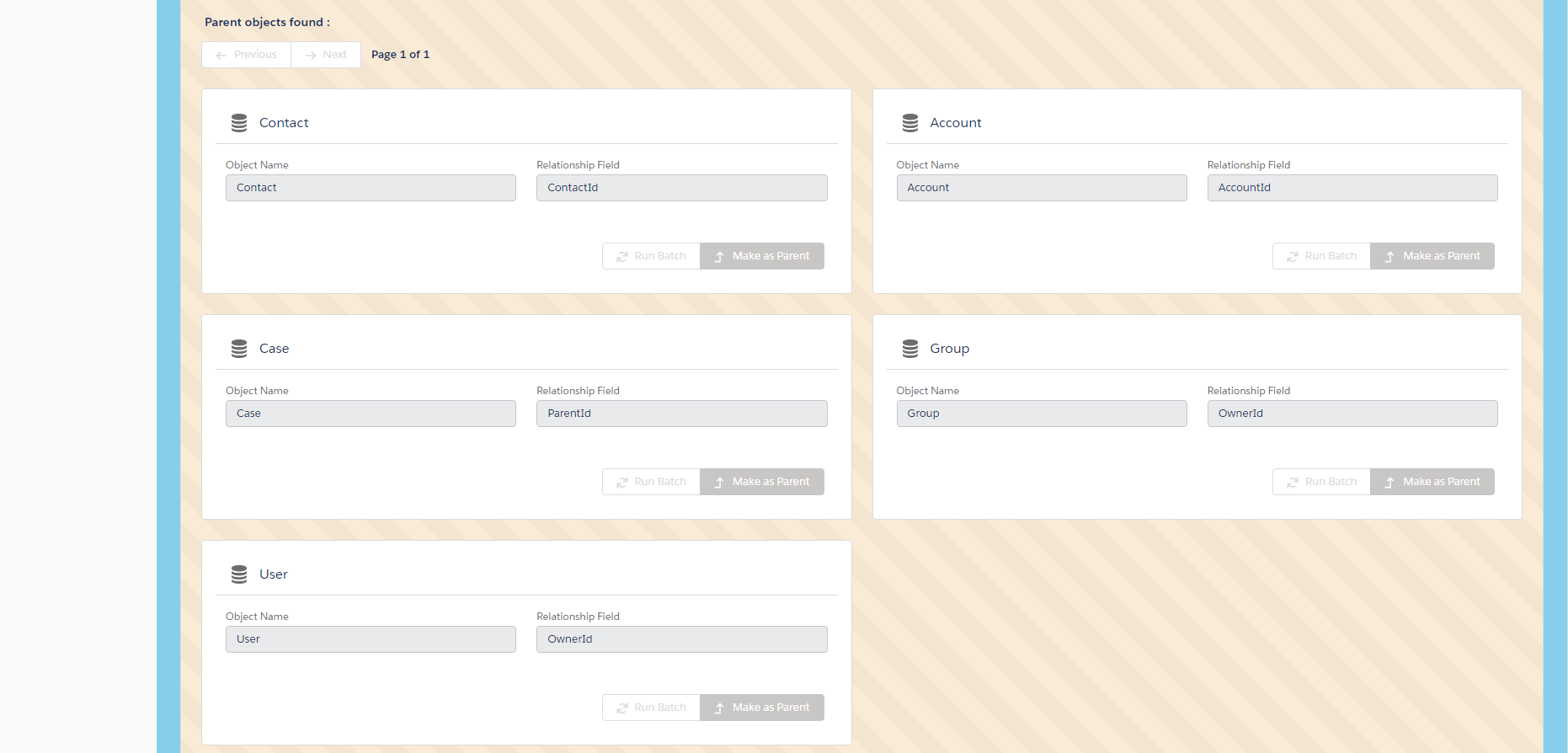
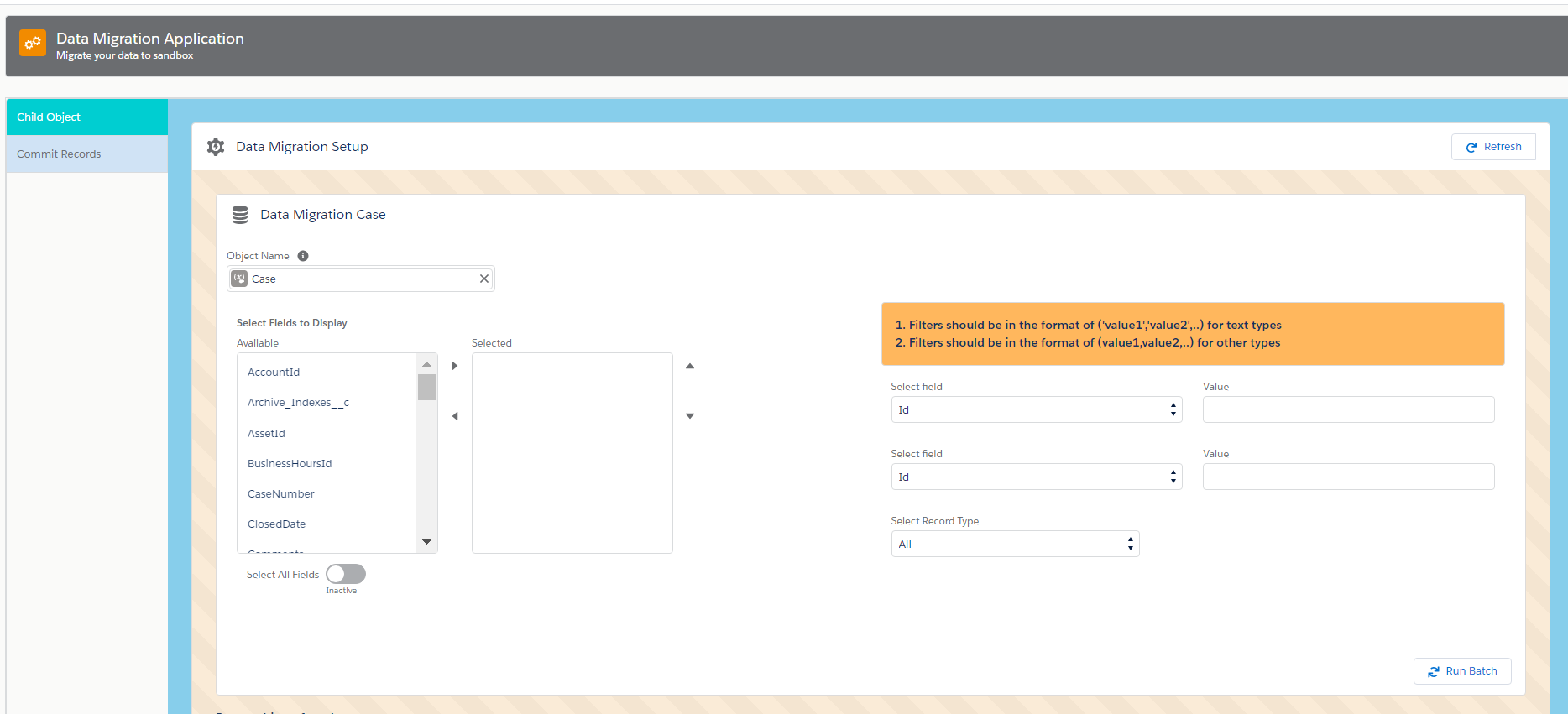
**Post Installation Steps:**

**Application overview:**

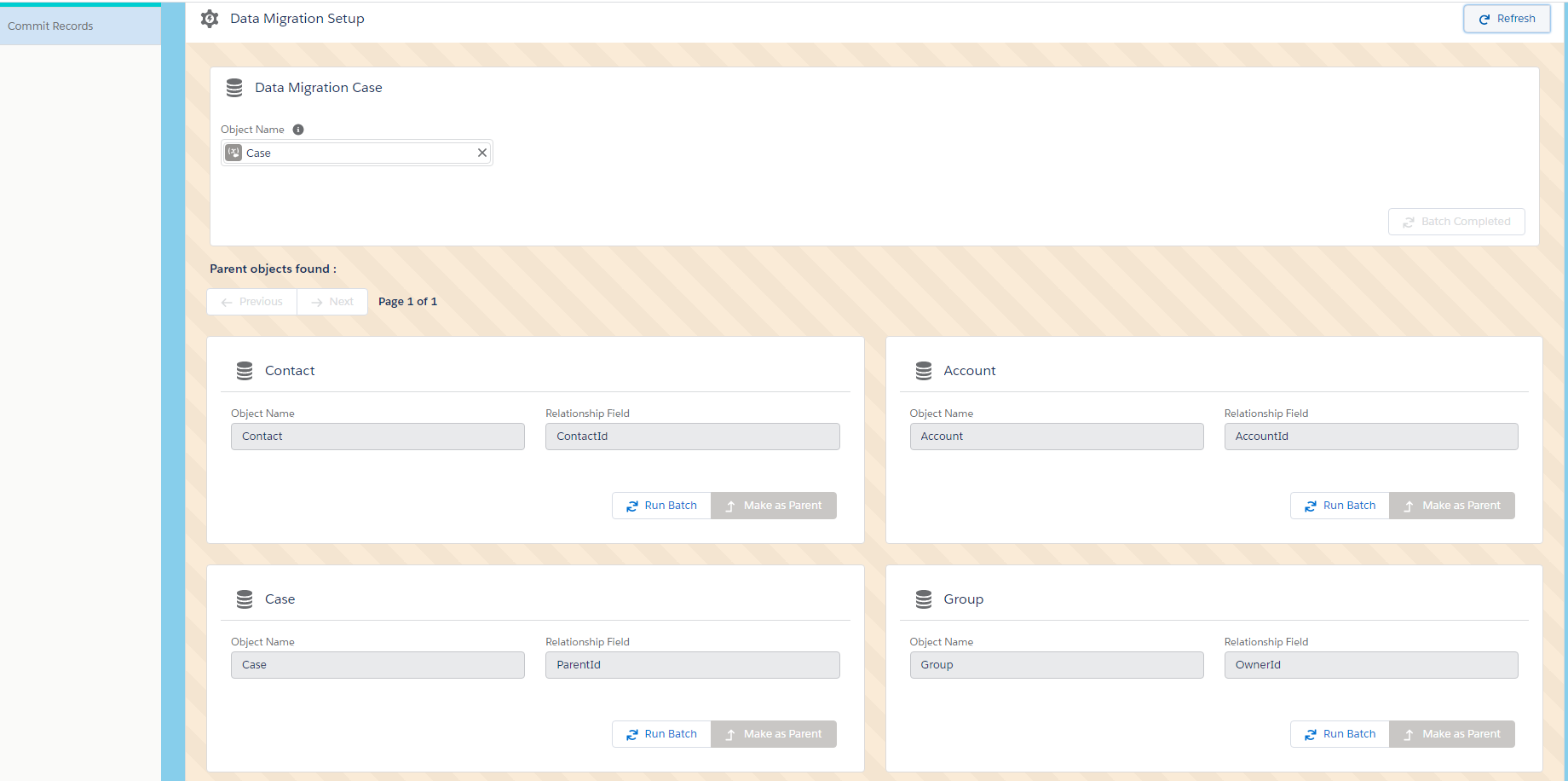
This application is used for migrating standard and custom object records from one salesforce org to another along with all parent object records in the hierarchy. It also provides a save and commit mechanism where the users can first run jobs to move the records to target org and then commit them to database when all the records in the hierarchy are moved by specifying the commit order.



1. User selects the object to migrate and saves the settings using ‘Save Settings’ button.
2. Once saved, there will be settings created for child parent objects as well which will be disabled until the job for the child object is run. They can select the fields to be migrated and provide filters for record type and 2 additional fields which are to be in the format of

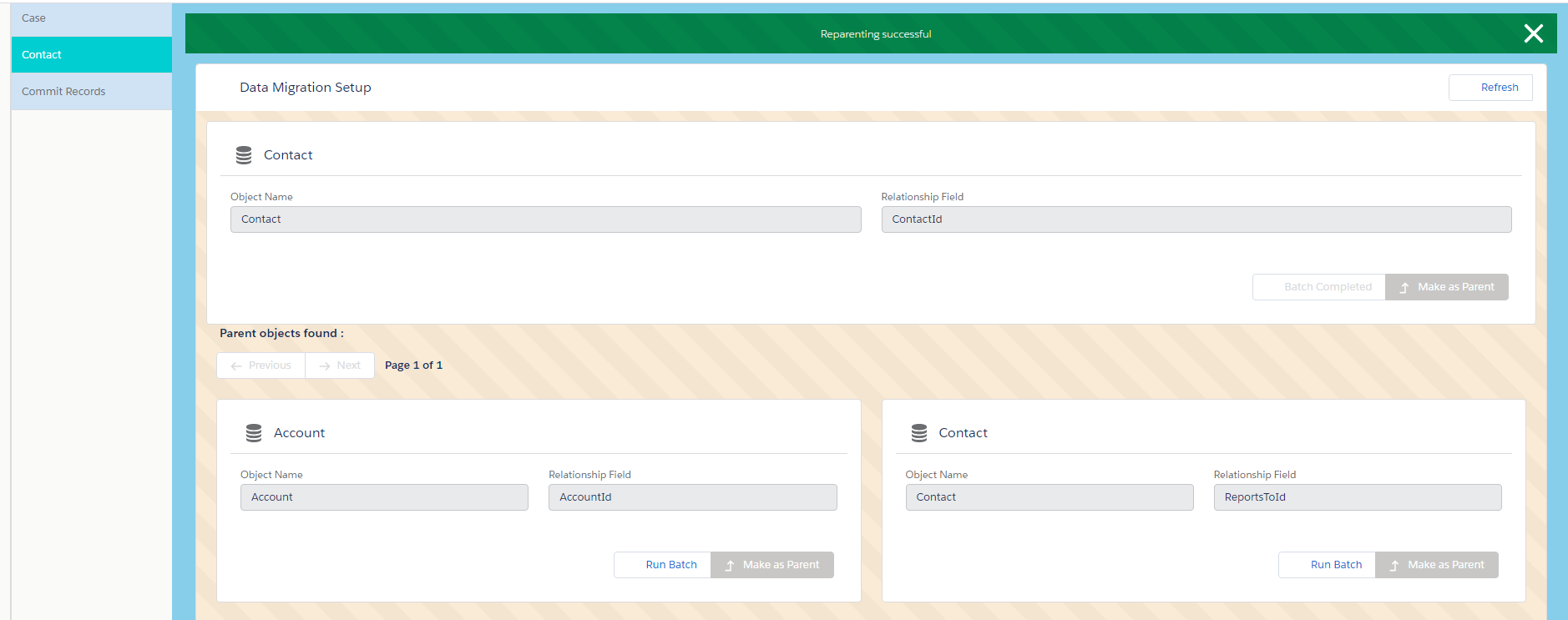
fieldname\_\_c=(‘fieldValue 1’ ,fieldValue 2’…)

1. The users then click on the ‘Run Batch’ button to start running the batch. This will keep the buttons disabled. Users can click on the refresh button to check the status of the batch whenever required.
2. Once the job is run the buttons ‘Run batch’ will be disabled on the child and will be enabled for the parent objects to run the batch for them.

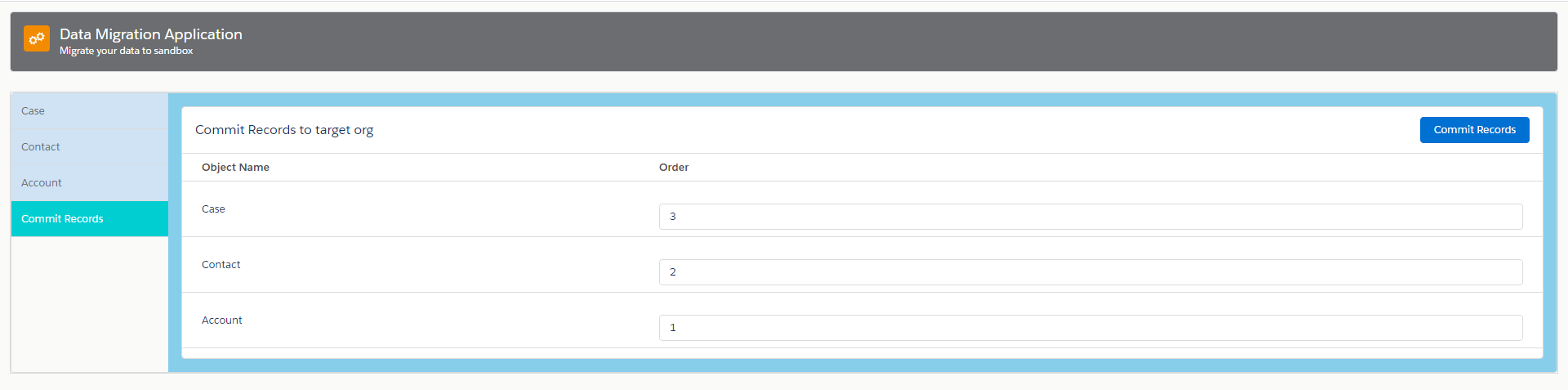


1. Users can run the batch for the parent objects and once it is completed, the ‘Make as Parent’ button will be enabled to make any parent object as the new child object in a different tab (this will remove the reparented parent object from the Parent objects list of the initial selected object which in this case is ‘Case’ object’) where the parent objects of this new child object will be visible in the same manner.

Ex: if Case Records have to be moved, once the case batch is run the parent Contact batch is run, then the account and so on. Then we make the Contact Object as the new child and the parents of Contact which are Account etc will be shown in the ‘Parent Objects found’ section. This can be done until we traverse the whole Schema of objects. Case -> Contact -> Account -> …

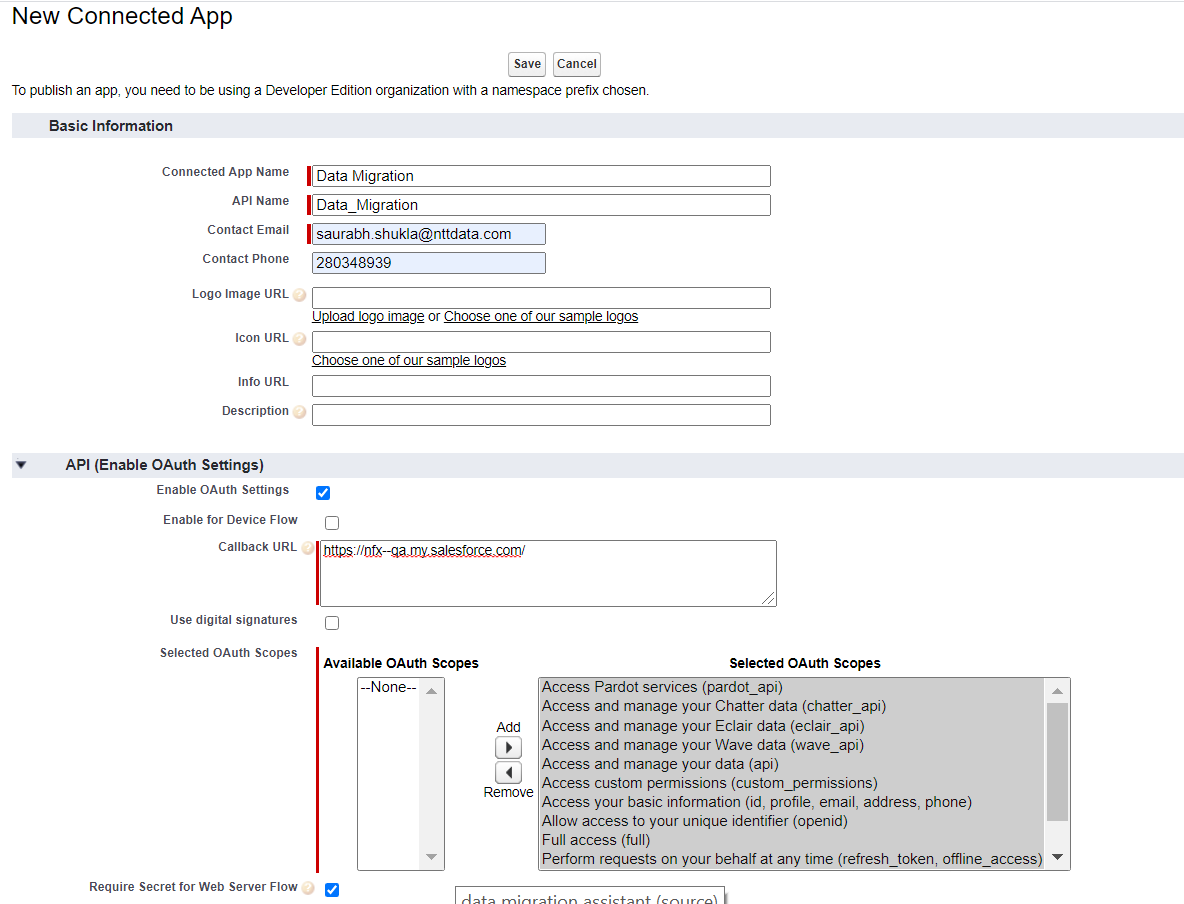


1. Once the batches are run for all the objects, we can go to the commit records tab and commit the records in the target org in the proper order. Parent records should be inserted first, so we start giving the order from parent and then to child. As seen in the below screenshot. The Account is given the order as 1 which means it should be committed first.

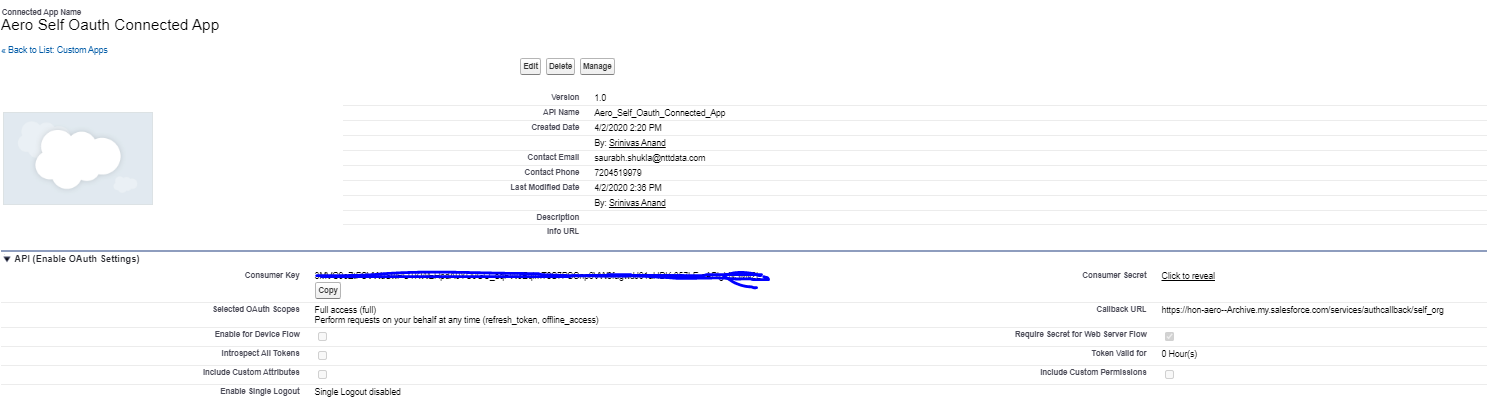


**Connected App setup:**

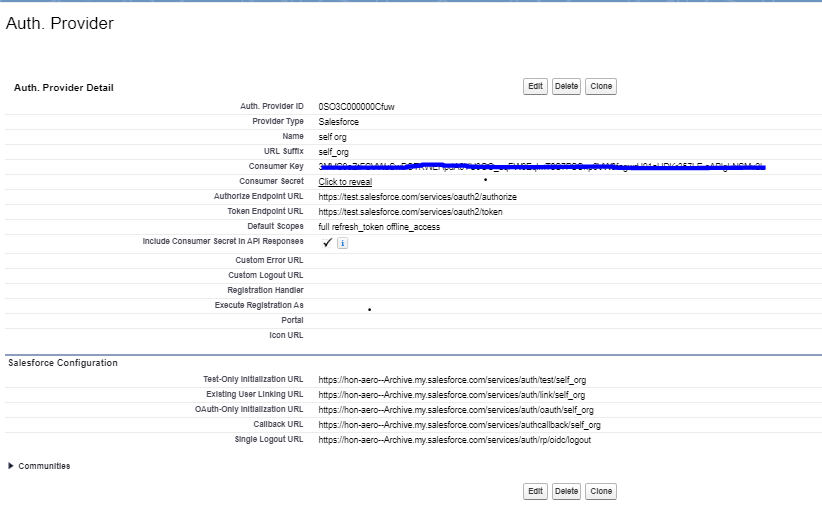
1. Create a connected app in the **target org** with oauth enabled and below settings. Callback url can be the source org for now and select all oauth scopes.



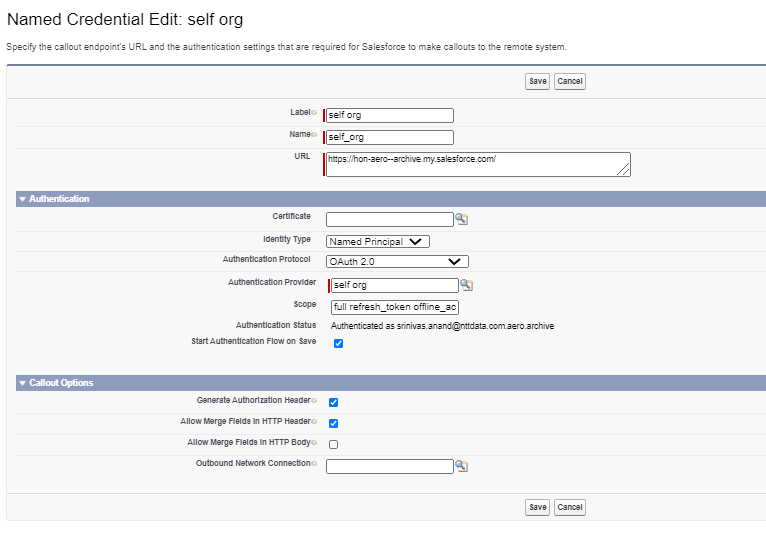
1. Save the app.



1. Create a auth. Provider in the **source org** with salesforce settings. Copy the client Id and client secret from the above created connected app and paste it in the auth provider.



1. Save the auth. Provider and you will find a callback url in the related list. Copy and paste the callback url in the connected app callback url.
2. Create a named credential in the **Source Org** use the Named principal authentication and oauth 2.0 option.
3. Lookup the auth provider that was created and use the same in the named credential.
4. Allow merge fields in the Http header, generate authorization header, start authentication flow on save should be checked.



1. Authorize the named credential on Save.

**Target Org:**

Create a custom setting record in Data\_migration\_index\_\_c with below details:

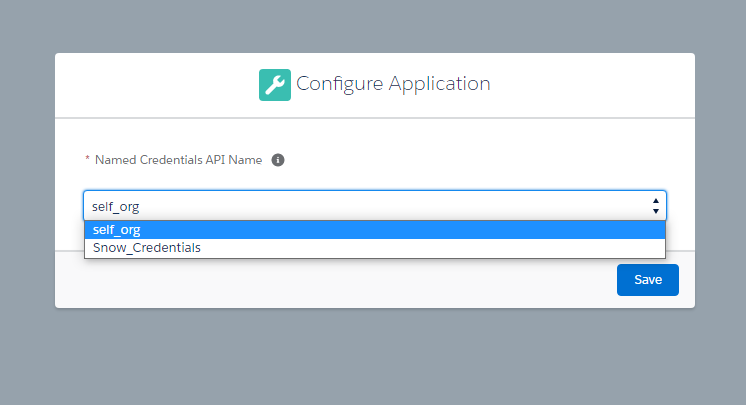
Name- migration

indexNumber- 1

Record Id - use the record Id of the same record after saving.

**Source Org**

On the first time opening the app, you will see the below screen.



Select the named credential that was created before and Save. This named credential will be used in all the callouts made to the target org.