

Azure Deployment-Anleitung für Krankmeldungs-App

Diese Anleitung erklärt, wie Sie die Krankmeldungs-App kostengünstig und serverless in Azure deployen.

Übersicht

Die App wurde für Azure serverless Hosting optimiert und kann auf verschiedene Weise deployed werden:

1. **Azure Static Web Apps** (Empfohlen) - Automatische CI/CD, kostenlose SSL, globales CDN
2. **Azure Container Apps** - Für erweiterte Serverless-Features
3. **Azure App Service** - Traditionelles Hosting mit serverless Skalierung

Voraussetzungen

- Azure-Konto (kostenlose Registrierung möglich)
- GitHub-Repository mit der App
- Azure CLI (optional, für CLI-Deployment)

Deployment-Option 1: Azure Static Web Apps (Empfohlen)

Schritt 1: Code zu GitHub hochladen

```
# Lokales Git-Repository initialisieren
cd /path/to/krankmeldung-app
git init
git add .
git commit -m "Initial commit: Krankmeldungs-App"

# Mit GitHub-Repository verbinden
git remote add origin https://github.com/IHR_GITHUB_USERNAME/krankmeldung-app.git
git push -u origin main
```

Schritt 2: Azure Static Web Apps erstellen

1. **Azure Portal öffnen:** <https://portal.azure.com>
2. **“Static Web Apps” suchen** und auswählen
3. **“Create”** klicken
4. **Einstellungen konfigurieren:**
 - **Resource Group:** Neue erstellen: `krankmeldung-rg`
 - **Name:** `krankmeldung-app`
 - **Plan Type:** `Free` (kostenlos)
 - **Region:** `West Europe` oder `North Europe`
 - **Source:** `GitHub`
 - **Repository:** Ihr GitHub-Repository auswählen
 - **Branch:** `main`

Schritt 3: Build-Konfiguration

```
# .github/workflows/azure-static-web-apps-xxx.yml
# Diese Datei wird automatisch erstellt
name: Azure Static Web Apps CI/CD

on:
  push:
    branches:
      - main
  pull_request:
    types: [opened, synchronize, reopened, closed]
    branches:
      - main

jobs:
  build_and_deploy_job:
    if: github.event_name == 'push' || (github.event_name == 'pull_request' &&
    github.event.action != 'closed')
    runs-on: ubuntu-latest
    name: Build and Deploy Job
    steps:
      - uses: actions/checkout@v3
        with:
          submodules: true
      - name: Build And Deploy
        id: builddeploy
        uses: Azure/static-web-apps-deploy@v1
        with:
          azure_static_web_apps_api_token: $
          {{ secrets.AZURE_STATIC_WEB_APPS_API_TOKEN }}
          repo_token: ${{ secrets.GITHUB_TOKEN }}
          action: "upload"
          app_location: "/"app" # NextJS app location
          api_location: "" # API source code path
          output_location: ".next" # Built app content directory
          app_build_command: "npm run build"
```

Schritt 4: Umgebungsvariablen konfigurieren

Im Azure Portal unter Ihrer Static Web App:

1. “Configuration” → “Environment variables” öffnen
2. Folgende Variablen hinzufügen:

```
# Datenbank (Azure Database for PostgreSQL)
DATABASE_URL=postgresql://username:password@server.postgres.database.azure.com:5432/database

# Authentication
NEXTAUTH_URL=https://IHR_APP_NAME.azurestaticapps.net
NEXTAUTH_SECRET=ihr_geheimer_32_zeichen_schlüssel

# Optional: Application Insights (für Monitoring)
APPLICATIONINSIGHTS_CONNECTION_STRING=InstrumentationKey=xxx
```

Datenbank-Setup: Azure Database for PostgreSQL

Option A: Serverless (Empfohlen für Kostenoptimierung)

```
# Azure CLI Commands
az postgres flexible-server create \
  --resource-group krankmeldung-rg \
  --name krankmeldung-db \
  --location westeurope \
  --admin-user dbadmin \
  --admin-password IHR_SICHERES_PASSWORT \
  --sku-name Standard_B1ms \
  --tier Burstable \
  --compute-units 1 \
  --storage-size 32 \
  --version 13
```

Option B: Azure Portal

1. "Azure Database for PostgreSQL" erstellen
2. "Flexible Server" wählen
3. Konfiguration:
 - **Server name:** krankmeldung-db
 - **Compute tier:** Burstable (kostengünstig)
 - **Compute size:** Standard_B1ms
 - **Storage:** 32 GB
 - **Firewall:** Azure Services zugelassen

Datenbank initialisieren

```
# Lokale Verbindung zur Azure-Datenbank
psql "host=krankmeldung-db.postgres.database.azure.com port=5432 dbname=postgres
user=dbadmin@krankmeldung-db sslmode=require"

# Prisma Migration ausführen
npx prisma db push --schema=./app/prisma/schema.prisma
npx prisma db seed --schema=./app/prisma/schema.prisma
```

Microsoft 365 EntraID Integration (Erweitert)

Hinweis: Die aktuelle App verwendet password-based Authentication. Für EntraID Integration sind zusätzliche Schritte erforderlich:

Schritt 1: App Registration in Azure AD

```
# Azure CLI App Registration
az ad app create \
  --display-name "Krankmeldung App" \
  --web-redirect-uris "https://IHR_APP_NAME.azurestaticapps.net/api/auth/callback/
azure-ad" \
  --enable-id-token-issuance true
```

Schritt 2: NextAuth Azure AD Provider

```
// lib/auth.ts anpassen
import AzureADProvider from "next-auth/providers/azure-ad"

export const authOptions: NextAuthOptions = {
  providers: [
    AzureADProvider({
      clientId: process.env.AZURE_AD_CLIENT_ID!,
      clientSecret: process.env.AZURE_AD_CLIENT_SECRET!,
      tenantId: process.env.AZURE_AD_TENANT_ID!,
    }),
    // Existing CredentialsProvider...
  ],
  // ... rest of config
}
```

Schritt 3: Umgebungsvariablen ergänzen

```
AZURE_AD_CLIENT_ID=6fdb4d2c-9057-40b2-b2ec-5efe2f7efa44
AZURE_AD_CLIENT_SECRET=IHR_CLIENT_SECRET
AZURE_AD_TENANT_ID=cb04a716-c693-40b8-ad22-bc7a7f8d525b
```

Monitoring und Logging

Application Insights einrichten

```
# Application Insights Resource erstellen
az monitor app-insights component create \
  --resource-group krankmeldung-rg \
  --app krankmeldung-insights \
  --location westeuropa \
  --application-type web
```

Next.js Integration

```
// next.config.js
module.exports = {
  experimental: {
    instrumentationHook: true,
  },
  env: {
    APPLICATIONINSIGHTS_CONNECTION_STRING: process.env.APPLICATIONINSIGHTS_CONNECTION_STRING,
  },
}
```

Kostenoptimierung

Geschätzte monatliche Kosten (EUR):

- Azure Static Web Apps: €0 (Free Tier)

- **Azure Database for PostgreSQL:** €8-15 (Burstable B1ms)
- **Application Insights:** €0-5 (erste 5GB kostenlos)
- **Gesamt:** ~€8-20/Monat

Kostenspar-Tipps:

1. **Free Tier nutzen** wo möglich
2. **Auto-pause** für Entwicklungs-Datenbanken
3. **Resource Groups** für einfache Verwaltung
4. **Azure Cost Management** für Monitoring

Wichtige Sicherheitseinstellungen

1. Firewall-Regeln

```
# Nur Azure Services erlauben
az postgres flexible-server firewall-rule create \
  --resource-group krankmeldung-rg \
  --name krankmeldung-db \
  --rule-name AllowAzureServices \
  --start-ip-address 0.0.0.0 \
  --end-ip-address 0.0.0.0
```

2. SSL erzwingen

```
az postgres flexible-server parameter set \
  --resource-group krankmeldung-rg \
  --server-name krankmeldung-db \
  --name require_secure_transport \
  --value ON
```

3. Backup-Konfiguration

```
az postgres flexible-server configuration set \
  --resource-group krankmeldung-rg \
  --server-name krankmeldung-db \
  --name backup_retention_days \
  --value 7
```

Deployment-Troubleshooting

Häufige Probleme:

1. Build-Fehler:

```
bash
# Package.json Pfad prüfen
app_location: "/app"
```

2. Datenbankverbindung:

```
bash
# SSL-Modus in CONNECTION_STRING
DATABASE_URL="...?sslmode=require"
```

3. Environment Variables:

```
bash
```

```
# In Azure Portal Configuration Tab prüfen
```

Weitere Ressourcen

- [Azure Static Web Apps Dokumentation](https://docs.microsoft.com/azure/static-web-apps/) (https://docs.microsoft.com/azure/static-web-apps/)
- [Azure Database for PostgreSQL](https://docs.microsoft.com/azure/postgresql/) (https://docs.microsoft.com/azure/postgresql/)
- [NextAuth.js Azure AD Provider](https://next-auth.js.org/providers/azure-ad) (https://next-auth.js.org/providers/azure-ad)
- [Azure Cost Management](https://docs.microsoft.com/azure/cost-management-billing/) (https://docs.microsoft.com/azure/cost-management-billing/)

Nach dem Deployment

Test-Credentials:

- **E-Mail:** max.mustermann@netlution.de
- **Passwort:** test123

Features testen:

1. Login/Register
2. Krankmeldung erstellen
3. E-Mail-Vorschau
4. Übersicht anzeigen
5. Responsive Design

Support: Bei Fragen zum Deployment können Sie die Azure-Dokumentation oder die Community-Foren nutzen.