**#ISSUES**

**#ISSUE 1**

**pricing in sbx : docdb , index manually create**

* Trigger the Pricing Export Jenkins Job from below URL :

https://devadmin.dev.ssi-cloud.com/view/Migration/job/PricingRefresh\_Export/

* Trigger the Pricing Import Jenkins Job from below URL :

<https://devadmin.dev.ssi-cloud.com/view/Migration/job/PricingRefresh_Import/>

* docdb -- shared dbs - connect to instance 10.3.50.156
* Copy Mongodump from S3 , Unzip it and do MongoRestore as eg. app.pricings.new s3://avaloncms/dellPricingRefresh/mysqlExport/exportedMYSQLDump.tar.gz

s3://avaloncms/PricingData/201908221245/test\_201908221245.zip

s3cp s3://avaloncms/PricingData/201909051305/test\_201909051305.zip .

* unzip test\_201909051305.zip &
* connect to docdb :

mongo --ssl --host shared-docdb.cluster-cpbvd3ebtcjx.us-east-1.docdb.amazonaws.com:27017 --sslCAFile rds-combined-ca-bundle.pem --username admin --password welcome

* Check whether data and index has been created or not in "app.pricings" collection in "test" database.

show databases ;

use automation\_dell

show collections ()

db.app.pricings.drop() “always drop this collection before restoring ”

* mongo restore :

nohup mongorestore --ssl --host shared-docdb.cluster-cpbvd3ebtcjx.us-east-1.docdb.amazonaws.com:27017 --sslCAFile rds-combined-ca-bundle.pem --db automation\_dell /mnt/workspace/pr/test/app.pricings.bson --username admin --password welcome --collection app.pricings --noIndexRestore >> SBX\_DocDB\_Pricing\_Restore.log &

* check logs :

tail -f SBX\_DocDB\_Pricing\_Restore2.log

**39578194 : records**

* create index on mongo :

nohup mongo sbx\_configdata\_frb5 --ssl --host shared-docdb.cluster-cpbvd3ebtcjx.us-east-1.docdb.amazonaws.com:27017 --username admin --password welcome --sslCAFile rds-combined-ca-bundle.pem -eval "var dbList=['dell#automation\_dell#admin#welcome'],auxDB='sbx\_configdata\_frb5', isDocumentDb =true" metadataIndexes.js >> createAppPricingsIndex.log &

* delete.new file : db["app.pricings.0713"].drop()
* verify indexes : db.app.pricings.getIndexes()

**#ISSUE 2**

**PRICING REFRESH :sbo and sqa**

* sbo and sqa : on mongo itself no manual index creation
* Trigger the Pricing Export Jenkins Job from below URL :

https://devadmin.dev.ssi-cloud.com/view/Migration/job/PricingRefresh\_Export/

* Trigger the Pricing Import Jenkins Job from below URL :

<https://devadmin.dev.ssi-cloud.com/view/Migration/job/PricingRefresh_Import/>

* launch a idx instance – sbx stk1 - get the publc ip
* df -h
* dwnload zip from s3Copy Mongodump from S3 , Unzip it and do MongoRestore as eg.

s3cp s3://avaloncms/PricingData/201908291248/test\_201908291248.zip .

unzip test\_201908011237.zip &

* Mongo Credientials:

mongo

use admin

db.auth("devops", "Ipk9b3Wro6PG7zdKfMsJ")  
 db.auth("devopsUserAdmin", "vP4MehogTEpSO6Wp4875")

prod  
 devopsUserAdmin: fqkOWvJxCZ3xqx7pC6gu  
 devops: TSYby5rSNyYuPnv4J1zc  
 db.auth("devops", "TSYby5rSNyYuPnv4J1zc")  
 db.auth("devopsUserAdmin", "fqkOWvJxCZ3xqx7pC6gu")  
 db.auth("admin", "welcome")  
 db.auth("appuser\_ciscoinc", "cyueri9O")  
 db.auth("appuser\_dell", "adth78hj")

db.auth("devops", "TSYby5rSNyYuPnv4J1zc")  
 db.auth("devopsUserAdmin", "fqkOWvJxCZ3xqx7pC6gu")

* connect to mongo :

db.auth("devops", "Ipk9b3Wro6PG7zdKfMsJ")

mongo --ssl --username admin --password welcome

* Check whether data and index has been created or not in "app.pricings" collection in "test" database.

show databases ;

use automation\_dell

show collections ;

db.app.pricings.count() 39522532: 39566334 old:new records

* mongo restore :

run mongo restore cmd from where we have .bson file :

nohup mongorestore --collection app.pricings\_new -u admin -p welcome --db automation\_dell app.pricings.bson &

* check logs :

tail -f nohup.out

* take backup of existing app.pricings

db.app.pricings.renameCollection("app.pricings\_backup\_30\_08\_2019");

* rename app.pricings\_new to app.pricings

db.app.pricings\_new.renameCollection("app.pricings");

* delete a collection

db.app.pricings.0713.drop()

* check collections : show collections ;
* get collection count : db.app.pricings.count()
* verify the newly created collections and records :

confmgt – dqi – take string n open in ui

<http://sqa-appsvc1-dqi-stk3.int.dev.ssi-cloud.com/> : check with pricecheckjson.txt

**#ISSUE 3**

**PRICING REFRESH :pe / if asked to clone app.pricings as app.pricings2 do restore after alunching idx**

* pe : on mongo itself no manual index creation
* Trigger the Pricing Export Jenkins Job from below URL :

https://devadmin.dev.ssi-cloud.com/view/Migration/job/PricingRefresh\_Export/

* Trigger the Pricing Import Jenkins Job from below URL :

<https://devadmin.dev.ssi-cloud.com/view/Migration/job/PricingRefresh_Import/>

* launch a idx instance –pe idx through ec2 instance
* login to ip
* df -h
* dwnload zip from s3Copy Mongodump from S3 , Unzip it and do MongoRestore as eg.

s3cp s3://avaloncms/PricingData/201909051305/test\_201909051305.zip .

s3://avaloncms/PricingData/201908291248/test\_201908291248.zip

unzip test\_201908011237.zip &

* connect ot mongo using : Mongo Credientials:

mongo

use admin

db.auth("devops", "Ipk9b3Wro6PG7zdKfMsJ")

* connect to mongo using : command

mongo --ssl --username admin --password welcome

* Check whether data and index has been created or not in "app.pricings" collection in "test" database.

show databases ;

use automation\_dell

show collections ;

db.app.pricings.count() 39566334 : 39656288

old:new records

* mongo restore :

run mongo restore cmd from where we have .bson file :

nohup mongorestore --collection app.pricings2 -u appuser\_dell -p adth78hj --db automation\_dell app.pricings.bson &

* check logs :

tail -f nohup.out

* take backup of existing app.pricings

db.app.pricings.renameCollection("app.pricings\_backup\_06\_09\_2019");

* rename app.pricings\_new to app.pricings

db.app.pricings.new.renameCollection("app.pricings");

* delete a collection

db.app.pricings.0713.drop()

* check collections : show collections ;
* get collection count : db.app.pricings.count()
* verify the newly created collections and records :

confmgt – dqi – take string n open in ui

[http://pe-dqi-rs.int.dev.ssi-cloud.com](http://pe-dqi-rs.int.dev.ssi-cloud.com/)

* to check that a collection is sharded or not :

db.app.pricings.getShardDistribution()

**#ISSUE 4**

**TO CREATE NEW MONGO DB ON PE**

* FOR MONGO : LAUNCH A STACK : PE-IDX :

bin/orca deploy pod-component V2Dev PE IDX rs --capabilities CAPABILITY\_NAMED\_IAM --dry-run

* LOGIN TO INSTANCE : pe-idx (stack -- instance )
* MONGO BACKUP

take backup of existing app.pricings

db.app.pricings.renameCollection("app.pricings\_backup\_30\_08\_2019");

rename app.pricings\_new to app.pricings

db.app.pricings\_new.renameCollection("app.pricings");

* MONGO DB CREATE

use automation\_app : create new mongo db

create new collection : db.createCollection("test")

use admin

db.auth("devopsUserAdmin", "vP4MehogTEpSO6Wp4875")

use automation\_app

create new user in db

db.createUser({user: "admin", pwd: "welcome", roles: [{role: "readWrite", db: "automation\_app"}]});

create new user in db

db.createUser({user: "reporting", pwd: "password123", roles: [{role: "readWrite", db: "automation\_app"}]});

grant roles :

db.grantRolesToUser("reporting",[ "read" , { role: "read", db: "automation\_app" } ])

use admin

db.Getusers() : list users in a db

* MONGO RESTORE

mongo restore for automation\_app db , run from test folder

nohup mongorestore --collection app.pricings -u reporting -p password123 --db automation\_app app.pricings.bson &

tail -f nohup.out

**#ISSUE 5**

**TO CREATE NEW POSTGRESS DB ON PE**

* create new role and new db on postgress
* postgress dump/backup :
* git ----configmgt:for pe rs pe.rs.rs.yaml ---- pe-pdb.int.dev.ssi-cloud.com
* login and hit the commands :

corev2dev : dev slave1 machine login : ssh -p 2222 ec2-user@10.3.40.189

* take workflow db dump

pg\_dump -h pe-pdb.int.dev.ssi-cloud.com -p 5432 -U workflow pe\_workflowdb\_2\_0 > workflowdb\_dump

* take scheduler db dump

pg\_dump -h pe-pdb.int.dev.ssi-cloud.com -p 5432 -U scheduler pe\_schedulerdb\_2\_0 > scheduler\_dump

* login to pgadmni : pe-pdb.int.dev.ssi-cloud.com
* create db

CREATE DATABASE pe\_schedulerdb\_etl WITH ENCODING = 'UTF8' OWNER = scheduler;

* restore postgres db : workflow

psql -h pe-pdb.int.dev.ssi-cloud.com -p 5432 -U scheduler pe\_schedulerdb\_etl < scheduler\_dump

* restore postgres db :

psql -h pe-pdb.int.dev.ssi-cloud.com -p 5432 -U scheduler pe\_schedulerdb\_2\_0< scheduler\_dump

* drop databases :

DROP DATABASE sbx\_workflowdata\_stk1;

* create db and grant permissioin :

CREATE DATABASE sqa\_schedulerdata\_stk1;

GRANT ALL PRIVILEGES ON DATABASE sqa\_schedulerdata\_stk1 to scheduler;

ALTER DATABASE sqa\_schedulerdata\_stk1 OWNER TO scheduler;

**#ISSUE 6**

**emr cluster with autoscaling on sbx-stk1**

jenkins – build – emr- launch wd autoscaling - sbk stk3 – build

check on aws – emr – active cluster – sbx-stk3 should be there

**#ISSUE 7**

**pe-etl cluster upscale**

nodes increase , core

aws- emr- active cluster – hardware – increase nodes

**#ISSUE 8**

**sbx-sqa env upgrade on 1.3.8**

* copy binaries :
* rundeck job : promote repo binaries (env same )
* (env not same ) chef- endpoint- pods.json – devcertified changed to qa – knife upload , - chef client
* deploy using jenkins :#!/bin/bash
* printf "\n\n getting cluster id"
* id=$(cat /mnt/var/lib/info/job-flow.json | grep jobFlowId | cut -f2 -d: | cut -f2 -d'"')
* printf "\n\n id is : $id"
* aws emr describe-cluster --cluster-id $id --region us-east-1 > clusterdetails.json
* clusterName=$(cat clusterdetails.json | jq -r '.Cluster.Name' clusterdetails.json)
* clusterNamee=$(echo "$clusterName" | tr -d '"')
* printf "\n clusterName : $clusterNamee"
* caseNumber=$(cat clusterdetails.json | jq --raw-output '.Cluster.Tags[4].Value' clusterdetails.json)
* caseNumberr=$(echo "$caseNumber" | tr -d '"')
* printf "\n caseNumber : $caseNumberr"
* caseOwnerName=$(cat clusterdetails.json | jq --raw-output '.Cluster.Tags[0].Value' clusterdetails.json)
* caseOwnerNamee=$(echo "$caseOwnerName" | tr -d '"')
* printf "\n caseOwnerName : $caseOwnerNamee"
* ClusterName=$clusterNamee
* printf "\n clusterName : $clusterName"
* caseNumber=$caseNumberr
* printf "\n caseNumber : $caseNumber"
* caseOwnerName=$caseOwnerNamee
* printf "\n caseOwnerName : $caseOwnerName"
* aws ses send-email --region us-east-1 --from "tdixit@servicesource.com" --destination "ToAddresses=anamika.nagar@impetus.co.in" --message "Subject={Data=EBS modified,Charset=UTF-8},Body={Text={Data=heyyy$
* echo "Mail sent to : $caseOwnerName , as the volume on the cluster is modified Volume by 50%"
* deploy- sqa-stk3- allservicesdeploy

**#ISSUE 9**

* **stop config deploymnt on sbx-stk1**

job – deploy – sbx-stk1- allservices – deploy – commnt

**#ISSUE 10**

**Access to kafka topics through consumers**

* temorary add :

opsBOARD -shared -kafkabroker – instance CHOOSE

cd .ssh/

cd /home /ssiro

add ssh keys : authorizaedkeys.sh

git- orca- scripts-ssiavlon-sshkeymgt

* permanently ADD :

ssh key add : sshkeymgt\_v2.sh

id add

**#ISSUE 11**

**deploymnt of packager on sbx-stk5**

* packager means implementation AND for dell – imp job for dell
* JENKINS - deploy sbx- stk5- implementation job – configure

tenant-dell – build

* jenkins job – deploy sbx-stk5/emr-deploy-stk5
* s3 loc : s3/bucket/srev-emr-test/sbx/stk5/repo/app
* git loc : ssiavalon/rs-etl

**#ISSUE 12**

**deploy config mgt on sbx-stk5**

configmgt means zk

deploy/sbx-stk5/

**#ISSUE 13**

**backup of prism\_case delta table**

* backup means – to create a b and copy tables into it
* master ssh 22 hadoop@<ip>
* jenkins ob : emr/hivedbintegration/ job run to cretae db
* source db ; orca sbx-stk5 frb5-dell-prism
* target db : case+ticketno
* cluster launch : emr/launch clusterwd spot fleet
* logs check : master : hive migration logs
* (orignal db )hive – show databased- sbx-stk-frb5-dell-prism
* select\* from prism\_case\_delta
* (backup db ) : hive – usecase\_caseno – show databases
* show create table prism\_delta\_case

**#ISSUE 14**

**metastore update**

* cmd :

analyse <table case..> <prism\_case\_delta> partition (ds).computestatistics

**#ISSUE 15**

**add steps in emr cluster**

* cluster : emr – sbx-stk5 dataservices (loc)
* step add : case\_step\_3.2 : name
* loc : s3 path/...hql
* step : hive program
* arg : --hive var dbname =”sbxstk5 frb5 dell prism” --hive var table suffix=” “ hiveconf hive.cli.error.ignore= true
* add step – success

**#ISSUE 16**

**refresh case view and chl col name in table**

* category\_c and country\_formula c in prismcase\_delta table
* show databases p\_C\_D ;
* loc : smr-erv-test/sbx/stk5/ ds/hive/warehouse/sbx-stk-frb-dellprism.db/ prism\_case.delta

**#ISSUE 17**

**register avro enrich schema**

* kafkaschema server – kafkaschreg
* ip : ssh ..
* jenkins -build -dev – implementation – schreg-srgs-dev-ci
* put the tar at : ec2-user/root (ip of kafka instance )
* hit command on entities (avro schema register)

**#ISSUE 18**

**create streamx connecter : for communication witihn kafka services**

* del :
* instance – kafkaconnecttst – ip – shared -
* ip:8083/connectors
* delete kafka connectors
* var/lib sfdc-toolbin/run /sfdctool.sh - delete stream
* connector kafka connector = <url>
* kafka connector = <connector name >

**#ISSUE 19**

**create a connector :**

* cmd – create streamx connectors
* streamx- connector – sbx-frb5- prism-dell
* ui : <ip>:8083/ connector /streamx

**#ISSUE 20**

**automation : cluster idle alert script :**

Hi Shargunani,

PE-ETL-DataServices Cluster is idle since last 1 hr.

Cluster will be terminated in next 30 min if it's still idle.

Case- 00437396

Regards,

Ops Team

#!/bin/bash

set -x

**Symentc versioning of orca on all env** caseOwnerName='Anamika'

clusterName='PE-ETL-DataServices'

caseNumber='00437396'

aws ses send-email \

--from "tdixit@servicesource.com" \

--destination "ToAddresses=mkhare@servicesource.com" \

--message "Subject={Data=Cluster Idle Alert,Charset=UTF-8},Body={Text={Data=heyyyyya,Charset=UTF-8},Html={Data=<!DOCTYPE html><html><body><p>Hi $caseOwnerName&#44;</p><p>$clusterName Cluster is idle since last 1 hr.Cluster will be terminated in next 30 min if it is still idle.</p><p>Case-$caseNumber</p><p style="line-height:0.6">Regards&#44;</p><p style="line-height:0.6">Ops Team.</p></body></html>,Charset=UTF-8}}" \

**#ISSUE 21**

**automation : KUBERNETES SCRIPT TO FETCH STATUS , IMAGE ID , AND NAMESPACE FOR ALL THE PODS**

#!/bin/bash

pods=`kubectl get pods |awk '{print $1}'|awk '{if(NR>1)print}'| xargs`

echo "pods and there status are: $name"

jq -R '[ name= kubectl describe pod $pods > pods.txt ]'

grep -e "^\\Name" -e "^\\Status" -e Image -e "^\\Age" pods.txt

**#ISSUE 22**

**sqa stk3 upgrade on 1.3.8 for all services**

* jenkins :build job : check sqa stk3 : for recent services (changes in the followng services)
* rundeck : promote binaries : rundeck : rs 1.3.8 ise local to rs 1.3.8 deevcertified local
* artifactory : check on artifactory , new reppos
* jenkins : deploy : all services deploy

**#ISSUE 23**

**pe rs upgrade on 1.3.8 for all services**

* jenkins :build job : check sqa stk3 : for recent services (changes in the followng services)
* rundeck : promote binaries : rundeck : rs 1.3.8 ise local to rs 1.3.8 deevcertified local
* artifactory : check on artifactory , new reppos
* jenkins : deploy : all services deploy
* if service fauls with error :
* like service c2r , crw , dqs not started :
* service not started on any instance
* --- sudo monit summary
* --- sudo monit reload
* sudo monit stop dqi
* sudo monit start dqi
* sudo monit staus dqi
* through init.d :
* ps -ef|grep nginx
* check rpm is present or not : rpm -qa |grep ssi
* sudo /etc/init.d/nginx stop
* sudo /etc/init.d/nginx status
* sudo /etc/init.d/nginx start
* sudo /etc/init.d/nginx status

**#ISSUE 24**

**create streamx connectors on pe :**

* KAFKACONNECTST: 10.3.22.34

pe-zkp-1.int.dev.ssi-cloud.com: git -congimgt –env.config – pe.rs.rs – solr.config.yml - zk

* **streamx connectors on pe for** cisconic :

sudo /var/lib/sfdc-tool/bin/runSFDCTool.sh createStreamxConnector -Denv=pe -Dversion=rs-DzkServer=pe-zkp-1.int.dev.ssi-cloud.com:2181 kafkaConnectUrl=<http://localhost:8083/connectors> -Dtenant=Ciscoinc nameSpace=onep

* **streamx connectors on pe for** netapp:

sudo /var/lib/sfdc-tool/bin/runSFDCTool.sh**Symentc versioning of orca on all env**  createStreamxConnector -Denv=pe -Dversion=rs-DzkServer=pe-zkp-1.int.dev.ssi-cloud.com:2181 kafkaConnectUrl=<http://localhost:8083/connectors> -Dtenant=Netapp nameSpace=onep

**#ISSUE 25**

**delete streamx connectors on pe :**

* **delete streamx connectors on pe for** ciscoinc  
  sudo /var/lib/sfdc-tool/bin/runSFDCTool.sh deleteStreamxConnector kafkaConnectUrl=[http://localhost:8083](http://localhost:8083/)/connectors kafkaConnectorName=streamx-connector-pe-rs-prism-ciscoinc
* **delete streamx connectors on pe for** netapp :  
  sudo /var/lib/sfdc-tool/bin/runSFDCTool.sh deleteStreamxConnector kafkaConnectUrl=[http://localhost:8083](http://localhost:8083/)/connectors kafkaConnectorName=streamx-connector-pe-rs-prism-netapp

**#ISSUE 26**

**latest config mgmt on SQA-STK3 and PE-RS from 1.3.8**

* deploy – sqa-stk3 : zkconfig-frb3-deploy-sqa-stk3
* deploy- pe-rs - : all\_zkconfig-deploy-p**Symentc versioning of orca on all env** e-rs

**#ISSUE 27**

**Connnectors restart sfdc(kafkacst) sbx frb5**  
Ip:8083 -- list on ui - cmd restart (wiki doc)  
Jenkins job to check connector status

**#ISSUE 28**

**Alert connectors were nt in active state**  
shared env p all connectors : (jenkins job)  
Sbo rs : connectors are nt in wrkng state

**#ISSUE 29**

**Change replication factor for kafka topics**  
1. Sqa stk3 (step1)  
Rs1.3.8 build common gateway and df  
Rundeck job to copy binaries for 2 services gateway and datafabric  
Deploy -  
sqa stk3 and on Pe rs - commons , gateway , df  
Restart services : job multiservicerestart for df and gateway on sbx stk3 n pe rs  
2. Pe (all steps )  
In pe env go to Kafaka broker instance - download script frm doc - run cmd - chnge replication factor - verify  
Chef -lines add , pull n merge requst  
Chef client run on all instances  
Sudo chef-client  
Kafka restart

**#ISSUE 30**

* **Symentc versioning of orca on all env :** sbx sqa sbo pe (snapshot build)  
  jenkins : Build trigger : orca-ci to copy rpm into libs-snapshot-local  
  Artifactory : check rpm – latest rpm is presnt or nt in libs-snapshot-local
* if present thn copy the same in all env , Lib snapshot local (arti) to xyz targets for each env
* first chk the pointing of all env and check the target loc   
  Copy manually all rpm from lib snappshot local to all env : sbx sqa sbo pe renew
* sbx – for stk1 and 5 --package-RS\_1.3.8-ise-local package-dev1.0-ise-local
* sqa---for stk3 --package-RS\_1.3.8-devcertified-local , package-RS\_1.3.7-qa-local
* sbo-- for rs -- package-RS\_1.3.7.5-qa-local , package-RS\_1.2-qa-local
* pe – for rs and etl-- package-RS\_1.3.8-devcertified-local  
    
  Similarly do it for all env  
  Rundeck : run the job after copying all the rpm - ssi chef -Emr rpm deploymnt  
  do the same for all   
  Check at s3 once done - srev emr test – pe -etl – repo – setup scripts – publish app metrices.sh

**-=----------------------------------------------------------------------------------------------------------------------**

**#ISSUE 31**

**Symentc versioning of orca on RENEW (TAGGED BUILD)**

create tagged build of orca -ci – build with option : a**rtifactory Release Staging**

tagged build generated at libs-release local

copy tagged build rpm from libs-release local to package-RS\_1.3.7.5-prod-local , as renew rs is pointing to package-RS\_1.3.7.5-prod-local

renew-- for rs use tagged build -- package-RS\_1.3.7.5-prod-local

Rundeck : run the job after copying all the rpm - ssi chef -Emr rpm deploymnt for renew rs

Check at s3 once done - srev emr test

**#ISSUE 32**

* **C2r simulation on sqa stk3**  
  Simulation disable enable to mock payment services od dell  
  Zoinavigator login: zoonavigator.dev.ssi-cloud.com

connection string : git – configmgt – sqa stk3 frb3 – solr cofig.yml – zk :

shared-zkp-1.int.dev.ssi-cloud.com  
/env/sqa/releases/frb3/env-config/dell  
C2r simulation : enable : true  
Change timestamp n refrsh

* **#ISSUE 33**
* **Environment upgrade on sqa stk3 and pe rs for 1.3.8**  
  Copy : chk pointing  
  Rundeck:promote binaries( ise local se dev certified )  
  Deploy :  
  Jenkins : sqa stk3 - all services deploy  
  Pr rs - all services

**#ISSUE 34**

* **Automated sshpass installation on sbx stk1**  
  9Code pull for orca  
  Made changes in orca using sublime  
  Create local branch  
  Commit n pushed chnges in local brnch  
  Uploadd the install dependencies file on s3  
  Edit branch in jenkins job for autoscaling  
  Disable symentc versioning on sbx for smtime  
  Deploy -allservices -sbx stk1 -comment  
  Trigger launch cluster job with new code chnges  
  Chk cluster health  
  Pull request create  
  Terminate the cluster from cf

**#ISSUE 35**

**Instances increase on pe for gtw , c2r , dqi , dqs , gwr , rbs**  
aws -ec2 -autoscaaling groups -pe rbs

**#ISSUE 36**

**Symntc versing rs etl for (pe etl)**  
Target - opsboard  
Chk pe etl - dev ise local (target)  
Chk jenkins  
Build - Dev folder as pe etl is on dev  
In dev chk rs-etl-dev-ci  
Chk pointing - dev ise local (source)  
Source n target same , no change no copy  
Run deploy job : emrrpm deploymnt  
Symentc versioning of orca  
Orca in artifactory - lib snapshot local -orca scripts - rpm  
Orca on jenkins : master  
Repo pointing diff : lib snapshot local to xyz  
Rundeck job : promotebinaries  
Disabled c2r simulation on sqa stk3  
Pricing refrsh on pe

**#ISSUE 37**  
**Upgrade sqa stk3 1.3.8 with all service**  
Rundeck : promote binaries  
Deploy : all services

**#ISSUE 38**  
  
**Upgrade pe rs1.3.8 with all services**  
Rundeck : promote binaries  
Deploy : Zk first-- All services nxt  
Dqs crw c2r : monit summary

**#ISSUE 39**

**pipeline for helm deployment using jenkins :**

pipeline {

def Namespace = "default"  
 def ImageName = "abc"  
 def Creds = "xyz"

stages {

stage('checkout') {

steps {

git 'https://github.com/SSI-Avalon/orca.git

sh "git rev-parse --short HEAD > .git/commit-id

imageTag= readFile('.git/commit-id').trim()

}

}

stage('RUN Unit Tests') {

sh "npm install"

sh "npm test"

}

}

stage('Docker Build, Push') {

steps {

withDockerRegistry([credentialsId: "${Creds}", url:'------------------------']) {

sh "docker build -t ${ImageName}:${imageTag} ."

sh "docker push ${ImageName}"

}

}

}

stage('Deploy on K8s') {

steps {

sh "ansible-playbook /var/lib/jenkins/ansible/deploy.yml --user=xyz --extra-vars ImageName=${ImageName} --extra-vars imageTag=${imageTag} –extra-vars Namespace=${Namespace}"

}

}

stage('run kubectl') {

steps {

sh 'kubectl get pods'

}

}

stage('Deploy helm ') {

steps {

sh 'ls templates'

sh 'helm install --name abc ./xyz'

sh 'kubectl get deploy'

sh 'helm list'

}

}

}

}

---------------------------------------------------------------------------------------------------------------**#ISSUE 40**

**00438484: parameterized jenkins job create** –

added conditional step to redirect from 1 job to other on the basis of parameters

demo job and demo job2

build job – datafabric-dev-ci – calling k8-genericdeployment -

added build – contional step single – strings match – project to call – postman\_dtf\_execution

for service datafabric as a parameter .

**#ISSUE 41**

**New Branch Creation on re\_etl and implementations : create\_new\_branch job**  
 git clone <https://github.com/SSI-Avalon/ConfigMgmt.git>  
 git branch  
 git checkout release/6.4.0  
 git checkout -b release/6.5.0  
 mvn versions:set -DgenerateBackupPoms=false -DnewVersion=i5glob\_1.0.1-SNAPSHOT  
 git add .  
 git status -s  
 git commit -m 'creating new branch feature/i5glob for implementations'   
 git push origin feature/i5glob  
 make the build job in jenkins copy config from previos job -- build- i5glob -implementations\_RS\_1.3.8

**#ISSUE 42**

**script to get the details for emr cluster : emrdetails.sh and mail.sh**

**ebs modified alert : script to send mail if ebs volume is used and modified . AUTOMATION : Ebs autoscaling alert through aws-ses**

#!/bin/bash

id=$(cat /mnt/var/lib/info/job-flow.json | grep jobFlowId | cut -f2 -d: | cut -f2 -d'"')

aws emr describe-cluster --cluster-id $id --region us-east-1 > clusterdetails.json

clusterName=$(cat clusterdetails.json | jq -r '.Cluster.Name' clusterdetails.json | tr -d '"')

caseNumber=$(cat clusterdetails.json | jq --raw-output '.Cluster.Tags[4].Value' clusterdetails.json | tr -d '"')

caseOwnerName=$(cat clusterdetails.json | jq --raw-output '.Cluster.Tags[0].Value' clusterdetails.json | tr -d '"')

aws ses send-email --region us-east-1 --from "Renew-Ops-T1@servicesource.com" --destination "ToAddresses=ServiceSourceDev@impetus.co.in" --message "Subject={Data=EBS modified,Charset=UTF-8},Body={Text={Data=heyyyyya,Charset=UTF-8},Html={Data=<!DOCTYPE html><html><body><p>Hi $caseOwnerName&#44;</p><p>The volume on the $clusterName is modified by 50% .</p><p>Case-$caseNumber</p><p style="line-height:0.6">Regards&#44;</p><p style="line-height:0.6">Ops Team.</p></body></html>,Charset=UTF-8}}"

echo "Mail sent to : $caseOwnerName , as the volume on the cluster is modified Volume by 50%"

steps to create a new brnch a push changes into new brnch to create pull request :

git branch : chk branch

git checkout -b ebs-tmp : chnge brnch

git pull origin master : pull latest code frm master

git status -s : chk chnges

git add . : add chnges

git commit -m 'ebs changes'

git push origin ebs-tmp

**#ISSUE 43**

**Prepare Kubernetes Environment on SQA STK1 "**

**Del indexes from sqa stk1 and restore wd proper records**

1. sqa.frb1.dell.app.affinity.orgs
2. sqa.frb1.dell.app.bookings
3. sqa.frb1.dell.app.coveredassets
4. sqa.frb1.dell.app.forecasts
5. sqa.frb1.dell.app.lineitems
6. sqa.frb1.dell.app.offers
7. sqa.frb1.dell.app.opportunities
8. sqa.frb1.dell.app.products
9. sqa.frb1.dell.app.quotes
10. sqa.frb1.dell.app.serviceassets
11. sqa.frb1.dell.app.target.data
12. sqa.frb1.dell.app.tasks
13. sqa.frb1.dell.core.addresses
14. sqa.frb1.dell.core.organizations\_non\_tx
15. sqa.frb1.dell.core.organizations\_tx
16. sqa.frb1.dell.core.persons

* list indexes
* launch indexer --ip
* del data frm existing indexes : “DO FOR ALL INDEXES”

curl 'http://<hostname>:8080/solr/sqa.frb1.dell.app.opportunities/update?stream.body=<delete><query>\*:\*</query></delete>&commit=true'

* chk on ui deleted/nt
* create new solr indexes : “DO FOR ALL INDEXES”
* for app.products : run from *var/*lib/idx/bin
* Non-TX  
  ./runmigration.sh solrIndexes '-Denv=sqa' '-DzkServer=shared-zkp-k8s.int.dev.ssi-cloud.com:2181' 'migration.tenant.name=dell' 'solr.default.routing.key=non\_tx' 'solr.index.collection.name=app.products' '-Dversion=frb1' 'mongo.select.query={"isReferenced" : { "$exists" : false }}' 'mongo.query.defaultHint.field=isReferenced' 'solr.skip.index=false' 'mongo.skip.data.population=true'   
  TX  
  ./runmigration.sh solrIndexes '-Denv=sqa' '-DzkServer=shared-zkp-k8s.int.dev.ssi-cloud.com:2181' 'migration.tenant.name=dell' 'solr.default.routing.key=tx' 'solr.index.collection.name=app.products' '-Dversion=frb1' 'mongo.select.query={"isReferenced" : { "$exists" : true }}' 'mongo.query.defaultHint.field=isReferenced' 'solr.skip.index=false' 'mongo.skip.data.population=true'
* for rest of the indexes :

./runmigration.sh solrIndexes '-Denv=sqa' '-DzkServer=shared-zkp-k8s.int.dev.ssi-cloud.com:2181' 'migration.tenant.name=dell' 'solr.index.collection.name=app.affinity.orgs,app.bookings,app.coveredassets,app.forecasts,app.lineitems,app.offers,app.quotes,app.serviceassets,app.target.data,app.tasks,core.addresses,core.persons' 'solr.skip.index=false' 'mongo.skip.data.population=true' '-Dversion=frb1'

* core.organization\_tx  
  ./runmigration.sh solrIndexes '-Dversion=frb1' '-Denv=sqa' '-DzkServer=shared-zkp-k8s.int.dev.ssi-cloud.com:2181' 'migration.tenant.name=dell' 'solr.index.collection.name=core.organizations' 'indexingEligibleCollectionSplits=tx' 'mongo.select.query={"\_id" : {"$gt" : {"$oid": "000000000000000000000000"}},"isReferenced":{"$exists":true}}

core.organization\_non tx  
./runmigration.sh solrIndexes '-Dversion=frb1' '-Denv=sqa' '-DzkServer=shared-zkp-k8s.int.dev.ssi-cloud.com:2181' 'migration.tenant.name=dell' 'solr.index.collection.name=core.organizations' 'indexingEligibleCollectionSplits=non\_tx' 'mongo.select.query={"\_id" : {"$gt" : {"$oid": "000000000000000000000000"}},"isReferenced":{"$exists":false}}'

* check count : login to mongo instance : shared-dbs-1.int.dev.ssi-cloud.com

1. db.app.affinity.orgs.count()

3861461 : 3861456

1. db.app.bookings.count()

6828 : 6334

1. db.app.coveredassets.count()

3384748 : 3384700

1. db.app.forecasts.count()

1936 : 1946

1. db.app.lineitems.count()

86939 : 82808

1. db.app.offers.count()

1030401 : 703283

1. db.app.opportunities.count()

60528 : 50776

1. db.app.products.count()

12498162 : 20872858

1. db.app.quotes.count()

53138 : 43346

1. db.app.serviceassets.count()

5159471 :

1. db.app.target.data.count()

4929 : 4915

1. db.app.tasks.count()

3232 : 3000

1. db.core.addresses.count()

1798238 : 1798075

1. db.core.organizations.count()

1742844 : 1102261 (tx) : 1079172 (non-tx)

1. db.core.persons.count()

1558730 : 1558534

* check collection count for solr db:

1. curl -s 'http://shared-sdb-1.int.dev.ssi-cloud.com:8080/solr/sqa.frb1.dell.app.affinity.orgs/select?indent=on&q=\*:\*&wt=json' | grep numFound
2. curl -s 'http://shared-sdb-1.int.dev.ssi-cloud.com:8080/solr/sqa.frb1.dell.app.bookings/select?indent=on&q=\*:\*&wt=json' | grep numFound
3. curl -s 'http://shared-sdb-1.int.dev.ssi-cloud.com:8080/solr/sqa.frb1.dell.app.coveredassets/select?indent=on&q=\*:\*&wt=json' | grep numFound
4. curl -s 'http://shared-sdb-1.int.dev.ssi-cloud.com:8080/solr/sqa.frb1.dell.app.forecasts/select?indent=on&q=\*:\*&wt=json' | grep numFound
5. curl -s 'http://shared-sdb-1.int.dev.ssi-cloud.com:8080/solr/sqa.frb1.dell.app.lineitems/select?indent=on&q=\*:\*&wt=json' | grep numFound
6. curl -s 'http://shared-sdb-1.int.dev.ssi-cloud.com:8080/solr/sqa.frb1.dell.app.offers/select?indent=on&q=\*:\*&wt=json' | grep numFound
7. curl -s 'http://shared-sdb-1.int.dev.ssi-cloud.com:8080/solr/sqa.frb1.dell.app.opportunities/select?indent=on&q=\*:\*&wt=json' | grep numFound
8. curl -s 'http://shared-sdb-1.int.dev.ssi-cloud.com:8080/solr/sqa.frb1.dell.app.products/select?indent=on&q=\*:\*&wt=json' | grep numFound
9. curl -s 'http://shared-sdb-1.int.dev.ssi-cloud.com:8080/solr/sqa.frb1.dell.app.quotes/select?indent=on&q=\*:\*&wt=json' | grep numFound
10. curl -s 'http://shared-sdb-1.int.dev.ssi-cloud.com:8080/solr/sqa.frb1.dell.app.serviceassets/select?indent=on&q=\*:\*&wt=json' | grep numFound
11. curl -s 'http://shared-sdb-1.int.dev.ssi-cloud.com:8080/solr/sqa.frb1.dell.app.target.data/select?indent=on&q=\*:\*&wt=json' | grep numFound
12. curl -s 'http://shared-sdb-1.int.dev.ssi-cloud.com:8080/solr/sqa.frb1.dell.app.tasks/select?indent=on&q=\*:\*&wt=json' | grep numFound
13. curl -s 'http://shared-sdb-1.int.dev.ssi-cloud.com:8080/solr/sqa.frb1.dell.core.addresses/select?indent=on&q=\*:\*&wt=json' | grep numFound
14. curl -s 'http://shared-sdb-1.int.dev.ssi-cloud.com:8080/solr/sqa.frb1.dell.core.organizations\_non\_tx/select?indent=on&q=\*:\*&wt=json' | grep numFound
15. curl -s 'http://shared-sdb-1.int.dev.ssi-cloud.com:8080/solr/sqa.frb1.dell.core.organizations\_tx/select?indent=on&q=\*:\*&wt=json' | grep numFound
16. curl -s 'http://shared-sdb-1.int.dev.ssi-cloud.com:8080/solr/sqa.frb1.dell.core.persons/select?indent=on&q=\*:\*&wt=json' | grep numFound

**#ISSUE 44**

**pd alert : sumologic collector restart : 10.3.20.83**

login to instance

ps -ef | grep sumologic

sudo ./collector start

sudo ./collector status

**#ISSUE 45**

**load app.pricings2 data into hive delta table on pe where source dbname – automation\_dell and source collection – app.pricings2 target dbname – dell and**

**target tablename - app\_pricings\_delta in new partition ds=2019-09-04**

* launch a pe idx from aws
* login to idx
* use automation\_app : create new mongo db
* create new collection : db.createCollection("test")
* use admin
* db.auth("devopsUserAdmin", "vP4MehogTEpSO6Wp4875")
* use automation\_app
* create new user in db

db.createUser({user: "admin", pwd: "welcome", roles: [{role: "readWrite", db: "automation\_app"}]});

* create new user in db

db.createUser({user: "reporting", pwd: "password123", roles: [{role: "readWrite", db: "automation\_app"}]});

* grant roles :

db.grantRolesToUser("reporting",[ "read" , { role: "read", db: "automation\_app" } ])

* use admin
* db.Getusers() : list users in a db
* mongo restore for automation\_app db , run from test folder

nohup mongorestore --collection app.pricings -u reporting -p password123 --db automation\_app app.pricings.bson &

* tail -f nohup.out
* new db on hive

launch a pe cluster

login to cluster

create database pricing\_db;

show databases;

* add steps in the cluster , clone step and name

use pricing\_db ;

show tables; (there will be app\_pricings and app\_pricings\_delte)

* added load data to hive tables for pe : add this command into the step of cluster

s3://srev-emr-test/pe/rs/repository/applications/dataload/scripts/dataload.sh -bsrev-emr-test/pe/rs/repository/applications -tdell -capp\_pricings -a7 -mreporting:password123@pe-dbs-1.int.dev.ssi-cloud.com -greporting:password123@pe-dbs-1.int.dev.ssi-cloud.com -kautomation\_app -oconfigdata\_rs -lpricing\_db -ntrue -itrue

**#ISSUE 46**

**Create DPS bad record hive tables for tenants dell and ciscoinc on PE env**

**added these commands as a step in existing pe-rs cluster**

**tenant dell : entity onep**

s3://srev-emr-test/pe/rs/repository/applications/warehouse/scripts/hiveDPSSchemaGen.sh -s s3://srev-emr-test/pe/rs/repository/applications/warehouse/hive/generated\_schemas/dps\_error\_schemas/onep -o s3://avaloncms/pe/dpsdata -d dell\_prism\_abu -t dell -v rs -b true

**tenant dell : entity onep : version :** [**pe.rs.abu**](https://github.com/SSI-Avalon/ConfigMgmt/tree/release/RS_1.3.8/envConfig/pe.rs.abu)

s3://srev-emr-test/pe/rs/repository/applications/warehouse/scripts/hiveDPSSchemaGen.sh -s s3://srev-emr-test/pe/rs/repository/applications/warehouse/hive/generated\_schemas/dps\_error\_schemas/onep -o s3://avaloncms/pe/dpsdata -d dell\_prism\_abu -t dell -v abu -b true

**tenant dell : entity onep : version :** [**pe.rs.abu\_sit**](https://github.com/SSI-Avalon/ConfigMgmt/tree/release/RS_1.3.8/envConfig/pe.rs.abu_sit)

s3://srev-emr-test/pe/rs/repository/applications/warehouse/scripts/hiveDPSSchemaGen.sh -s s3://srev-emr-test/pe/rs/repository/applications/warehouse/hive/generated\_schemas/dps\_error\_schemas/onep -o s3://avaloncms/pe/dpsdata -d dell\_prism\_abu\_sit -t dell -v abu\_sit -b true

**tenant dell : entity onep : version :** [**pe.rs.apj**](https://github.com/SSI-Avalon/ConfigMgmt/tree/release/RS_1.3.8/envConfig/pe.rs.apj)

s3://srev-emr-test/pe/rs/repository/applications/warehouse/scripts/hiveDPSSchemaGen.sh -s s3://srev-emr-test/pe/rs/repository/applications/warehouse/hive/generated\_schemas/dps\_error\_schemas/onep -o s3://avaloncms/pe/dpsdata -d dell\_prism\_apj -t dell -v apj -b true

**tenant dell : entity onep : version :** [**pe.rs.apj\_sit**](https://github.com/SSI-Avalon/ConfigMgmt/tree/release/RS_1.3.8/envConfig/pe.rs.apj_sit)

s3://srev-emr-test/pe/rs/repository/applications/warehouse/scripts/hiveDPSSchemaGen.sh -s s3://srev-emr-test/pe/rs/repository/applications/warehouse/hive/generated\_schemas/dps\_error\_schemas/onep -o s3://avaloncms/pe/dpsdata -d dell\_prism\_apj\_sit -t dell -v apj\_sit -b true

**tenant dell : entity onep : version :** [**pe.rs.emea**](https://github.com/SSI-Avalon/ConfigMgmt/tree/release/RS_1.3.8/envConfig/pe.rs.emea)

s3://srev-emr-test/pe/rs/repository/applications/warehouse/scripts/hiveDPSSchemaGen.sh -s s3://srev-emr-test/pe/rs/repository/applications/warehouse/hive/generated\_schemas/dps\_error\_schemas/onep -o s3://avaloncms/pe/dpsdata -d dell\_prism\_emea -t dell -v emea -b true

**tenant dell : entity onep : version :** [**pe.rs.emea\_sit**](https://github.com/SSI-Avalon/ConfigMgmt/tree/release/RS_1.3.8/envConfig/pe.rs.emea_sit)

s3://srev-emr-test/pe/rs/repository/applications/warehouse/scripts/hiveDPSSchemaGen.sh -s s3://srev-emr-test/pe/rs/repository/applications/warehouse/hive/generated\_schemas/dps\_error\_schemas/onep -o s3://avaloncms/pe/dpsdata -d dell\_prism\_emea\_sit -t dell -v emea\_sit -b true

**tenant dell : entity onep : version :** [**pe.rs.latam**](https://github.com/SSI-Avalon/ConfigMgmt/tree/release/RS_1.3.8/envConfig/pe.rs.latam)

s3://srev-emr-test/pe/rs/repository/applications/warehouse/scripts/hiveDPSSchemaGen.sh -s s3://srev-emr-test/pe/rs/repository/applications/warehouse/hive/generated\_schemas/dps\_error\_schemas/onep -o s3://avaloncms/pe/dpsdata -d dell\_prism\_latam -t dell -v latam -b true

**tenant dell : entity onep : version :** [**pe.rs.latam\_sit**](https://github.com/SSI-Avalon/ConfigMgmt/tree/release/RS_1.3.8/envConfig/pe.rs.latam_sit)

s3://srev-emr-test/pe/rs/repository/applications/warehouse/scripts/hiveDPSSchemaGen.sh -s s3://srev-emr-test/pe/rs/repository/applications/warehouse/hive/generated\_schemas/dps\_error\_schemas/onep -o s3://avaloncms/pe/dpsdata -d dell\_prism\_latam\_sit -t dell -v latam\_sit -b true

**tenant dell : entity renew**

s3://srev-emr-test/pe/rs/repository/applications/warehouse/scripts/hiveDPSSchemaGen.sh -s s3://srev-emr-test/pe/rs/repository/applications/warehouse/hive/generated\_schemas/dps\_error\_schemas -o s3://avaloncms/pe/dpsdata -d dell -t dell -v rs -b true

**tenant ciscoinc : entity onep**

s3://srev-emr-test/pe/rs/repository/applications/warehouse/scripts/hiveDPSSchemaGen.sh -s s3://srev-emr-test/pe/rs/repository/applications/warehouse/hive/generated\_schemas/dps\_error\_schemas/onep -o s3://avaloncms/pe/dpsdata -d ciscoinc -t ciscoinc -b true

**tenant ciscoinc : entity renew**

s3://srev-emr-test/pe/rs/repository/applications/warehouse/scripts/hiveDPSSchemaGen.sh -s s3://srev-emr-test/pe/rs/repository/applications/warehouse/hive/generated\_schemas/dps\_error\_schemas -o s3://avaloncms/pe/dpsdata -d ciscoinc -t ciscoinc -b true

for sbo :

**tenant dell : entity onep**

s3://srev-emr-test/sbo/rs/repository/applications/warehouse/scripts/hiveDPSSchemaGen.sh -s s3://srev-emr-test/sbo/rs/repository/applications/warehouse/hive/generated\_schemas/dps\_error\_schemas/onep -o s3://avaloncms/pe/dpsdata -d sbo\_rs\_dell\_prism\_rs -t dell -v rs -b true

**tenant dell : entity renew**

s3://srev-emr-test/sbo/rs/repository/applications/warehouse/scripts/hiveDPSSchemaGen.sh -s s3://srev-emr-test/sbo/rs/repository/applications/warehouse/hive/generated\_schemas/dps\_error\_schemas -o s3://avaloncms/pe/dpsdata -d sbo\_rs\_dell\_prism\_rs -t dell -v rs -b true

for sbx stk1 :

**tenant dell : entity onep**

s3://srev-emr-test/sbx/stk1/repository/applications/warehouse/scripts/hiveDPSSchemaGen.sh -s s3://srev-emr-test/sbx/stk1/repository/applications/warehouse/hive/generated\_schemas/dps\_error\_schemas/onep -o s3://avaloncms/sbx/dpsdata -d sbx\_frb1\_dell\_prism -t dell -v frb1 -b true

**tenant dell : entity renew**

s3://srev-emr-test/sbx/stk1/repository/applications/warehouse/scripts/hiveDPSSchemaGen.sh -s s3://srev-emr-test/sbx/stk1/repository/applications/warehouse/hive/generated\_schemas/dps\_error\_schemas -o s3://avaloncms/sbx/dpsdata -d sbx\_frb1\_dell\_prism -t dell -v frb1 -b true

for sbx stk5 :

**tenant dell : entity onep**

s3://srev-emr-test/sbx/stk5/repository/applications/warehouse/scripts/hiveDPSSchemaGen.sh -s s3://srev-emr-test/sbx/stk5/repository/applications/warehouse/hive/generated\_schemas/dps\_error\_schemas/onep -o s3://avaloncms/sbx/dpsdata -d sbx\_stk5\_frb5\_dell\_prism

-t dell -v frb5 -b true

**tenant dell : entity renew**

s3://srev-emr-test/sbx/stk5/repository/applications/warehouse/scripts/hiveDPSSchemaGen.sh -s s3://srev-emr-test/sbx/stk5/repository/applications/warehouse/hive/generated\_schemas/dps\_error\_schemas -o s3://avaloncms/sbx/dpsdata -d sbx\_stk5\_frb5\_dell\_prism -t dell -v frb5 -b true

**#ISSUE 47**

**add five s3 events in PE-RS-AddPartitionFunction lambda function  where bucket name - avaloncms   
and Keys are ----  pe/dpsdata/dell/emea/v1 , pe/dpsdata/dell/abu/v1 , pe/dpsdata/dell/apj/v1  pe/dpsdata/dell/rs/v1  , pe/dpsdata/ciscoinc**

login to s3-

check bucket avaloncms-

add event as -

event name (some value )

choose : All object create events

prefix (add keys one by one to create 5 events in lambda function )- pe/dpsdata/dell/emea/v1 , pe/dpsdata/dell/abu/v1 , pe/dpsdata/dell/apj/v1  pe/dpsdata/dell/rs/v1  , pe/dpsdata/ciscoinc

snd to lambda

add lambda arn -

lambda arn from lambda function - save

**#ISSUE 48**

**Case 00437232:- Refresh PRISM Dell EMEA DS :- bulkin completed and dataload is in progress for all entities on PE-RS-TEST . Please monitor and terminate the cluster once done .**

Steps:

* compare their SFDC & Mongo Counts.
* MongoToHive Dataload: DATALOAD : run-dataload : mongo 2 hive and bulkin : sfdc to mongo
* downscale DPS/DTF instances
* Enable automatic upscale job https://devadmin.dev.ssi-cloud.com/job/dpsJob\_monitoring\_pe\_rs/

MONGO COUNT FROM : COUNT MONGO ENTITY JOB – get list

SFDC COUNT FROM : [https://ssidellemea--dsdellemea.cs37.my.salesforce.com](https://ssidellemea--dsdellemea.cs37.my.salesforce.com/)

utility : <https://workbench.developerforce.com/query.php>

and compare

1. Account : 1480184 1481878
2. entitlement : 1481878 1481878
3. feeditem : 108 108
4. currencytype : 39 39
5. servicesource1\_\_chl\_partner\_renewal\_relationship\_\_c : 493302 493302
6. asset : 19915524
7. attachment : 336613 336613
8. case : 81461 81461
9. contact: 29259 29331
10. note : 67657 67657
11. opportunity : 1229697 1229697
12. opportunitylineitem : 1192094 1192094
13. pricebook2 : 1 1
14. pricebookentry : 24624 12312
15. product2 : 2694 2694
16. quote : 155934 155934
17. quotelineitem : 159559 159559
18. recordtype : 23 25
19. servicesource1\_\_ren\_renews\_to\_\_c : 19054871
20. servicesource1\_\_ren\_source\_\_c : 50556 50556
21. task : 3853 3740
22. ssi\_zth\_\_location\_address\_\_c : 343342 50556
23. servicesource1\_\_chl\_partner\_opportunity\_\_c : 1081370 1081370
24. user : 425 425

DATALOAD : run-dataload : mongo 2 hive

bulkin : sfdc to mongo

add the steps to a pe cluster

1. s3://srev-emr-test/pe/rs/repository/applications/dataload/scripts/dataload.sh -bsrev-emr-test/pe/rs/repository/applications -tdell -cprism\_account -a7 -mreporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -greporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -konepdata\_dell\_emea -sappdata\_sup -oconfigdata\_rs -ldell\_prism\_emea -ntrue -itrue
2. s3://srev-emr-test/pe/rs/repository/applications/dataload/scripts/dataload.sh -bsrev-emr-test/pe/rs/repository/applications -tdell -cprism\_entitlement -a7 -mreporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -greporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -konepdata\_dell\_emea -sappdata\_sup -oconfigdata\_rs ldell\_prism\_emea -ntrue -itrue
3. s3://srev-emr-test/pe/rs/repository/applications/dataload/scripts/dataload.sh -bsrev-emr-test/pe/rs/repository/applications -tdell -cprism\_feeditem -a7 -mreporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -greporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -konepdata\_dell\_emea -sappdata\_sup -oconfigdata\_rs -ldell\_prism\_emea -ntrue -itrue
4. s3://srev-emr-test/pe/rs/repository/applications/dataload/scripts/dataload.sh -bsrev-emr-test/pe/rs/repository/applications -tdell -cprism\_currencytype -a7 -mreporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -greporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -konepdata\_dell\_emea -sappdata\_sup -oconfigdata\_rs -ldell\_prism\_emea -ntrue -itrue
5. s3://srev-emr-test/pe/rs/repository/applications/dataload/scripts/dataload.sh -bsrev-emr-test/pe/rs/repository/applications -tdell -cprism\_servicesource1\_\_chl\_partner\_renewal\_relationship\_\_c -a7 -mreporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -greporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -konepdata\_dell\_emea -sappdata\_sup -oconfigdata\_rs -ldell\_prism\_emea -ntrue -itrue
6. s3://srev-emr-test/pe/rs/repository/applications/dataload/scripts/dataload.sh -bsrev-emr-test/pe/rs/repository/applications -tdell -cprism\_asset -a7 -mreporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -greporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -konepdata\_dell\_emea -sappdata\_sup -oconfigdata\_rs -ldell\_prism\_emea -ntrue -itrue
7. s3://srev-emr-test/pe/rs/repository/applications/dataload/scripts/dataload.sh -bsrev-emr-test/pe/rs/repository/applications -tdell -cprism\_attachment -a7 -mreporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -greporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -konepdata\_dell\_emea -sappdata\_sup -oconfigdata\_rs -ldell\_prism\_emea -ntrue -itrue
8. s3://srev-emr-test/pe/rs/repository/applications/dataload/scripts/dataload.sh -bsrev-emr-test/pe/rs/repository/applications -tdell -cprism\_case -a7 -mreporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -greporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -konepdata\_dell\_emea -sappdata\_sup -oconfigdata\_rs -ldell\_prism\_emea -ntrue -itrue
9. s3://srev-emr-test/pe/rs/repository/applications/dataload/scripts/dataload.sh -bsrev-emr-test/pe/rs/repository/applications -tdell -cprism\_contact -a7 -mreporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -greporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -konepdata\_dell\_emea -sappdata\_sup -oconfigdata\_rs -ldell\_prism\_emea -ntrue -itrue
10. s3://srev-emr-test/pe/rs/repository/applications/dataload/scripts/dataload.sh -bsrev-emr-test/pe/rs/repository/applications -tdell -cprism\_note -a7 -mreporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -greporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -konepdata\_dell\_emea -sappdata\_sup -oconfigdata\_rs -ldell\_prism\_emea -ntrue -itrue
11. s3://srev-emr-test/pe/rs/repository/applications/dataload/scripts/dataload.sh -bsrev-emr-test/pe/rs/repository/applications -tdell -cprism\_opportunity -a7 -mreporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -greporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -konepdata\_dell\_emea -sappdata\_sup -oconfigdata\_rs -ldell\_prism\_emea -ntrue -itrue
12. s3://srev-emr-test/pe/rs/repository/applications/dataload/scripts/dataload.sh -bsrev-emr-test/pe/rs/repository/applications -tdell -cprism\_opportunitylineitem -a7 -mreporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -greporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -konepdata\_dell\_emea -sappdata\_sup -oconfigdata\_rs -ldell\_prism\_emea -ntrue -itrue
13. s3://srev-emr-test/pe/rs/repository/applications/dataload/scripts/dataload.sh -bsrev-emr-test/pe/rs/repository/applications -tdell -cprism\_pricebook2 -a7 -mreporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -greporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -konepdata\_dell\_emea -sappdata\_sup -oconfigdata\_rs -ldell\_prism\_emea -ntrue -itrue
14. s3://srev-emr-test/pe/rs/repository/applications/dataload/scripts/dataload.sh -bsrev-emr-test/pe/rs/repository/applications -tdell -cprism\_pricebookentry -a7 -mreporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -greporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -konepdata\_dell\_emea -sappdata\_sup -oconfigdata\_rs -ldell\_prism\_emea -ntrue -itrue
15. s3://srev-emr-test/pe/rs/repository/applications/dataload/scripts/dataload.sh -bsrev-emr-test/pe/rs/repository/applications -tdell -cprism\_product2 -a7 -mreporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -greporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -konepdata\_dell\_emea -sappdata\_sup -oconfigdata\_rs -ldell\_prism\_emea -ntrue -itrue
16. s3://srev-emr-test/pe/rs/repository/applications/dataload/scripts/dataload.sh -bsrev-emr-test/pe/rs/repository/applications -tdell -cprism\_quote -a7 -mreporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -greporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -konepdata\_dell\_emea -sappdata\_sup -oconfigdata\_rs -ldell\_prism\_emea -ntrue -itrue
17. s3://srev-emr-test/pe/rs/repository/applications/dataload/scripts/dataload.sh -bsrev-emr-test/pe/rs/repository/applications -tdell -cprism\_quotelineitem -a7 -mreporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -greporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -konepdata\_dell\_emea -sappdata\_sup -oconfigdata\_rs -ldell\_prism\_emea -ntrue -itrue
18. s3://srev-emr-test/pe/rs/repository/applications/dataload/scripts/dataload.sh -bsrev-emr-test/pe/rs/repository/applications -tdell -cprism\_recordtype -a7 -mreporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -greporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -konepdata\_dell\_emea -sappdata\_sup -oconfigdata\_rs -ldell\_prism\_emea -ntrue -itrue
19. s3://srev-emr-test/pe/rs/repository/applications/dataload/scripts/dataload.sh -bsrev-emr-test/pe/rs/repository/applications -tdell -cprism\_servicesource1\_\_ren\_renews\_to\_\_c -a7 -mreporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -greporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -konepdata\_dell\_emea -sappdata\_sup -oconfigdata\_rs -ldell\_prism\_emea -ntrue -itrue
20. s3://srev-emr-test/pe/rs/repository/applications/dataload/scripts/dataload.sh -bsrev-emr-test/pe/rs/repository/applications -tdell -cprism\_servicesource1\_\_ren\_source\_\_c -a7 -mreporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -greporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -konepdata\_dell\_emea -sappdata\_sup -oconfigdata\_rs -ldell\_prism\_emea -ntrue -itrue
21. s3://srev-emr-test/pe/rs/repository/applications/dataload/scripts/dataload.sh -bsrev-emr-test/pe/rs/repository/applications -tdell -cprism\_task -a7 -mreporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -greporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -konepdata\_dell\_emea -sappdata\_sup -oconfigdata\_rs -ldell\_prism\_emea -ntrue -itrue
22. s3://srev-emr-test/pe/rs/repository/applications/dataload/scripts/dataload.sh -bsrev-emr-test/pe/rs/repository/applications -tdell -cprism\_ssi\_zth\_\_location\_address\_\_c -a7 -mreporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -greporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -konepdata\_dell\_emea -sappdata\_sup -oconfigdata\_rs -ldell\_prism\_emea -ntrue -itrue
23. s3://srev-emr-test/pe/rs/repository/applications/dataload/scripts/dataload.sh -bsrev-emr-test/pe/rs/repository/applications -tdell -cprism\_servicesource1\_\_chl\_partner\_opportunity\_\_c -a7 -mreporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -greporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -konepdata\_dell\_emea -sappdata\_sup -oconfigdata\_rs -ldell\_prism\_emea -ntrue -itrue
24. s3://srev-emr-test/pe/rs/repository/applications/dataload/scripts/dataload.sh -bsrev-emr-test/pe/rs/repository/applications -tdell -cprism\_user -a7 -mreporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -greporting:password123@pe-dbs-1.int.dev.ssi-cloud.com:27017 -konepdata\_dell\_emea -sappdata\_sup -oconfigdata\_rs -ldell\_prism\_emea -ntrue -itrue

-----------------------------------------------------------------------------------------------

**#ISSUE 49**

**Launched pe etl cluster also migrated a table from prod/rs to pe/etl**

**job : hivedbmigration**

-----------------------------------------------------------------------------------------------

**#ISSUE 50**

update latest code for PE-ETL-FtpFunction and PE-ETL-EtlFunction.

renewtools master ci --- build job

copy rpm frpm libs/snapshot /local to package hf 1.3.7.5 ise local

deploy job on rundeck : lambda deployment , dynamodb deployment

**#ISSUE 51**

**execute step (2 and 7 only) of below document on SBX STK5   
https://wiki.dev.ssi-cloud.com/pages/viewpage.action?pageId=100237921**

step2 :

i) Deploy packager for dell : DEPLOY- SBX-STK5- IMPLEMENTATION JOB – CONFIGURE -TENANT DELL ONLY

ii) Perform Semantic Versioning : DEPLOY- SBX-STK5- EMR-DEPLOY-SBX-STK5 (AUTO)

step7 :Generate onep dataload hive schemas on environments , for tenant dell and on sbx stk5

Added this as a step on sbx-stk5 cluster :

s3://srev-emr-test/sbx/stk5/repository/applications/warehouse/scripts/hiveDPSSchemaGen.sh -b false -s s3://srev-emr-test/sbx/stk5/repository/applications/warehouse/hive/generated\_schemas/dps\_schemas/onep -o s3://avaloncms/sbx/stk5/data -d sbx\_stk5\_frb5\_dell\_prism -t dell -p prism\_ -f \_load -v frb5

------------------------------------------------------------------------------------------------------

**#ISSUE 52**

revert changes of case 00439244 : Please configure SQS for "srev-v2dev-datapipeline-in" bucket with path "sbx/stk5/data/", it should be similar to sbx\_stk5\_sqsqueueconfig (DATA\_LOAD\_QUEUE) which we have currently in SBX environment.

Deleted the event :sbx\_stk5\_sqsqueconfig

prefix: sbx/stk5/data/

send to : SQS queue

sqs: SBX\_STK5\_DATA\_LOAD\_QUEUE

from s3 bucket : srev-v2dev-datapipeline-in

**#ISSUE 52**

create new stack : <https://wiki.dev.ssi-cloud.com/display/OPSDOC/Create+New+Stack+from+existing+Environment+for+RS>

emr setup : Create mysql databases for hive,hue and oozie

* connect to mysql client using the command:

mysql --host=dataservices-db.int.dev.ssi-cloud.com --user=ds\_root –password=aUONVvg1BSseySTkV4Dp

* Copy the mongo database from SQA-frb1 to SQA-frb4

onepdata

supdata

configdata

appdata

use admin

db.auth("devopsUserAdmin", "<password>")

use sqa\_configdata\_frb5

db.createUser({user: "admin", pwd: "welcome", roles: [{role: "readWrite", db: "sqa\_configdata\_frb5"}]})

db.createUser({user: "reporting", pwd: "password123", roles: [{role: "read", db: "sqa\_configdata\_frb5"}]})

use sqa\_onepdata\_ciscoinc\_frb5

db.createUser({user: "admin", pwd: "welcome", roles: [{role: "readWrite", db: "sqa\_onepdata\_ciscoinc\_frb5"}]})

db.createUser({user: "reporting", pwd: "password123", roles: [{role: "read", db: "sqa\_onepdata\_ciscoinc\_frb5"}]})

use sqa\_supdata\_ciscoinc\_frb5

db.createUser({user: "admin", pwd: "welcome", roles: [{role: "readWrite", db: "sqa\_supdata\_ciscoinc\_frb5"}]})

db.createUser({user: "reporting", pwd: "password123", roles: [{role: "read", db: "sqa\_supdata\_ciscoinc\_frb5"}]}) use sqa\_appdata\_ciscoinc\_frb5

db.createUser({user: "admin", pwd: "welcome", roles: [{role: "readWrite", db: "sqa\_appdata\_ciscoinc\_frb5"}]})

db.createUser({user: "reporting", pwd: "password123", roles: [{role: "read", db: "sqa\_appdata\_ciscoinc\_frb5"}]})

* Mongo Dump and Restore  
  nohup mongodump -u admin -p welcome --db sqa\_configdata\_frb1 --out /home/ec2-user/sqastk4/ &  
  nohup mongodump --db sqa\_supdata\_ciscoinc\_frb1 -u admin -p welcome --out /home/ec2-user/ &

nohup mongorestore -u admin -p welcome --db sqa\_configdata\_frb4 /home/ec2-user/sqastk4/sqa\_configdata\_frb1/ &  
nohup mongorestore -u admin -p welcome --db sqa\_supdata\_ciscoinc\_frb4 /home/ec2-user/sqastk4/sqa\_supdata\_ciscoinc\_frb1/ &

#for restore issue

grep -s safe \*json  
sed -i 's/,\"safe\":true//g' <filename>  
sed -i 's/,\"safe\":null//g' <filename>

* To Change Platform Version for tenants

db.core.tenants.find({"name":"ciscoinc", "systemProperties.expiredOn":ISODate("9999-01-01T00:00:00.000Z")}, {"\_id":1}).pretty()  
db.core.tenants.find({"\_id":ObjectId("532a9205d5a5edb6f00003e8")})  
db.core.tenants.update({"\_id":ObjectId("532a9205d5a5edb6f00003e8")},{$set : {"platformVersion":"frb5"}})

* Add reporting user in admin

1. use admin db.auth("devopsUserAdmin", "<password>")
2. db.grantRolesToUser("reporting",[ "read" , { role: "read", db: "sqa\_configdata\_frb5" } ])
3. db.grantRolesToUser("reporting",[ "read" , { role: "read", db: "sqa\_onepdata\_ciscoinc\_frb5" } ])
4. db.grantRolesToUser("reporting",[ "read" , { role: "read", db: "sqa\_supdata\_ciscoinc\_frb5" } ])
5. db.grantRolesToUser("reporting",[ "read" , { role: "read", db: "sqa\_appdata\_ciscoinc\_frb5" } ])

* Copy the postgress database

Connect Postgres with superUser for given environment (Eg PE : [pe-pdb.int.dev.ssi-cloud.com](http://ise-pdb.int.dev.ssi-cloud.com/)) :

Username: ssiadmin  
Password: letmein!

GRANT scheduler TO ssiadmin;  
GRANT workflow TO ssiadmin;

Also this can be done via connecting with respective user

* CREATE DATABASE sqa\_workflowdata\_stk4 WITH ENCODING = 'UTF8' OWNER = workflow;
* CREATE DATABASE sqa\_schedulerdata\_stk4 WITH ENCODING = 'UTF8' OWNER = scheduler;

Backup and Restore : (from devslave2)

pg\_dump -h [shared-pdb.int.dev.ssi-cloud.com](http://shared-pdb.int.dev.ssi-cloud.com/) -p 5432 -U scheduler sqa\_schedulerdata\_stk1 > sqa\_schedulerdata\_stk1\_dump  
psql -h [shared-pdb.int.dev.ssi-cloud.com](http://shared-pdb.int.dev.ssi-cloud.com/) -p 5432 -U scheduler sqa\_schedulerdata\_stk4 < sqa\_schedulerdata\_stk1\_dump

pg\_dump -h [shared-pdb.int.dev.ssi-cloud.com](http://shared-pdb.int.dev.ssi-cloud.com/) -p 5432 -U workflow sqa\_workflowdata\_stk1 > sqa\_workflowdata\_stk1\_dump  
psql -h [shared-pdb.int.dev.ssi-cloud.com](http://shared-pdb.int.dev.ssi-cloud.com/) -p 5432 -U workflow sqa\_workflowdata\_stk4 < sqa\_workflowdata\_stk1\_dump

Similar for Workflow

Cleanup of below tales is required

delete from qrtz\_cron\_triggers;  
delete from qrtz\_fired\_triggers;  
delete from qrtz\_simple\_triggers;  
delete from qrtz\_triggers;  
delete from qrtz\_job\_details;

SCH and DPS configuration

Please make sure to schedule these jobs with proper paltform version.

* Run metadataindexes on all new dbs (as we have restore dbs without mongo index - due to mismatch of mongo versions)

Copy latest "RST/scrubs/CT/metadataIndexes.js" file from [RST master](https://github.com/SSI-Avalon/RST/tree/master) branch to any machine where mongo client is present and execute below command:  
mongo <AUX\_DB> -u <AUXDB\_USERNAME> -p <AUX\_DB\_PASSWORD> -eval "var dbList=['dell#<ONEP\_DATABASE\_NAME>#<ONEP\_DB\_USER\_NAME>#<ONEP\_DB\_PASSWORD>'],auxDB='<AUX\_DB>'" metadataIndexes.js >> /home/ec2-user/createIndex.log

* Create new config management with the name sqa.stk4.frb4 (if already not created)

<https://github.com/SSI-Avalon/ConfigMgmt/pull/2208>

* Schema Upload and Collection Creation : core.users and core.tenants  
  [Create New Stack from existing Environment for RS](https://wiki.dev.ssi-cloud.com/display/OPSDOC/Create+New+Stack+from+existing+Environment+for+RS)  
    
  Command : sh updateConfigAndSolrIndex.sh -z [shared-zkp-1.int.dev.ssi-cloud.com](http://shared-zkp-1.int.dev.ssi-cloud.com/):2181/env/shared/solr/cluster1 -c core.tenants,core.users -p sqa.frb4. -d sqa/frb4/ -u <http://shared-sdb-1.int.dev.ssi-cloud.com:8080/solr> -m 1 -n 2 -s [shared-sdb-1.int.dev.ssi-cloud.com](http://shared-sdb-1.int.dev.ssi-cloud.com/):8080\_solr,[shared-sdb-2.int.dev.ssi-cloud.com](http://shared-sdb-2.int.dev.ssi-cloud.com/):8080\_solr -h /opt/solr-6.4.2/server/scripts/cloud-scripts/ -a CREATE
* Make changes in pods.json  
  <https://github.com/SSI-Avalon/chef/commit/5b99c54d6afc1829e12594c3e43e812d51bcd033>

Changes in Stack Config   
<https://github.com/SSI-Avalon/ConfigMgmt/pull/2209>

* Launch APPSVC3 stack
* command to launch stack

bin/orca deploy pod-component V2Dev SBX APPSVC2 STK2 -o SpotInstanceTypeConstraints/AlwaysCompute=false -o SpotInstanceTypeConstraints/OrderByPrice=false -o SpotInstanceTypes=m3.xlarge,c3.xlarge -o UseSpotFleet=false -o UseSpotInstances=true

* Perform semantic versioning
* Copy ranger folder

FROM : s3://srev-emr-test/sqa/stk1/repository

TO ; s3://srev-emr-test/<pod>/<version>/repository

* Copy lambda folder

FROM : s3://srev-emr-test/sqa/stk1/repository

TO : s3://srev-emr-test/<pod>/<version>/repository

* create database sbx\_hive\_stk2;  
  grant all privileges on sbx\_hive\_stk2.\* to hive@'%' identified by 'aUONVvg1BSseySTkV4Dp';  
  grant all privileges on sbx\_hive\_stk2.\* to hive@'%';  
  show grants for hive@'%';
* create database sbx\_hue\_stk2;  
  grant all privileges on sbx\_hue\_stk2.\* to hue@'%' identified by 'Zo3wgJTrsA4YuYTFZ89P';  
  grant all privileges on sbx\_hue\_stk2.\* to hue@'%';  
  show grants for hue@'%';
* create database sbx\_oozie\_stk2;  
  grant all privileges on sbx\_oozie\_stk2.\* to oozie@'%' identified by 'oozie';  
  grant all privileges on sbx\_oozie\_stk2.\* to oozie@'%';  
  show grants for oozie@'%';
* Launch EMR cluster

bin/orca deploy dataservices-emr-cluster V2Dev SBX DataServices-5.14.0 STK2

* Modify hive mysql metastore tables after cluster is ready

mysql> use sbx\_hive\_stk2;

mysql> alter table SERDE\_PARAMS MODIFY COLUMN PARAM\_VALUE text;

show columns **from** SERDE\_PARAMS;

* Create mongodb on hive by connecting through hue

> create database mongodb;

* Create ZKP job and Deploy ZKP
* Re-index rs.core.user and rs.core.tenant

./runmigration.sh solrIndexes '-Denv=sqa' '-DzkServer=[shared-zkp-1.int.dev.ssi-cloud.com](http://shared-zkp-1.int.dev.ssi-cloud.com/):2181' '[migration.tenant.name](http://migration.tenant.name/)=ciscoinc' '[solr.index.collection.name](http://solr.index.collection.name/)=core.tenants,core.users' '-Dversion=frb4'

* Provision Stack instances, exactly same what we have for 1.0
* ArchivePostDataLoadJob <https://wiki.ssi-cloud.com/display/OPSDOC/How+To+schedule+ArchivePostDataLoadJob>
* PurgeSchedulingHistoryJob <https://wiki.ssi-cloud.com/display/OPSDOC/How+to+purge+data+from+job_scheduling_history>
* UpdateParentPostDataLoadStatusJob <https://wiki.ssi-cloud.com/display/OPSDOC/DPS+Schedule+UpdateParentPostDataLoadStatusJob>
* ManifestFilePollingIterativeJob

[https://wiki.ssi-cloud.com/display/OPSDOC/HowTo%3A+Data+Pipeline+Services+%28DPS%29+Usage](https://wiki.ssi-cloud.com/display/OPSDOC/HowTo%3A+Data+Pipeline+Services+(DPS)+Usage)

* PostDataLoadJob

[https://wiki.ssi-cloud.com/display/OPSDOC/HowTo%3A+Data+Pipeline+Services+%28DPS%29+Usage](https://wiki.ssi-cloud.com/display/OPSDOC/HowTo%3A+Data+Pipeline+Services+(DPS)+Usage)

* Check valid artifactory repository.
* Prepare deploy job & ensure that should deploy on correct repository
* Test the end to end flow [Login, basic navigation, export]
* Deploy Implementations
* Create allservices deploy
* Deploy all services with RS branch

-------------------------------------------------------------------------------------------------------

**#ISSUE 53**

restarted solr nodes on pe

ssh -p 2222 ec2-user@pe-sdb2-19.int.dev.ssi-cloud.com

[http://pe-sdb2-8.int.dev.ssi-cloud.com:8080/solr/#/~cloud](http://pe-sdb2-8.int.dev.ssi-cloud.com:8080/solr/" \l "/~cloud)

sudo /etc/init.d/solr stop

sudo /etc/init.d/solr start

ps -ef | grep solr

sudo /etc/init.d/solr status

-------------------------------------------------------------------------------------------------------

**#ISSUE 53**

**changes in presto.json**

* added code in presto.json : chef/data\_bags/presto/presto.json

"i5glob" : {

"distribution": "starburst",

"root\_password": "hive",

"hive\_password": "X8WVLyNKXekW9wLOlcTn",

"connection\_url": "jdbc:mysql://dataservices-db.int.dev.ssi-cloud.com:3306/sbx\_hive\_i5glob?createDatabaseIfNotExist=true",

"metastore\_warehouse\_dir": "s3a://srev-emr-test/sbx/i5glob/dataservices/hive/warehouse/",

"metastore\_uri": "thrift://sbx-i5glob-metastore-1.int.dev.ssi-cloud.com:9083,thrift://sbx-i5glob-metastore-2.int.dev.ssi-cloud.com:9083",

"discovery\_uri" : "http://sbx-i5glob-presto-coordinator.int.dev.ssi-cloud.com:8889",

"discovery\_domain" : "dev.ssi-cloud.com",

"external\_url" : "https://sbx-i5glob-emr-presto-internal.dev.ssi-cloud.com",

"keystore\_pass" : "L7EOLZv1bpqh",

"ldap\_url" : "ldaps://10.105.6.22:636",

"ldap\_user\_base\_dn" : "DC=SERVICESOURCE,DC=COM",

"ldap\_group\_auth\_pattern" : "(&(objectClass=person)(sAMAccountName=${USER})(|(memberOf=CN=Renew-Ops-T1,CN=Users,DC=SERVICESOURCE,DC=com)(memberOf=CN=Renew-Ops-T2,CN=Users,DC=SERVICESOURCE,DC=com)(memberOf=CN=Renew-Ops-DevOps,CN=Users,DC=SERVICESOURCE,DC=com)(memberOf=CN=RenewEng,CN=Users,DC=SERVICESOURCE,DC=com)(memberOf=CN=Rev-Services-Eng,OU=Distribution Lists,OU=Exchange,DC=SERVICESOURCE,DC=com)(memberOf=CN=SecGrp - Data Services NALA,OU=DataServices,OU=Corp,OU=Security Groups,OU=SAC,DC=SERVICESOURCE,DC=com)(memberOf=CN=SecGrp - Data-Services,OU=DataServices,OU=Corp,OU=Security Groups,OU=SAC,DC=SERVICESOURCE,DC=com)(memberOf=CN=secgrp - Presto Users,OU=Security Groups,OU=SFO,DC=SERVICESOURCE,DC=com)))",

"fs\_awsAccessKeyId" : "AKIAIIQBO4JY3NRJ6ORQ",

"fs\_awsSecretAccessKey" : "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"

},

* merged the code
* changes in chef : push data to chef server using knife :
* knife data bag from file [presto](https://github.com/SSI-Avalon/chef/tree/master/data_bags/presto) presto.json
* knife data bag show presto [presto](https://github.com/SSI-Avalon/chef/tree/master/data_bags/presto)
* launch a stack on sbx - appsvc2 : For i5glob

bin/orca deploy pod-component V2Dev SBX APPSVC3 i5glob -o UseSpotInstances=false --capabilities CAPABILITY\_NAMED\_IAM

* export AWS\_PROFILE=v2dev
* RUN : sudo chef client
* created deploy job for implementations-sbx-i5glob