Azure Migration Deployment Fix - Applied

Summary

Fixed the database migration failure during Azure deployment where the migration script `migrate.py` could not be found on the deployed site.

Error Fixed:

```
/usr/bin/python3: can't open file
'/home/site/wwwroot/scripts/migrate.py': [Errno 2] No such file or
directory
```

Root Cause

The migration step was failing because:

1. The deployment packaging process wasn't ensuring `scripts/migrate.py` was reliably copied to the deployment zip
2. The migration invocation command didn't check if the file existed before trying to execute it
3. Error messages were unclear, making it difficult to diagnose whether the file was missing from the package or from the deployment

Changes Applied

1. Enhanced Deployment Package Creation

File: `.github/workflows/azure-deploy-saas.yml` (lines ~268-320)

What Changed:

- Added explicit `mkdir -p deploy/scripts` to ensure scripts directory exists
- Changed from copying entire `scripts/` directory to copying contents with `scripts/*`
- Added verification that `deploy/scripts/migrate.py` exists before zipping
- Enhanced zip validation to confirm `scripts/migrate.py` is present in the package
- Added failure exit codes if migration script is missing

Key Improvements:

```
# Ensure scripts are included - create deploy/scripts if needed
mkdir -p deploy/scripts

# Copy scripts directory contents if it exists
if [ -d scripts ]; then
  cp -r scripts/* deploy/scripts/ || true
fi
```

```
# Verify critical migration script is present
if [ ! -f deploy/scripts/migrate.py ]; then
  echo "❌ ERROR: deploy/scripts/migrate.py not found after copy -
migrations will fail" >&2
  exit 1
fi

# Validate it's in the zip
if unzip -l deploy.zip | grep -q "scripts/migrate.py"; then
  echo "✅ scripts/migrate.py found in zip"
else
  echo "❌ ERROR: migrate.py not found in zip!" >&2
  exit 1
fi
```

2. Robust Migration Invocation with Path Checking

File: `.github/workflows/azure-deploy-saas.yml` (lines ~505-520)

What Changed:

- Replaced simple command with conditional bash script that checks file existence first
- Tries both possible locations: `/home/site/wwwroot/scripts/migrate.py` and `/home/site/wwwroot/app/scripts/migrate.py`
- Returns clear error message "MIGRATE_NOT_FOUND" if neither file exists
- Exits with code 2 for missing file (distinct from execution errors)

Old Command:

```
$PYTHON_PATH /home/site/wwwroot/scripts/migrate.py || $PYTHON_PATH
/home/site/wwwroot/app/scripts/migrate.py
```

New Command:

```
if [ -f /home/site/wwwroot/scripts/migrate.py ]; then
  $PYTHON_PATH /home/site/wwwroot/scripts/migrate.py;
elif [ -f /home/site/wwwroot/app/scripts/migrate.py ]; then
  $PYTHON_PATH /home/site/wwwroot/app/scripts/migrate.py;
else
  echo 'ERROR: MIGRATE_NOT_FOUND - Neither
/home/site/wwwroot/scripts/migrate.py nor
/home/site/wwwroot/app/scripts/migrate.py exists' && exit 2;
fi
```

Benefits of This Approach

1. Early Detection

- CI build fails immediately if `migrate.py` isn't packaged correctly
- No need to wait for deployment to Azure to discover the issue
- Saves time and debugging effort

2. Clear Diagnostics

- Build logs show exactly what files are in the zip package
- Migration step provides clear error if file is missing
- Distinguishes between "file not found" vs "migration failed"

3. Robustness

- Works regardless of where `migrate.py` is located in the repo
- Handles both `/scripts/migrate.py` and `/app/scripts/migrate.py` locations
- Creates necessary directories automatically

4. Backward Compatibility

- Still supports alternative locations as fallback
- Doesn't break existing deployments
- Gracefully handles edge cases

Verification Steps

When this fix is deployed, the GitHub Actions workflow will:

1. ✅ Verify `scripts/migrate.py` exists locally before packaging
2. ✅ Copy it to `deploy/scripts/migrate.py` in the staging directory
3. ✅ Confirm it's present in `deploy.zip` before upload
4. ✅ After deployment, verify it exists on Azure at `/site/wwwroot/scripts/`
5. ✅ Check if file exists before attempting to run migration
6. ✅ Provide clear error message if file is missing

Testing Recommendations

1. **Verify in CI Logs:**
   - Check "Create deployment package" step shows: ✅ `scripts/migrate.py found in zip`
   - Check "Verify deployed files on Kudu" step shows: ✅ `migrate.py found in scripts directory`
2. **Monitor Migration Step:**
   - Should see: 🚀 `Running database migration with Python: [path]`
   - Should see: ✅ `Database migration completed successfully`
   - If it fails, check for "MIGRATE_NOT_FOUND" error

3. **Manual Verification (if needed):**

```
# Check the deployed file structure on Azure
az webapp ssh --name ai-event-planner-saas-py --resource-group
[your-rg]
ls -la /home/site/wwwroot/scripts/
```

Files Modified

- `.github/workflows/azure-deploy-saas.yml` - Enhanced deployment package creation and migration invocation

Files Verified (No Changes Needed)

- `scripts/migrate.py` - Exists and is properly structured
- `alembic.ini` - Properly configured for migrations
- `migrations/` directory - Contains migration files

Next Steps

1. Commit these changes to the repository
2. Push to trigger the GitHub Actions workflow
3. Monitor the deployment in GitHub Actions
4. Verify the migration runs successfully
5. Check the Azure Web App is running correctly

Rollback Plan (if needed)

If issues arise, you can revert to the previous workflow by:

```
git revert [commit-hash]
git push
```

The previous workflow will attempt to run migrations but may fail if the file isn't found.

Related Documentation

- See `AZURE_MIGRATION_DEPLOYMENT_FIXES.md` for the original analysis
- See `MIGRATION_MODULE_FIX_SUMMARY.md` for previous migration fixes
- See `AZURE_DEPLOYMENT_SETUP_GUIDE.md` for complete deployment setup

Date Applied: October 22, 2025
Applied By: Automated Fix
Status: Ready for Testing
Priority: High - Fixes Critical Deployment Issue