Salesforce Deployment Guide

Authentication Issue Encountered

The automated deployment encountered an authentication error:

```
Authentication failed (code: INVALID_LOGIN): Invalid username, password, security
token; or user locked out.
```

Possible Causes:

1. **IP Restrictions**: Your Salesforce org may have IP restrictions. You need to add your IP to the trusted list.
2. **User Locked**: The user account may be temporarily locked due to failed login attempts.
3. **Security Token**: The security token may have expired or changed.
4. **Password**: The password may have been changed.

Manual Deployment Steps

Option 1: Using Salesforce CLI (Recommended)

1. **Install Salesforce CLI** (if not already installed):
   ```
   bash
      # Already installed at /usr/local/bin/sf/bin/sf
      export PATH=/usr/local/bin/sf/bin:$PATH
   ```

2. **Authenticate to Salesforce**:
   ```
   bash
      cd /home/ubuntu/collaboratemd-salesforce-middleware/salesforce
      sf org login web --set-default --instance-url https://test.salesforce.com --alias Col-
   labMDSandbox
   ```
   This will open a browser window for you to log in.

3. **Deploy the Apex Classes**:
   ```
   bash
      sf project deploy start --source-dir force-app/main/default/classes
   ```

Option 2: Using Workbench

1. **Go to Workbench**: https://workbench.developerforce.com/
2. **Login** with your Salesforce credentials
3. Select **Environment**: Sandbox
4. **Deploy** → **Deploy**
5. **Upload** the deployment package:
   - File: `/home/ubuntu/collaboratemd-salesforce-middleware/salesforce/force-app/main/default/classes`
   - Select “Rollback On Error”
   - Select “Run All Tests” (for production)

6. Click **Deploy**

Option 3: Using Visual Studio Code with Salesforce Extensions

1. **Open VS Code** with Salesforce Extensions installed
2. **Authorize** your Salesforce org:
   - Ctrl+Shift+P → "SFDX: Authorize an Org"
3. **Deploy** the classes:
   - Right-click on `force-app` folder → "SFDX: Deploy Source to Org"

Required Metadata Components

1. Apex Classes

Located in: `/home/ubuntu/collaboratemd-salesforce-middleware/salesforce/force-app/main/default/classes/`

- **CollabBatch.cls**: Main batch class that fetches claims from CollaborateMD API
- **ColborateMDRes.cls**: Response wrapper class for API responses

2. Custom Objects (Must exist in your Salesforce org)

The Apex classes reference these custom objects. Ensure they exist:

- **Services_Authorization__c**
- Fields: Start_Date__c, End_Date__c, Level_of_Care__c, Authorization_Number__c, Related_Patient__c, MR_Number__c

- **Claims__c**

- Fields: Claim_Number__c, Claim_Payor__c, DOS__c, DOS_End__c, Claim_Submitted_Date__c, Charged_Amount__c, Total_BDP__c, Paid_Amount__c, EFT_or_Paper_Check__c, Paid_Date__c, Related_Services_Authorization__c, Payer__c, MR_Number__c, Paid_Y_or_N__c, LOC__c, Insurance_Authorization_Number__c, ServiceAuth_Record_ID__c, Related_Patient__c

- **Claim_Payor__c**

- Fields: Name

3. Named Credential Setup (Critical!)

After deploying the Apex classes, you MUST create a Named Credential:

1. **Setup** → **Named Credentials** → **New Named Credential**
2. **Configure**:

```
Label: Claims API
   Name: Claims_API
   URL: [Your CollaborateMD API endpoint - Ask your CollaborateMD administrator]
   Identity Type: Named Principal
   Authentication Protocol: Password Authentication
   Username: nelser
   Password: May052023!@#$%%
   Generate Authorization Header: ✓
   Allow Merge Fields in HTTP Header: ✓
   Allow Merge Fields in HTTP Body: ✓
```

3. **Save**

4. Remote Site Settings

1. **Setup** → **Remote Site Settings** → **New Remote Site**
2. **Configure**:
   `Remote Site Name: CollaborateMD_API`
   `  Remote Site URL: [Your CollaborateMD API endpoint]`
   `  Active: ✓`

5. Schedule the Batch Job

After deployment, schedule the batch job to run automatically:

Open Developer Console → **Debug** → **Open Execute Anonymous Window**

```
// Schedule to run daily at 2 AM
CollabBatch batch = new CollabBatch();
String sch = '0 0 2 * * ?'; // Daily at 2 AM (cron expression)
System.schedule('CollaborateMD Claims Sync', sch, batch);
```

Or run manually:

```
// Run immediately
CollabBatch batch = new CollabBatch();
Database.executeBatch(batch, 200);
```

Testing the Deployment

1. **Run Tests** (if you have test classes):
   `apex`
   `  Test.startTest();`
   `  CollabBatch batch = new CollabBatch();`
   `  Database.executeBatch(batch, 1);`
   `  Test.stopTest();`

2. **Check Batch Status**:
   - Go to **Setup** → **Apex Jobs**
   - Look for “CollabBatch” jobs
   - Check the status and any errors

3. **Verify Claims Creation**:
   `sql`
   `  SELECT Id, Name, Claim_Number__c, DOS__c, Paid_Amount__c`
   `  FROM Claims__c`
   `  ORDER BY CreatedDate DESC`
   `  LIMIT 10`

Troubleshooting

Authentication Issues:

- **Check IP Restrictions**: Setup → Security → Network Access → Add your IP

- **Reset Security Token**: Setup → My Personal Information → Reset My Security Token
- **Unlock User**: Setup → Users → find user → Unlock

Deployment Errors:

- **Missing Custom Objects**: Create them first before deploying Apex
- **Missing Fields**: Add all required custom fields to the objects
- **Named Credential**: Must be created for API callouts to work

Runtime Errors:

- **Check Debug Logs**: Setup → Debug Logs
- **Check Apex Jobs**: Setup → Apex Jobs (for batch job status)
- **Check API Endpoint**: Verify the Named Credential URL is correct

Next Steps

After successful Salesforce deployment:
1. ✅ Deploy AWS Lambda function (separate step)
2. ✅ Configure Lambda with Salesforce credentials
3. ✅ Test end-to-end integration
4. ✅ Monitor batch jobs and Lambda executions

Support

If you encounter issues:
1. Check Salesforce Debug Logs
2. Verify all custom objects and fields exist
3. Ensure Named Credential is properly configured
4. Check API endpoint accessibility