

SDC Cassandra Data Migration

- [Introduction](#)
- [Migration Steps](#)
 - [Memtable flush](#)
 - [Copy data from Amsterdam to Casablanca](#)
 - [Deploy SDC on Casablanca](#)
 - [Create Pod for Data Migration](#)
 - [Run Data Migration Script](#)
 - [Redistribute Migrated Services](#)

Introduction

This document contains the steps required to migrate the SDC Cassandra database data from Amsterdam environment to the Casablanca environment.

Migration Steps

Memtable flush

Execute nodetool flush utility on the Amsterdam's Cassandra instance to flush the memtables to disk:

- Connect to SDC Amsterdam Environment
- Go inside 'sdc-cs pod'
- Execute the command: 'nodetool flush'

Copy data from Amsterdam to Casablanca

Copy the Cassandra data directory from Amsterdam to Casablanca environment:

- Source: /dockerdata-nfs/<namespace>/sdc/sdc-cs/CS/data
- Destination: /dockerdata-nfs/<Release name>-sdc/sdc/sdc-cs/CS/

Note: Replace /dockerdata-nfs with real nfs path.

Deploy SDC on Casablanca

Deploy SDC on the Casablanca environment with SDC Cassandra instance configured to use the above mentioned destination directory having the Amsterdam data.

Create Pod for Data Migration

Migration project exists inside the 'sdc-cs-config' pod which is by default deployed as a kubernetes job. Hence, this pod will not be available after the job execution. So, as a workaround, create another pod using the 'sdc-cs-config' image to run the data migration script after the SDC Casablanca deployment is done.

After pod is created, make below changes in the chef code to fix the generated 'elasticsearch.yml' file:

- Go inside newly created 'sdc-cs-config' pod
- Fix the '03-schemaCreation.rb' file (by adding the below [highlighted](#) text) inside /root/chef-solo/cookbooks/cassandra-actions/recipes:

```
variables({  
  
  :elastic_ip => node['Nodes']['ES'].first  
  
})
```

- Run the startup script:
bash-4.4# cd /root

```
bash-4.4# ./startup.sh
```

Run Data Migration Script

Run the migration tasks by executing the 'sdc-migration.sh' script:

```
bash-4.4# cd /tmp/sdctool/scripts/
```

```
bash-4.4# chmod u+x sdc-migration.sh
```

```
bash-4.4# ./sdc-migration.sh -c /tmp/sdctool/config/
```

Note:

Some of the migration tasks may fail in case some properties are missing inside '*configuration.yaml*' file under '/tmp/sdctool/config'.

In our case we saw failure inside '*ForwardPathMigration.java*', '*ResourceLifecycleMigration.java*', '*UpgradeMigration1710.java*' due to some missing properties.

A null check can be added in the above migration tasks to skip them and copy the updated/fixed '*asdctool-1.3.5-jar-with-dependencies.jar*' under '/tmp/sdctool' on 'sdc-cs-config' pod. You can also refer to IntelliJ Idea patch ([migration_fix.patch](#)) for the fixes done to the above mentioned migration tasks.

Redistribute Migrated Services

After migrating the data, you will see the migrated services in Casablanca's SDC portal. Redistribute the services to make them available to other ONAP components.