduce hardcoded email lists (they were removed from `NavMenu/Home/InvoiceDetails`).

8 · How email recipients work

Recipients live in table `NotificationRecipients` via `INotificationRecipientService` , with flags **ForInvoices** (Create-Invoice notify picker) and **ForReferrals** (auto-emailed on referral events). SMTP is in `appsettings.json` (`Email` , smtp2go). Manage them at **/manage-recipients**.

9 · Connect to the database

```
# A) direct (only if 1433 is reachable to you):
sqlcmd -S host.appsoluteit.co.za -U sa -P <SA_PASSWORD> -d AmpleFDIPS -Q "SELECT DB_NAME()"

# B) usual: you're already on the box over SSH, so just run it there.

# C) tunnel + SSMS from your PC:
ssh -i %USERPROFILE%\ssh\mytoyota_deploy -L 14333:localhost:1433 Administrator@197.242.64.87
# then SSMS -> localhost,14333 (sa / <SA_PASSWORD>, db AmpleFDIPS)
```

10 · Make a database / schema change

```
# 1) add/adjust an entity in Models/ + its DbSet in ApplicationDbContext.cs
# 2) create a migration:
dotnet ef migrations add MyChangeName
# 3) open it and REMOVE anything you didn't intend (EF can bundle pre-existing model drift). Keep additive.
# 4) idempotent script for just your migration:
dotnet ef migrations script <previousMigrationId> <yourMigrationId> --idempotent -o deploy\change.sql
# 5) apply it (you're on the box):
sqlcmd -S host.appsoluteit.co.za -U sa -P <SA_PASSWORD> -d AmpleFDIPS -b -i deploy\change.sql
```

Never run `dotnet ef database update` at production, and never include destructive operations. Additive only; review the script first.

11 · Build & deploy to production

Reversible: publish to a NEW dated folder, preserve live config, repoint, verify, keep the old folder for one-command rollback. Never overwrite the running folder.

```
cd C:\Sources\FinanceDIPS2
git pull
set NEW=C:\inetpub\wwwroot\Ample\FinanceDips\<yyyy-MM-dd>
dotnet publish FinanceDIPS2\FinanceDIPS2\FinanceDIPS2.csproj -c Release -o "%NEW%"

# PRESERVE live config (real DB + SMTP secrets) - copy from the CURRENT live folder:
copy /Y "<current-live-folder>\appsettings.json" "%NEW%\appsettings.json"

# repoint + recycle:
C:\Windows\System32\inetsrv\appcmd.exe set vdir "Ample - financedips.ampleinsurance.co.za/" /physicalPath:"%NEW%"
C:\Windows\System32\inetsrv\appcmd.exe recycle apppool /apppool.name:"Ample - financedips.ampleinsurance.co.za"

# verify (healthy = 302 -> /Account/Login; login page = 200):
curl.exe -s -k -D - -o NUL --resolve financedips.ampleinsurance.co.za:443:127.0.0.1 https://financedips.ampleinsurance.co.za/
```

Rollback

```
C:\Windows\System32\inetsrv\appcmd.exe set vdir "Ample - financedips.ampleinsurance.co.za/" /physicalPath:"<previous-dated-folder>"
C:\Windows\System32\inetsrv\appcmd.exe recycle apppool /apppool.name:"Ample - financedips.ampleinsurance.co.za"
```

Never overwrite `appsettings.json` with the repo copy — the repo version lacks the working prod DB/SMTP secrets. Copy the live one forward. New additive tables are harmless to an older build, so a folder rollback alone is a safe revert.

12 · Worked example — add a new admin-only page

- Create `Components/Pages/MyPage.razor` with `@page "/my-page"` + `@attribute [Authorize]`.
- Inject `IUserService`, load the allowed set in `OnInitializedAsync`, render content only for allowed emails (§7).
- Add a gated nav link in `NavMenu.razor` inside the `@if (isAllowedUser)` block.
- New table? Follow §10.
- Build (§5), commit & push (§4), deploy (§11). Verify logged in as an authorised user.

13 · Troubleshooting

SYMPTOM	FIX
SSH "connection refused / timed out"	Your PC's public IP isn't whitelisted on port 22. Get it at https://api.ipify.org and ask Dirk van Tonder to allow it for <code>.87</code> (one-time).
SSH "Permission denied (publickey)" / asks for a password	You didn't pass <code>-i <key></code> , or the key file's permissions are too open — re-run the <code>icacls</code> line in §1.
<code>git</code> "is not recognized"	Open a new SSH session (git was added to PATH recently). Worst case call <code>C:\Git\cmd\git.exe</code> directly.
<code>dotnet ef</code> error about <code>System.Runtime 10.0.0</code>	You're using the global v10 tool. In the project folder use the pinned local one: <code>dotnet tool install dotnet-ef --version 9.0.3</code> .
<code>sqlcmd</code> "Login failed for user 'sa'" from your PC	1433 isn't public. Run <code>sqlcmd</code> on the box over SSH, or use the <code>-L</code> tunnel (§9). Password is in the live <code>appsettings.json</code> .
Site errors right after a deploy	Roll back: repoint the IIS vdir to the previous dated folder and recycle the pool (§11 Rollback), then investigate.
User logs in but sees no finance menus	Their email isn't in <code>AuthorizedUsers</code> — add them at <code>/manage-users</code> .
Notification emails not arriving	Check the recipient is active with the right flag at <code>/manage-recipients</code> , and the <code>Email</code> (smtp2go) block in <code>appsettings.json</code> .

14 · Secrets checklist

PLACEHOLDER	WHERE TO GET IT
<code>{mytoyota_deploy}</code> private key	Sent to you separately (1Password/USB). Save at <code>%USERPROFILE%\.ssh\mytoyota_deploy</code> .
<code><SA_PASSWORD></code>	The server's live <code>appsettings.json</code> (visible once you SSH in) or the Ample credentials workbook.

Security: the SSH key grants Administrator on a box hosting ~50+ production sites. Treat it and this document as passwords — keep them local, never email or commit them. If a PC is retired, remove its public key from `C:\ProgramData\ssh\administrators_authorized_keys` on the server.

Generated 2026-07-18 for Ample Insurance / FinanceDIPS2. Verified against server `197.242.64.87` — git 2.55 + .NET 9 SDK installed; central repo `C:\GitRepos\FinanceDIPS2.git` , working copy `C:\Sources\FinanceDIPS2` ; IIS site "Ample - financedips.ampleinsurance.co.za"; DB `AmpleFDIPS` . This single document supersedes the earlier onboarding / deploy-runbook notes. (A one-time code snapshot also exists in Azure DevOps from the initial import, but the server repo is the home — Azure is not used.)