

S3 backup : EDL

What application team needs to provide to EDL admin:

- AWS Access Key ID &
- AWS Secret Key
- Container name
- Databases and: Table name that needs to be backed up.
- Schedule time : Daily(Time) / weekly (Day & Time)

How to get above mentioned things, Mike can help Application team to get these.

Go to Cloudera Manager in respective environment from where the backup needs to happen.

Cloudera → Administration → External Accounts → Click on Add access key Credentials

Add Access Key Credentials

Name *

Enter a friendly name to identify this credential.

AWS Access Key ID *

AWS Secret Key *

Cancel

Add

Name: Application code_ keys . Example: clsd_keys

Put the key id and secret key and click on ADD. Once added you can see the key in list like below:

External Accounts					
<div> AWS Credentials Altus Credentials Azure Credentials </div> <p>AWS Credentials allow CDH services and Cloudera tools to securely query data, browse data, backup and restore data/metadata, search metadata and view data lineage of data in Amazon S3. More Details</p> <div> Add Access Key Credentials Add IAM Role-based Authentication </div>					
Name	Type	Connectivity	Creation	Last Modified	
itx_aya	Access Key Credentials		November 3, 2018 1:26 PM	November 3, 2018 1:26 PM	Actions
itx_gora_agp	Access Key Credentials		November 20, 2018 10:45 AM	November 20, 2018 10:45 AM	Actions
gora_keys	Access Key Credentials		January 23, 2019 3:23 PM	April 5, 2019 3:56 PM	Actions
itx_jaware	Access Key Credentials		April 2, 2019 2:51 PM	April 2, 2019 2:51 PM	Actions
its_gbp	Access Key Credentials		April 2, 2019 2:52 PM	April 11, 2019 2:46 PM	Actions
gbp_keys	Access Key Credentials		April 5, 2019 3:57 PM	April 5, 2019 3:57 PM	Actions
gbp_new_keys	Access Key Credentials		April 11, 2019 3:51 PM	April 11, 2019 3:51 PM	Actions
clsd_keys	Access Key Credentials		May 28, 2019 1:35 PM	May 28, 2019 1:35 PM	Actions
Gora_new_Keys	Access Key Credentials		May 30, 2019 3:32 PM	May 30, 2019 3:32 PM	Actions
Align_keys	Access Key Credentials		June 11, 2019 5:13 PM	June 11, 2019 5:13 PM	Actions

Once the key is visible here that means the key is successfully added To EDL Cluster.

Now We need list of Database and table name that needs to be backed up.

Backup and Disaster Recovery :

Cloudera Manager → Backup → Replication Schedules → Create Schedule → Hive replication

Create HIVE Replication

General

Resources

Advanced

Name

Unique Name

Source

Hive (JnJ_EDL_NewProdCluster)

Use [AWS Credentials](#) or [Azure Credentials](#) to add more cloud replication sources.

Destination

itx_aya (Amazon S3)

Use [AWS Credentials](#) or [Azure Credentials](#) to add more cloud replication destinations.

Cloud Root Path

s3a://bucket/path

This field is required.

Databases

☒ Replicate All

Replication Option

☒ Metadata and Data

☐ Metadata Only

Replicate Hive metadata and the underlying HDFS data of the selected tables.

Schedule

Immediate

Run As Username

Default

Cancel

Save Schedule

Name: weekly_hive_clsd_bkp or daily_hive_clsd_bkp

Source: Will remain same as in Screenshot. EDL cluster (only when backing up not in restore process)

Destination: Will be the key which you added above in External accounts, Where you want the backup to be pushed on S3.

Cloud root path : s3a://its-edl-cls-backup-prd/current_backups/

In the above path : its-edl-cls-backup-prd changes to the s3 bucket name that application team gave. Everything remains the same. Please do not change anything other than container name.

Now untick the Replication all box. Then it will display boxes like below :

Databases	Database Name	Table Name or Regular Expression	<input type="checkbox"/> <input type="checkbox"/>
-----------	---------------	----------------------------------	---

For each table you have to mention database and table name, keep adding it from plus sign on the left.

Run as username: hdfs

No other changes are needed except mentioned above. As soon as you save schedule the job will run to copy files from cluster to S3. You can check it in Running commands on Cloudera manager.

Scheduling the BDR job to run on daily or weekly : CDSW

1: Login to CDSW with SAEDLMIA , if you do not have password please connect Ajai or Sharmila.

The screenshot shows the Cloudera Data Science Workbench (CDSW) interface. The top navigation bar includes a search bar and a 'New Project' button. The main content area displays a list of projects with their status (running) and last worked time. The projects listed are Jarvits, TEST, GSDL_Proto, xyz, delete, Audit_Jobs, EDL_BDR_S3, and metlog. The interface also shows a sidebar with navigation options like Projects, Jobs, Sessions, and Settings. The bottom right corner shows the version 1.3.0 (9bb84f6).

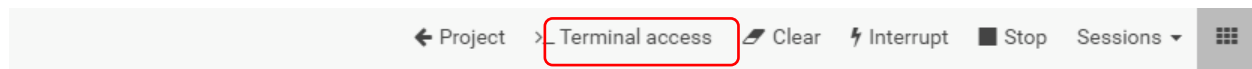
1: Create python script to make new current folders in the S3 container and does not over write the old ones.

Open workbench

Select smallest kernel available and launch session

Once the session is ready

Click on terminal access



Untitled Session

By saedlmia — Python 3 Session — 1 vCPU / 2 GiB Memory — just now

Collapse Share Running

Getting Started

This is your **Python 3 session**. Your **editor** is on the left and your **input prompt** is on the bottom.

To install a package type: `!pip3 install [package_name]` at the input prompt.

To execute code from the editor, select the code and execute it with `Command-Enter` on Mac or `Ctrl-Enter` on Windows. You can also enter code at the prompt below.

Use `?command` to get help on a particular command.

Run below commands in terminal :

Ls

And you can see :

```
cdsw@npjq3i3bf6m2bcxt:~$ ll
```

```
total 32
```

```
drwxrwx--x 8 cdsw cdsw 4096 Feb  7 20:47 aiops
```

```
-rwxrwx--x 1 cdsw cdsw 1367 Jan  2 21:34 analysis.py
```

```
drwxrwx--x 2 cdsw cdsw 4096 Apr 11 19:00 junk
```

```
-rwxrwx--x 1 cdsw cdsw  378 Jan  2 21:34 README.md
```

```
drwxrwx--x 2 cdsw cdsw 4096 Jan  2 21:34 seaborn-data
```

```
-rwxrwx--x 1 cdsw cdsw  422 Jan 31 03:04 Untitled1.ipynb
```

```
-rwxrwx--x 1 cdsw cdsw 881 Jan 31 03:08 Untitled.ipynb
-rwxrwx--x 1 cdsw cdsw 61 Apr 10 13:16 Untitled.py
cdsw@npjq3i3bf6m2bcxt:~$
```

cd aiops → cd Backups → cd scripts

```
dsw@npjq3i3bf6m2bcxt:~/aiops/Backups/scripts$ ll
```

```
total 64
```

```
-rwxrwx--x 1 cdsw cdsw 767 May 31 17:19 clsd_weekly_prod.py
-rwxrwx--x 1 cdsw cdsw 831 Mar 21 01:20 dumb.py
-rwxrwx--x 1 cdsw cdsw 644 Apr 23 16:35 EDL_dev_test.py
-rwxrwx--x 1 cdsw cdsw 221 Feb 7 20:44 edl.pk
-rwxrwx--x 1 cdsw cdsw 661 Apr 24 20:42 gbp_scheduled.py
-rwxrwx--x 1 cdsw cdsw 767 May 31 16:26 gora_prod.py
-rwxrwx--x 1 cdsw cdsw 835 Apr 3 15:14 iaware_prod.py
```

Create a copy of any of the files , please do not touch any existing file.

DO NOT MAKE CHANGES TO ANY SCRIPTS. Work only on a copy of a file.

Cp gbp_scheduled.py app_scheduled.py ; app is the application name for which you are creating the backup .

Below is the script : DO not touch anything except what is needed and mentioned :

```
import subprocess
```

```
import datetime
```

```
import sys
```

```
sys.path.insert(0, "/home/cds/aiops/Backups/scripts")
```

```
import util_repl
```

```
mv_cmd = "aws s3 mv s3://S3bucketname/current_backups/ " + \
```

```
"s3:// S3bucketname/current_backup_" + datetime.datetime.now().strftime("%Y_%m_%d") + \
```

```
" --recursive --sse --profile appcode"
```

```
def cmdexec(cmd):
```

```
# print cmd

proc = subprocess.Popen(cmd, stdout=subprocess.PIPE,stderr=subprocess.PIPE, shell=True)

(out, err) = proc.communicate()

# out = out.rstrip('\r\n')

print ("cmdexec\n", out, err)

return out, err

out, err = cmdexec(mv_cmd)

obj1 = util_repl.SrvrConnect('PROD')

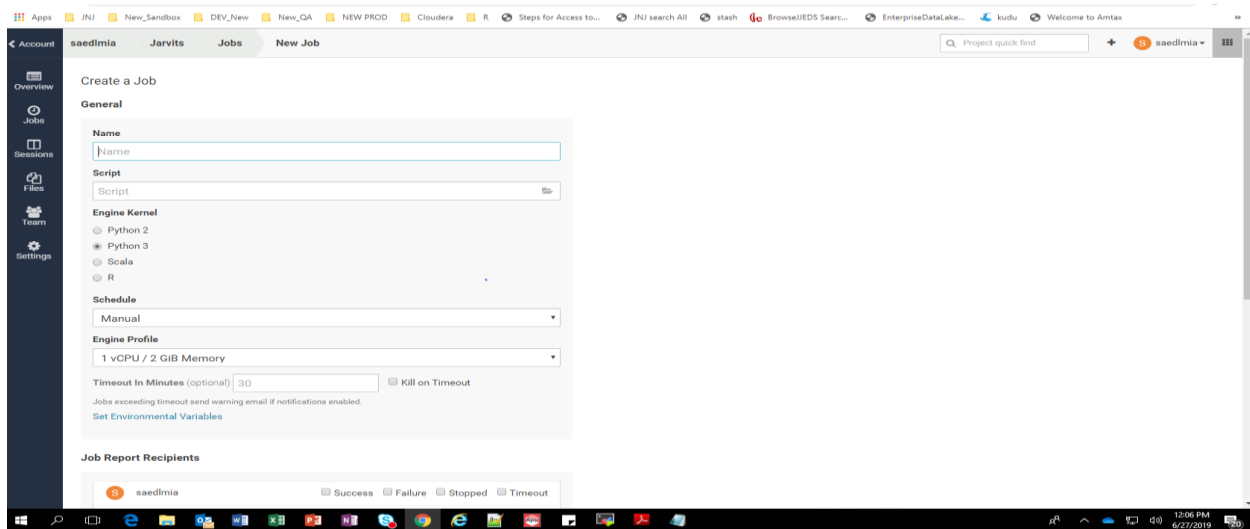
obj1.repl_run('hive', '165')
```

S3bucketname: This will be the S3 container which you put in BDR. Do not change anything in starting and ending of the lines. Only these needs to be changed.

appcode : This is supposed to be your appcode for which you are doing the Backup.

165 number is the replication schedule id ; this is the number you can get from cm when you create a schedule there. See the screenshot below .

Be careful, this number cannot be wrong.



1: Name : Backups_weekly_**Gora**_Prod or Daily_weekly_**Gora**_Prod

GORA in above line is app name , please use your app name for which you are doing the backup.

2: Script : aiops/Backups/scripts/gora_prod.py

3: Schedule : recurring and then schedule according to user demand.

Schedule

Recurring

Every week on Saturday at 09 : 00

Engine Profile

1 vCPU / 2 GiB Memory

Timeout In Minutes (optional) 30 ☐ Kill on Timeout

4: Add External Email : Add email of user to whom the job run details needs to be sent.

5: Untick the include console log. As user does not need to see the console logs.

Attachments

No attachment has been added yet.

Add Attachment

Please enter the path in the project files (relative to **/home/cdsw** in sessions).

☒ Include Console Log

Create the job this will trigger the job once in cluster and you can see the jobs in run commands in CM.

Once it completes as the user to check.

Go through this , it helps :

https://www.cloudera.com/documentation/enterprise/5-9-x/topics/cm_bdr_howto_hive.html#howto_backup_restore_hive_db

for proving the restoration is working to Application team. Just follow the steps till Cloudera part. No cdsw involved.

Do it in Dev create a table, back it up to a new folder on there S3 and then in BDR change the source to destination and destination to source. Do not run. Drop or rename the table in Dev and then run the opposite of what we do. Do a copy from s3 to EDL. And the table will be back.

Be very careful in any of the steps as this may effect the data of users.