

# IndieStream Go-Live Deployment Strategy

## Summary

Deployment strategy for taking IndieStream from development to production, covering environment setup, database handling, testing strategy, and content migration.

---

## Quick Answers to Your Questions

Question	Answer
Dev, Acc, Prod or just Dev + Prod?	<b>Dev + Prod</b> is sufficient
What to do with database?	Create new prod Supabase project, run migrations
Where to test with friends?	Vercel preview deployments (dev environment)
Can we reuse uploaded content?	Technically yes, but recommend fresh start for clean launch

---

## Recommended Environment Setup: Dev + Production

For a project of this size, **Dev + Prod is sufficient**. A Staging/Acceptance environment adds complexity without significant benefit until you have a larger team.

Environment	Purpose	When to Use
<b>Development</b>	Local dev, feature testing, beta testing with friends	Daily development
<b>Production</b>	Live platform for real users	After launch

### Why skip Staging?

- Solo/small team project
  - No enterprise compliance requirements
  - Vercel preview deployments give you "staging-like" functionality for free
- 

## Infrastructure Per Environment

Service	Dev	Production
<b>Vercel</b>	Preview deployments	Production deployment
<b>Supabase</b>	Current project (free tier)	New Pro project (\$25/mo)
<b>Cloudflare R2</b>	Current bucket	Same bucket OR new /prod prefix
<b>Stripe</b>	Test mode	Live mode (same account)
<b>Resend</b>	Free tier	Same

### Supabase: Two Separate Projects (Recommended)

Create a new `indiestream-prod` Supabase project for production:

- Complete isolation from dev data
  - Run all 39 migrations on the new project
  - Pro plan (\$25/mo) gives you daily backups, no row limits, better performance
- 

## Environment Variables Setup

**Current: Single `.env` file**

**Target: Environment-specific configuration**

**Local development:** Keep using `.env` or `.env.local`

**Production (Vercel Dashboard → Settings → Environment Variables):**

```
SUPABASE_URL=https://your-prod-project.supabase.co  
SUPABASE_KEY=your-prod-anon-key  
SUPABASE_SERVICE_KEY=your-prod-service-key  
STRIPE_SECRET_KEY=sk_live_...  
STRIPE_WEBHOOK_SECRET=whsec_...  
NUXT_PUBLIC_STRIPE_PUBLISHABLE_KEY=pk_live_...  
R2_BUCKET_NAME=indiestream-prod (or same bucket)  
NUXT_PUBLIC_R2_PUBLIC_URL=https://cdn.indiestream.app  
NUXT_PUBLIC_SITE_URL=https://indiestream.app
```

---

## Database Migration Strategy

**For Production Launch:**

1. Create fresh production Supabase project
2. Link and push migrations:

```
supabase link --project-ref your-prod-project-ref  
supabase db push
```

3. Verify schema matches dev

**Dev data will NOT transfer to production** - this is intentional for a clean start.

---

## Testing Strategy: Where to Test with Friends

**Phase 1: Vercel Preview (Pre-launch)**

- Push to a branch → Vercel creates preview URL
- Share URL: <https://indiestream-git-feature-yourname.vercel.app>
- Uses dev Supabase + Stripe test mode
- **This is your beta testing phase**

**Phase 2: Production Beta (Invite-Only)**

- Deploy to production with invite-only access
- Only invited users can access the platform

- Real Stripe, real data
  - Test everything before public launch
- 

## Invite-Only Access Feature

Add a "closed beta" mode where only invited users can access the site.

### Database Changes

```
-- Add invite_codes table
CREATE TABLE invite_codes (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    code VARCHAR(20) UNIQUE NOT NULL,
    created_by UUID REFERENCES profiles(id),
    used_by UUID REFERENCES profiles(id),
    used_at TIMESTAMPTZ,
    expires_at TIMESTAMPTZ,
    created_at TIMESTAMPTZ DEFAULT NOW()
);

-- Add invited flag to profiles
ALTER TABLE profiles ADD COLUMN is_invited BOOLEAN DEFAULT false;
```

### Implementation

1. **Environment variable:** NUXT\_PUBLIC\_INVITE\_ONLY=true

2. **Middleware** ( middleware/invite-only.global.ts ):

```
export default defineNuxtRouteMiddleware((to) => {
    const config = useRuntimeConfig()
    const user = useSupabaseUser()

    // Skip if not in invite-only mode
    if (!config.public.inviteOnly) return

    // Allow public pages: landing, login, register, invite redemption
    const publicPaths = ['/','/login','/register','/invite']
    if (publicPaths.some(p => to.path.startsWith(p))) return

    // Require login and invite for all other pages
    if (!user.value) return navigateTo('/login')

    // Check if user is invited (stored in profile)
    // Redirect to "request invite" page if not invited
})
```

3. **Invite redemption page** ( pages/invite/[code].vue ):

- Validates invite code

- Marks user as invited after registration/login
- Redirects to home

#### 4. Admin: Manage invites (add to admin dashboard):

- Generate invite codes
- See who used which code
- Revoke unused codes

### Flow

1. You generate invite codes in admin
2. Share codes with friends: `indiestream.app/invite/ABC123`
3. Friend clicks link, registers/logs in
4. Code is marked used, user is marked as invited
5. User can now access the full site

### Disable for Public Launch

When ready to go public:

- Set `NUXT_PUBLIC_INVITE_ONLY=false` in Vercel
  - Redeploy
  - Site is now open to everyone
- 

## Content Migration: Fresh Start Recommended

**Can you keep uploaded albums?** Technically yes, but **recommend fresh start**:

- Test data often has issues
- Clean start ensures data integrity
- Re-upload a few showcase albums for quality

**If you must preserve content:**

1. Export album/track data from dev Supabase
  2. Copy R2 files if using separate buckets
  3. Import records to prod Supabase
- 

## Go-Live Checklist

### Pre-Launch Setup

- Create production Supabase project (Pro plan)
- Run all migrations: `supabase db push`
- Run invite system migration
- Configure R2 bucket for production
- Set up Stripe live mode webhooks
- Add custom domain in Vercel
- Set all env vars in Vercel (including `NUXT_PUBLIC_INVITE_ONLY=true` )
- Generate invite codes for beta testers

**Create `vercel.json` :**

```
{
  "buildCommand": "npm run build",
  "outputDirectory": ".output",
  "framework": "nuxt"
}
```

## Final Tests Before Launch

- User registration works
- Stripe payments process (real \$1 test)
- Audio uploads work
- Audio playback works
- Emails send
- Admin dashboard accessible

## Cost Estimate (Monthly)

Service	Cost
Vercel	Free (Hobby) or \$20 (Pro)
Supabase Pro	\$25
Cloudflare R2	~\$0.015/GB (10GB free)
Stripe	2.9% + \$0.30 per transaction
Resend	Free (3k emails/mo)

**Estimated: \$25-50/month at launch**

## Files to Create/Modify

File	Purpose
supabase/migrations/XXX_add_invite_system.sql	Invite codes table + profile flag
middleware/invite-only.global.ts	Gate all pages for uninvited users
pages/invite/[code].vue	Invite code redemption page
server/api/invites/redeem.post.ts	Validate and redeem invite code
server/api/admin/invites.get.ts	List all invite codes
server/api/admin/invites.post.ts	Generate new invite codes
components/admin/AdminInvitesTab.vue	Admin UI for managing invites
pages/admin/index.vue	Add Invites tab
nuxt.config.ts	Add <code>inviteOnly</code> runtime config

---

## Verification

After implementation:

1. **Build passes:** `npm run build`
2. **Invite flow works:** Generate code → Share → Redeem → Access granted
3. **Uninvited blocked:** New user without code sees "request invite" page
4. **Admin works:** Can generate, view, and manage invite codes
5. **Toggle works:** Setting `INVITE_ONLY=false` opens site to everyone