AWS_REGION Handling in Lambda

Overview

This document explains how AWS_REGION is handled in the CollaborateMD-Salesforce middleware Lambda function.

The Issue

AWS Lambda considers `AWS_REGION` a **reserved environment variable**. When you try to manually set it in the Lambda environment variables, deployment fails with an error like:

```
InvalidParameterValueException: AWS_REGION is a reserved environment variable
```

The Solution

AWS Lambda **automatically provides** the `AWS_REGION` environment variable to all Lambda functions at runtime. Therefore:
- ✅ **DO NOT** manually set AWS_REGION in Lambda environment variables
- ✅ **DO** use `os.getenv('AWS_REGION')` in your code - it will work automatically

Code Implementation

Our code properly handles this in `src/config.py`:

```
# DynamoDB Configuration (for state management)
DYNAMODB_TABLE_NAME: str = os.getenv('DYNAMODB_TABLE_NAME', 'collaboratemd-sync-state')
DYNAMODB_REGION: str = os.getenv('AWS_REGION', 'us-east-1')
```

How It Works

1. **In Lambda Runtime**:
   - AWS automatically sets `AWS_REGION` environment variable
   - Our code reads it using `os.getenv('AWS_REGION')`
   - Example: If Lambda is deployed in `us-east-1`, `AWS_REGION` will be `'us-east-1'`

2. **In Local Development**:
   - If `AWS_REGION` is not set, it defaults to `'us-east-1'`
   - This allows local testing without Lambda runtime

3. **State Manager Usage** (`src/state_manager.py`):
   `python`
   `self.region = Config.DYNAMODB_REGION`
   `self.dynamodb = boto3.resource('dynamodb', region_name=self.region)`

Deployment Changes

The deployment script (`scripts/deploy_lambda.sh`) has been updated to:
1. ❌ **Exclude** `AWS_REGION` from Lambda environment variables
2. ✅ Include all other necessary environment variables
3. 📝 Add documentation comment explaining this behavior

.env.production File

The `.env.production` file now includes a comment:

```
# AWS Configuration
# AWS_REGION is automatically provided by Lambda runtime - do not set it manually
```

Testing

To verify this works correctly:
1. Deploy Lambda function without AWS_REGION in environment variables
2. Lambda runtime will automatically provide AWS_REGION
3. Code reads it using `os.getenv('AWS_REGION')`
4. DynamoDB client connects to the correct region

References

- AWS Lambda Environment Variables: https://docs.aws.amazon.com/lambda/latest/dg/configuration-envvars.html
- Reserved Environment Variables: https://docs.aws.amazon.com/lambda/latest/dg/configuration-envvars.html#configuration-envvars-runtime