AI Event Planner SaaS - Deployment Guide

Overview

This guide walks you through deploying the AI Event Planner SaaS application to Azure with real AI agents.

Current Status

✅ **Application is configured for real agents**

- LangGraph-based AI agents for event planning
- Supports GPT-4, Google AI, and Tavily search
- Multi-tenant architecture with Azure PostgreSQL
- Comprehensive health check endpoints

Issues Identified & Fixed

1. Deploy Script Issue ❌ → ✅ Fixed

Problem: `scripts/deploy_to_azure.sh` uses `az webapp deploy` which has known Azure CLI issues.

Solution: Created new deployment workflow using GitHub Actions (recommended by Azure).

2. Missing Deployment Scripts ❌ → ✅ Fixed

Problem: No easy way to verify GitHub secrets or deployment status.

Solution: Created three new scripts:

- `scripts/setup_github_secrets.sh` - Verify and configure GitHub secrets
- `scripts/deploy_via_github.sh` - Deploy via GitHub Actions
- `scripts/verify_deployment.sh` - Verify deployment and agent functionality

Deployment Steps

Step 1: Configure GitHub Secrets

GitHub secrets are required for deployment with real agents. Run:

```
./scripts/setup_github_secrets.sh
```

This will guide you through setting up:

- `AZURE_CREDENTIALS` - Azure service principal for deployment
- `AZURE_RESOURCE_GROUP` - Resource group name (e.g., 'ai-event-planner-rg')

- `AZURE_LOCATION` - Azure region (e.g., 'eastus')
- `DATABASE_URL` - PostgreSQL connection string
- `SECRET_KEY` - JWT secret key
- `OPENAI_API_KEY` - For GPT-4 agents ⭐ **Required for real agents**
- `GOOGLE_API_KEY` - For Google AI features ⭐ **Required for real agents**
- `TAVILY_API_KEY` - For web search ⭐ **Required for real agents**

Manual Setup (if GitHub CLI not installed)

Visit: https://github.com/d1hawkins/AI-EventPlanner/settings/secrets/actions

Getting API Keys

1. **OpenAI API Key**: https://platform.openai.com/api-keys
2. **Google AI API Key**: https://aistudio.google.com/app/apikey
3. **Tavily API Key**: https://tavily.com/

Getting Azure Credentials

```
# Get your subscription ID
az account show --query id -o tsv

# Create service principal
az ad sp create-for-rbac \
  --name 'github-actions-ai-event-planner' \
  --role contributor \
  --scopes /subscriptions/{subscription-id}/resourceGroups/ai-event-
planner-rg \
  --sdk-auth
```

Copy the entire JSON output and add it as `AZURE_CREDENTIALS` secret.

Step 2: Deploy to Azure

Once secrets are configured, deploy using:

```
./scripts/deploy_via_github.sh
```

This will:

1. Commit any changes
2. Push to GitHub main branch
3. Trigger GitHub Actions workflow
4. Monitor deployment progress (if GitHub CLI installed)

The GitHub Actions workflow will:

- ✅ Run tests
- ✅ Verify all secrets are present
- ✅ Create/update Azure Web App with Python 3.10
- ✅ Deploy application code
- ✅ Set environment variables
- ✅ Run database migrations
- ✅ Verify deployment

Deployment takes 5-10 minutes.

Monitor at: https://github.com/d1hawkins/AI-EventPlanner/actions

Step 3: Verify Deployment

After deployment completes, verify everything is working:

```
./scripts/verify_deployment.sh
```

This script checks:

- ✅ Site accessibility
- ✅ Health endpoint response
- ✅ Real agent functionality
- ✅ Environment configuration
- ✅ Azure Web App status

Application URL

Once deployed, your application will be available at:

https://ai-event-planner-saas-py.azurewebsites.net

Health Check Endpoints

`/health`

Returns application health status including real agent test:

```
{
  "status": "healthy",
  "version": "1.0.0",
  "environment": "production",
  "real_agents_available": true,
  "agent_test": {
    "test_performed": true,
    "using_real_agent": true,
    "response_preview": "...",
    "error": null
```

```
    }
  }
```

`/debug/env`

Returns environment configuration status (without exposing secrets):

```
{
  "tavily_key_present": true,
  "google_key_present": true,
  "llm_model_present": true,
  "llm_model_value": "gpt-4",
  "real_agents_available": true
}
```

Agent Configuration

The application uses real LangGraph agents when these API keys are configured:

- **Coordinator Agent** - Orchestrates event planning workflow
- **Resource Planning Agent** - Manages resources and logistics
- **Financial Agent** - Handles budgeting and financial analysis
- **Stakeholder Management Agent** - Manages communications
- **Marketing Agent** - Creates marketing content and strategies
- **Project Management Agent** - Tracks tasks and timelines
- **Analytics Agent** - Provides insights and reports
- **Compliance Agent** - Ensures regulatory compliance

Agent Requirements

For real agents to work:

1. ✅ `OPENAI_API_KEY` must be set (for GPT-4)
2. ✅ `GOOGLE_API_KEY` must be set (for Google AI)
3. ✅ `TAVILY_API_KEY` must be set (for web search)
4. ✅ `LLM_MODEL=gpt-4` (default, already configured)
5. ✅ `LLM_PROVIDER=openai` (default, already configured)

Troubleshooting

Agents Not Working

If `/health` shows `using_real_agent: false`:

1. **Check API Keys are Set**:

```
az webapp config appsettings list \
  --name ai-event-planner-saas-py \
  --resource-group ai-event-planner-rg \
  --query "[?name=='OPENAI_API_KEY' || name=='GOOGLE_API_KEY' ||
name=='TAVILY_API_KEY']"
```

2. **Add Missing Keys**:

```
az webapp config appsettings set \
  --name ai-event-planner-saas-py \
  --resource-group ai-event-planner-rg \
  --settings \
    "OPENAI_API_KEY=sk-..." \
    "GOOGLE_API_KEY=..." \
    "TAVILY_API_KEY=..."
```

3. **Restart the App**:

```
az webapp restart \
  --name ai-event-planner-saas-py \
  --resource-group ai-event-planner-rg
```

Deployment Fails

1. **Check GitHub Actions Logs**: https://github.com/d1hawkins/AI-EventPlanner/actions
2. **Verify Secrets**: Run `./scripts/setup_github_secrets.sh`
3. **Check Azure Resources**: Ensure resource group and web app exist

View Logs

```
# Stream live logs
az webapp log tail \
  --name ai-event-planner-saas-py \
  --resource-group ai-event-planner-rg

# Download logs
az webapp log download \
  --name ai-event-planner-saas-py \
  --resource-group ai-event-planner-rg \
  --log-file webapp-logs.zip
```

Local Development

To test locally with real agents:

1. Copy `.env.test` to `.env`
2. Add your API keys to `.env`:

```
OPENAI_API_KEY=sk-...
GOOGLE_API_KEY=...
TAVILY_API_KEY=...
```

3. Run locally:

```
python -m uvicorn app.main_saas:app --reload
```

Next Steps

1. ✅ Configure GitHub secrets: `./scripts/setup_github_secrets.sh`
2. ✅ Deploy application: `./scripts/deploy_via_github.sh`
3. ✅ Verify deployment: `./scripts/verify_deployment.sh`
4. 🌐 Visit: https://ai-event-planner-saas-py.azurewebsites.net
5. 🧪 Test agent functionality
6. 📊 Monitor with Azure Application Insights

Support

For issues or questions:

- GitHub Issues: https://github.com/d1hawkins/AI-EventPlanner/issues
- Azure Support: https://portal.azure.com

Architecture

```
GitHub Actions (CI/CD)
    ↓
Azure Web App (Python 3.10)
    ↓
FastAPI Application (app/main_saas.py)
    ↓
LangGraph Agents (8 specialized agents)
    ↓
External APIs:
- OpenAI (GPT-4)
- Google AI
- Tavily (Search)
    ↓
Azure PostgreSQL (Database)
```

Files Created/Modified

New Files

- ✅ `scripts/setup_github_secrets.sh` - GitHub secrets management
- ✅ `scripts/deploy_via_github.sh` - GitHub Actions deployment
- ✅ `scripts/verify_deployment.sh` - Deployment verification
- ✅ `DEPLOYMENT_GUIDE.md` - This guide

Existing Files (Working)

- ✅ `.github/workflows/azure-deploy-saas.yml` - GitHub Actions workflow
- ✅ `app/main_saas.py` - FastAPI application with health checks
- ✅ `app/agents/agent_factory.py` - Agent creation with real LangGraph
- ✅ `app/config.py` - Configuration management
- ✅ `startup.sh` - Azure startup script
- ✅ `requirements.txt` - Python dependencies

Security Notes

- 🔒 Never commit API keys to git
- 🔒 Use GitHub Secrets for sensitive data
- 🔒 Rotate API keys regularly
- 🔒 Enable Azure Application Insights for monitoring
- 🔒 Use PostgreSQL with SSL (sslmode=require)