



Deployment Guide - Business Archetypes Static App

What's Ready to Push

Your complete static app is ready for deployment with:

Core Application Files

- `index.html` - Main application
- `manifest.json` - PWA configuration
- `sw.js` - Service Worker
- `vercel.json` - Deployment configuration

JavaScript Files

- `js/app.js` - Main application logic with MailerLite integration
- `js/archetypen.js` - 12 archetype definitions
- `js/questions.js` - 185 assessment questions

API Serverless Functions

- `api/mailchimp.js` - Legacy Mailchimp integration
- `api/mailerlite.js` - Active MailerLite integration (English, new API token)

Email Templates (MailerLite)

- `mailerlite_templates/email1_archetype_result.html` - Results email
- `mailerlite_templates/email2_archetype_guidance.html` - Guidance + CTA
- `mailerlite_templates/email3_business_sovereignty.html` - Business strategies + CTA

Documentation

- `README.md` - Main project documentation
- `MAILERLITE_SETUP_GUIDE.md` - Complete MailerLite setup guide
- `MAILERLITE_INTEGRATION.md` - Integration details
- `MIGRATION_MAILCHIMP_TO_MAILERLITE.md` - Migration notes
- Plus 20+ other documentation files

Excluded Files (.gitignore)

- `*.pdf` - PDF versions of markdown files
- `test-*.*.html` - Test files
- `node_modules/` - Dependencies
- `.env*` - Environment files
- `.DS_Store`, `Thumbs.db` - OS files

Git Setup (Already Done)

```
# Git repository initialized ✓
# All files staged ✓
```

Deployment Steps

Option 1: Deploy to New GitHub Repository (Recommended)

```
# 1. Create a new repository on GitHub (e.g., business-archetypes-static)
#     - Go to https://github.com/new
#     - Name: business-archetypes-static
#     - Don't initialize with README (we have one)

# 2. Navigate to project folder
cd /home/ubuntu/archetypen_static

# 3. Commit all files
git commit -m "Initial commit: Business Archetypes with MailerLite integration"

# 4. Add remote (replace YOUR_USERNAME with your GitHub username)
git remote add origin https://github.com/YOUR_USERNAME/business-archetypes-static.git

# 5. Push to GitHub
git branch -M main
git push -u origin main
```

Option 2: Push to Existing Repository

```
cd /home/ubuntu/archetypen_static

# Commit files
git commit -m "Update: MailerLite integration with English templates"

# Add your existing remote (if not already added)
git remote add origin YOUR_EXISTING_REPO_URL

# Push
git push -u origin main
```

Option 3: Deploy Directly with Vercel CLI

```
cd /home/ubuntu/archetypen_static

# Install Vercel CLI (if not installed)
npm i -g vercel

# Login to Vercel
vercel login

# Deploy to production
vercel --prod
```

After Deployment

1. Test the Application

Production URL: `https://your-app.vercel.app` (or your custom domain)

Test Checklist:

- [] Complete the archetype test
- [] Verify email received via EmailJS
- [] Check MailerLite dashboard for new subscriber
- [] Verify all 10 custom fields populated
- [] Confirm subscriber added to Group 176508406386918528

2. Set Up MailerLite Automation

Follow [MAILERLITE_SETUP_GUIDE.md](#) (`./MAILERLITE_SETUP_GUIDE.md`) to:

1. **Import email templates** to MailerLite
2. **Create automation workflow** triggered by group join
3. **Add 3 emails** with proper delays:
 - Email 1: Immediate (archetype results)
 - Email 2: +3 days (guidance + Shadow Work intro)
 - Email 3: +6 days (business sovereignty + final CTA)
4. **Activate automation**

3. Update EmailJS Configuration (if needed)

Edit `js/app.js` to update EmailJS settings:

```
const EMAILJS_CONFIG = {
  serviceId: 'YOUR_SERVICE_ID',
  templateId: 'YOUR_TEMPLATE_ID',
  ownerTemplateId: 'YOUR_OWNER_TEMPLATE_ID',
  publicKey: 'YOUR_PUBLIC_KEY',
  ownerEmail: 'your@email.com'
};
```

Then commit and push changes:

```
git add js/app.js
git commit -m "Update EmailJS configuration"
git push
```



Monitoring

Vercel Function Logs

1. Go to [Vercel Dashboard](#) (`https://vercel.com/dashboard`)
2. Select your project
3. Click “Functions” tab
4. Monitor `api/mailerlite.js` logs

MailerLite Dashboard

1. Go to [MailerLite](https://app.mailerlite.com) (<https://app.mailerlite.com>)
2. **Subscribers** → View new subscribers
3. **Groups** → Check Group 176508406386918528
4. **Automations** → Monitor email sequence performance

EmailJS Dashboard

1. Go to [EmailJS](https://dashboard.emailjs.com) (<https://dashboard.emailjs.com>)
2. Monitor sent emails
3. Check quota usage

Making Updates

Update Code

```
cd /home/ubuntu/archetypen_static

# Make your changes
nano js/app.js # or any file

# Stage changes
git add .

# Commit
git commit -m "Your update message"

# Push
git push
```

Vercel Auto-Deploy

Vercel automatically redeploys when you push to the main branch (if connected to Git).

Manual Deploy

```
vercel --prod
```

What Happens When User Completes Test

1. **User submits email and completes 185 questions**
2. **Results calculated** (top 3 archetypes with percentages)
3. **EmailJS sends immediate email** with detailed results to user and owner
4. **MailerLite API called** via `/api/mailerlite` serverless function
5. **User added to MailerLite** with 10 custom fields:
 - `name`, `archetype`, `archetype_percent`
 - `archetype_1` to `archetype_3` (with percentages)
 - `test_date`
6. **User added to Group 176508406386918528**
7. **Automation triggers** → Email sequence begins:
 - **Day 0:** Welcome + results

- **Day 3:** Guidance + Shadow Work intro + CTA
- **Day 6:** Business sovereignty + final CTA

Troubleshooting

API Key Not Working

The MailerLite API token is hardcoded in `api/mailerlite.js`. If it expires or changes:

```
// Edit api/mailerlite.js
const MAILERLITE_API_KEY = 'YOUR_NEW_TOKEN';
```

Then push:

```
git add api/mailerlite.js
git commit -m "Update MailerLite API token"
git push
```

Emails Not Triggering

1. Check MailerLite automation is “Active”
2. Verify Group ID matches: 176508406386918528
3. Check subscriber was added to group
4. Review Vercel function logs for errors

CORS Errors

If you see CORS errors, it means the serverless function isn’t working. Ensure:

- `vercel.json` includes `api/mailerlite.js`
- Function is deployed to Vercel
- Not testing on `file://` protocol (use deployed URL)



Environment Variables (Optional)

Currently, the API token is hardcoded in `api/mailerlite.js` for simplicity.

If you prefer environment variables:

1. **Vercel Dashboard** → Settings → Environment Variables
2. Add: `MAILERLITE_API_KEY` = your token
3. Update `api/mailerlite.js`:

```
const MAILERLITE_API_KEY = process.env.MAILERLITE_API_KEY || 'fallback_token';
```

1. Redeploy

Final Checklist

- [] Git repository initialized
- [] All files committed
- [] Pushed to GitHub/remote repository

- [] Deployed to Vercel (or other platform)
- [] Application tested in production
- [] EmailJS sending test results
- [] MailerLite receiving subscribers with custom fields
- [] MailerLite automation created and activated
- [] Email templates imported to MailerLite
- [] Test email sequence working

You're Ready!

Your Business Archetypes application is now:

- Version controlled with Git
- Ready to push to any Git hosting service
- Configured for Vercel deployment
- Integrated with MailerLite for automated sequences
- Fully documented

Next step: Push to Git and deploy! 