f2mdbX Cluster
Amazon AWS

Document X1757-03 Rev 6, 2023

XtremeData Technologies Pvt. Ltd.
Bangalore, India

XtremeData, Inc.
Rolling Meadows, USA

www.xtremedata.com

f2mdbX Cluster – Amazon AWS

Document No. X1757-03 Rev 6, 2023

Contents

1 Introduction

This document outlines operational steps to create a single or multi-node, Amazon AWS cluster to deploy XtremeData's metadata product suite *nLite f2md Platform* on it, and subsequently use and manage the cluster. This document assumes that the reader is familiar with Amazon AWS technology (instances, EBS, S3, template, cloud formation, EC2, etc.) and has an AWS login account for deploying resources. Pre-requisites for creating the AWS cluster:

a) login account on Amazon AWS with privileges to deploy resources
b) access to AWS Marketplace AMI of XtremeData product
c) AWS template, from XtremeData, to create a single or multi-node cluster using the selected AWS instances
d) cluster authorization keys, from Amazon console, for creating and later connecting to the cluster through ssh or putty

The deployment and installation procedure outlined in this document deploys product components of XtremeData on the created AWS cluster. On successful completion of creation and launch of the AWS cluster it shall have the following XtremeData products installed on it for your use:

- ***f2mdbX***: backend server; distributed, high performance, analytics SQL engine supporting ANSI SQL with enhanced SQL-enabled functionality for metadata discovery and analysis directly from data lakes, cloud object storage files, HDFS, MAPR, on-premise file system, dbX tables & SQL queries
- ***nLite***: browser-based, no-code, GUI tool for metadata discovery and analysis
- ***sqldbx***: interactive command line utility to run metadata f2md SQL API
- ***xdudb***: command line utility to administer, monitor and manage f2mdbX server
- ***xdAdm***: browser enabled GUI to administer, monitor and manage f2mdbX server
- ***xdc***: Linux command line utility to administer, manage and monitor the AWS cluster from login account of *ec2-user*

You may launch a single-node cluster from AWS Marketplace console (section 2) or use cloud formation templates (sections 3 and 4) to create single or multi-node clusters. Launching the cluster from cloud formation templates is the preferred method of deployment. You may customize your installation with choice of AWS instance node types, authorization keys for the cluster and private data on S3. The cluster may be shut down when not in use and may be restarted later. As the cluster uses EBS volumes for persistent data, your data remains accessible for subsequent restarts of the cluster after stop.

For explanatory purposes and ease of understanding, this document includes a number of screen-shots generated from Amazon AWS console, which are all copyright of Amazon. The contents and layout of pages of AWS GUI and its command navigation are frequently modified by Amazon. The AWS screen-shots used in this document may not always be current. For more details on AWS console page and its commands refer to appropriate AWS documentation.

2 Single Node Cluster Launch from Marketplace AMI

This section describes the launch of a single node cluster directly from Marketplace AMI through AWS GUI without use of a cloud formation template. You must have a

login account on AWS with privileges to deploy resources. Login to your AWS web services account using your AWS account Id/role and password through your browser.

The steps to be followed are:

- Search *AWS Marketplace* using the string *nLite f2mdb*
- Select the AMI titled *XtremeData nLite f2md Platform*
- Provide a name of your choice for the single node cluster
- Enable public IP for the cluster
- Select one of the existing VPCs in your account
- Select authorization key pair (refer section 3.3) for the cluster
- Specify the required block storage (EBS); if EBS is used for storage it is recommended to use at least four devices with a minimum volume of 128 GB each; storage type for EBS may be gp3 or gp2 SSD
- Security group defaults should be used to allow ssh over port 22, and ports for tools (refer section 8) of the product namely, 2443 for metadata processing tool *nLite*, GUI admin through port 2400, port 8444 for *uitools* and port 8446 for command line terminals

Once you have provided all the above inputs, click on *Launch Instance* button of AWS GUI page. Cluster creation may take a little while and you may monitor its status through *Refresh* button on *EC2 Instances* page.

Once the single node cluster reaches the state *Running* and is ready you may *ssh* to the node using the private key of RSA key pair specified for cluster creation. Refer to section 5 for details of ssh remote connection to cluster.

After establishing remote connection to single node cluster, at the Linux command prompt of *ec2-user* you must type the following three commands to complete cluster creation with XtremeData product suite. The commands must be typed at Linux shell prompt, the second after completion of the first and third after completion of the second. For each command, the screen is updated with messages and returns to shell prompt of *ec2-user* after completion of the command.

```
xdc attach cloud +y
xdc provision +y +hdata --data 0
xdc bootable on
```

Once all three commands are successfully completed, the single node cluster launched from Marketplace AMI is ready for use. You may examine the status of the cluster as given in section 6 and continue with server creation as given in section 7.

Refer to section 9 to stop the cluster, section 10 to restart a stopped cluster and section 9 to terminate the cluster instance created.

3 Cloud Formation Template: Pre-Requisites

You must have a login account on AWS with privileges to deploy resources. Login to your AWS web services account using your AWS account Id/role and password through your browser.

3.1 Marketplace Access

If you are using XtremeData's AMI and template for the first time from your AWS account, this step is mandatory else this step may be skipped. This step is essentially to conform to Amazon's legal requirements to access images deployed on marketplace by enabling the software for access from your AWS account.

From AWS marketplace, select the image titled "*XtremeData nLite f2md*" and proceed as though you are deploying the image directly from marketplace. Navigate through the GUI, and accept the terms and conditions outlined by Amazon. Once you have gone past the terms and conditions step, abort the deployment and use the predefined template provided by XtremeData for product deployment.

This step needs to be performed only the first time that you deploy XtremeData image from marketplace. If you do not perform this step, then an error would be raised during product deployment.

3.2 Cluster AMI and Template

You must first download the *nLite f2md Platform* cluster template from Amazon AWS using the following URL in your browser:

https://s3.amazonaws.com/xtremedata-dbx-public/AWS-newVPC_f2mdbx_cluster.template

The above template is to be used for cluster creation into a new VPC with NAT and S3 endpoint. If you already have an existing VPC, and would like to deploy *nLite f2md Platform* on it then use the template provided at this URL.

https://s3.amazonaws.com/xtremedata-dbx-public/AWS-existVPC_f2mdbx_cluster.template

Both the above templates from XtremeData use EBS (1 TB per node) for persistent data storage and, by default, create an AWS cluster of just one AWS node instance of the selected instance type. In order to deploy in an existing VPC, you have to provide additional parameters during cluster stack creation through AWS dialogs.

If you plan to process large volume data, metadata discovery performance may be enhanced by scaling up the AWS cluster with multiple nodes. In order to scale up the cluster from a single node, the template must be modified. As this could be an error prone task for the average user, XtremeData provides a scale-up script that you may download and modify the template. Follow the steps outlined in section 3.2.1 to create a template for a multi-node cluster.

Remote access to the cluster is only through the head node of the cluster using its public IP that is generated by AWS when the cluster is created or started. The public IP of the head node changes every time the cluster is stopped and restarted.

3.2.1 Cluster Nodes Scale-up

This step may be skipped if you are satisfied with the default single node cluster provided by the downloaded template.

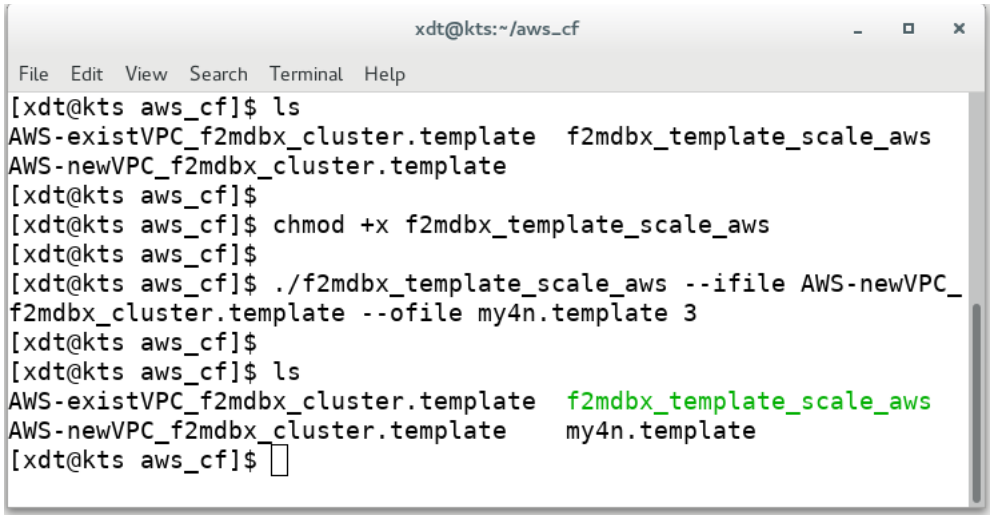
In order to scale-up the nodes of the cluster, download the Python script provided by XtremeData at the following URL:

https://s3.amazonaws.com/xtremedata-dbx-public/f2mdbx_template_scale_aws

The Python script must be run on your desktop to modify the downloaded template for scaling up nodes of your f2mdbX cluster, and it must be done before cluster creation. You must specify as input to the script the following parameters

- name of the downloaded template (option *--ifile*)
- name of the new template (option *--ofile*) to be generated
- number of data nodes in the cluster; for a cluster of *N* nodes, the number must be specified as *(N-1)*.

On a Linux desktop you must first grant execute permission to the downloaded script, and then run the script to scale-up nodes in cluster template. The Linux desktop screen shown below displays contents of current path (*ls*), changes execute privilege (*chmod*), runs the script and finally displays current path contents (*ls*) again. The scale-up script execution shown below, assumes that you are generating a new template named *my4n.template* with a total cluster size of 4 nodes.



Once you have scaled up the cluster template to the required size, you may discard the downloaded template and use the newly generated template for cluster creation. Cluster scale up may be done for both scripts of section 3, to deploy with a new VPC or in an existing VPC.

3.3 Authorization Key for Cluster Access

Access to f2mdbX cluster from remote terminals (either using *ssh* from Linux desktops, or *PuTTY* from Windows desktops) uses public-private RSA keys pair. The public key remains on the AWS cluster and the private key must be used by the remote user.

Your AWS sign-in account, would already have such a key pair defined. If you do not wish to use the key pair defined for the sign-in role, you may create a new key pair for the f2mdbX cluster based on another AWS role (provided your sign-in role has the necessary privileges). Such an additional key-pair must be created prior to f2mdbX cluster deployment, as this key would be required during AWS cluster creation.

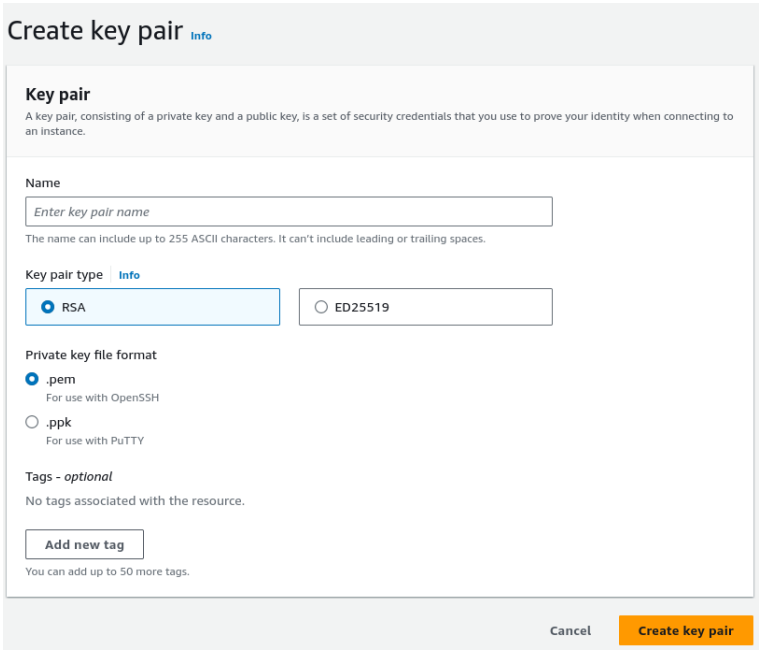
This step is not mandatory and is to be performed only if you wish to change the authorization key-pair for f2mdbX cluster from the default one available for your sign-in account.

You must be logged in to your AWS account and viewing the *EC2 Dashboard* page of AWS. You may create a new key-pair for f2mdbX cluster using the following steps on AWS console:

a) In the *EC2 Dashboards* page scroll down in menu bar on left of screen and click on *Key Pairs* under *Network & Security*.

b) The AWS console page is updated to display available keys for your account. Click on button labeled *Create key pair*.

c) Key pair creation dialog appears; type of key pair must be *RSA*; *PEM* keys are used by Linux *ssh* and *PPK* keys by Windows *PuTTY*; fill in appropriate values and click on button *Create key pair*.



d) On key creation, a dialog to save the key file appears; save key file in a suitable path for future use. If you do not save it now, key file cannot be retrieved later.

e) The new key created appears in the list of keys defined for the AWS account.

4 Cluster Launch: Marketplace AMI with Templates

If you are not launching a cluster with cloud formation template, you may skip this section.

You are now ready to create and deploy the cluster by launching the template of section 3.2 using XtremeData's AMI. If you are not already signed in to your Amazon account, you must sign in to the account by providing your login credentials and also have necessary privileges for cluster creation.

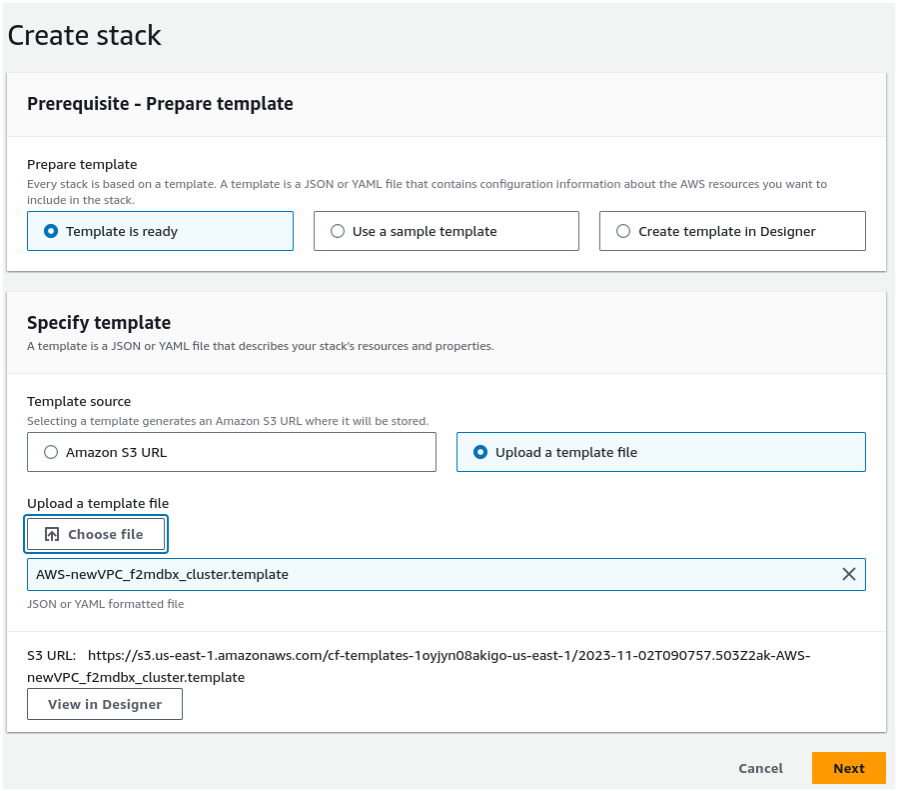
The subsequent numbered steps of this section walk you through the steps required to create the AWS cluster for f2mdbX installation.

1) Select *CloudFormation* page from *Services*.

2) In the AWS console click on button labeled *Create stack* and choose *With new resources (standard)* from pull-down menu.

3) The browser displays the *Create stack* page to specify parameters for the stack to be created.

4) By default, radio button labeled *Template is ready* is already selected; enable the radio button labeled *Upload a template file* and click on *Choose file* button to select the template (section 3.2) stored on your system running the browser: either a single node template or scaled-up multi-node template for a cluster deployed with new or existing VPC.



Once you have chosen the template file (say, new VPC template of section 3.2), the dialog is updated with name of the template file. Click on button *Next*.

5) A new dialog as shown below appears and several values have to be specified in the dialog box. The first step is to assign a name to your new cluster.

   Type in the name of your choice in the edit controlled labeled *Stack name*; name assigned is *f2md-cluster*.

6) Next, you must choose the type of instance that is to be provisioned for nodes of your cluster; all cluster nodes would use the same AWS instance type. Click on combo-box labeled *InstanceType* to display the type of AWS instances that the template is configured for.

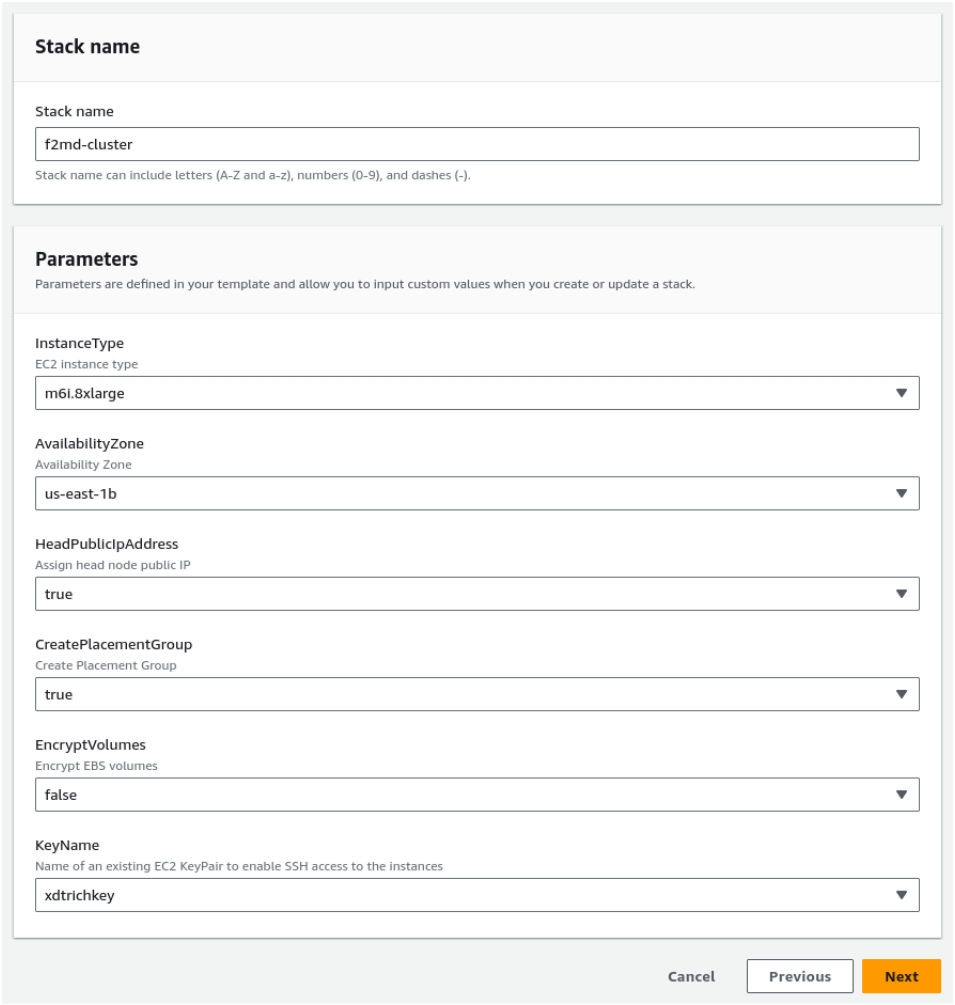
The cloud formation template of XtremeData is configured to choose from the following general purpose or compute optimized AWS instances:

m5.4xlarge, m5.8xlarge,
m6i.4xlarge, m6i.8xlarge, m6i.12xlarge, m6i.16xlarge
c6i.8xlarge, c6i.12xlarge, c6i.16xlarge

For example, the instance *m6i.8xlarge* with 32 vCPUs and 128 GB memory may be selected for moderate sized data less than a TB. Select one of the displayed instances by clicking on it.

7) Select the zone where your cluster is to be deployed, by clicking on the combo-box labeled *AvailabilityZone*, for example *us-east1-b*. Zone selection is not applicable when deploying in an existing VPC as the zone would be chosen from VPC.

8) By default, combo-box labeled *HeadPublicIpAddress* is selected to be *true*; it must be so for a remote client to access your cluster using the head node IP in browser URL or to ssh to the cluster head node. If you are creating a multi-node cluster, only the head node gets a public IP not the data nodes.



9) Selection for combo-box labeled *CreatePlacementGroup* may be *true* or *false*; making it *true* may provide better performance on the cluster; set it to *true*.

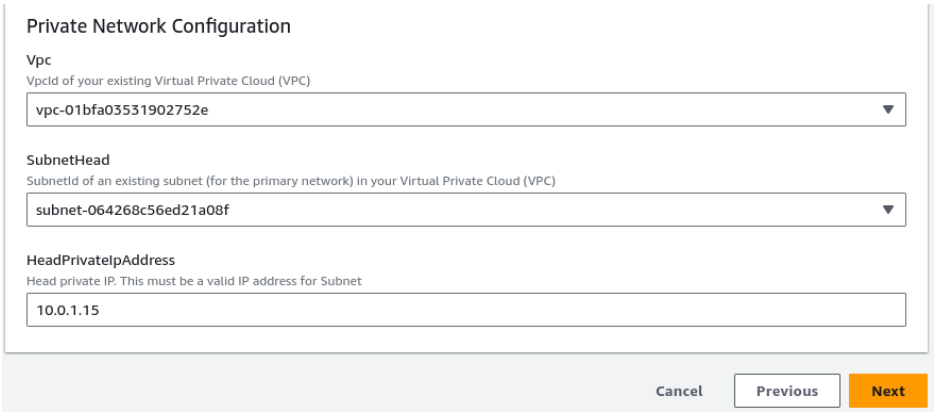
10) Selection for combo-box labeled *EncryptVolumes* must be made *false*; if selection is *true*, then performance may degrade; set it to *false*.

11) You must now select the remote access authorization key for the new cluster to be created. Click on combo-box labeled *KeyName* to display names of all keys available in your AWS account. If you had created a new key-pair following steps of section 3.3, the name of the key file would be displayed in the combo-box. Pick the appropriate key for the cluster.

12) If you are using *AWS-newVPC_f2mdbx_cluster.template* (or its scaled-up version) to deploy the cluster in a new VPC, this step is not applicable and may be skipped.

    If you are using AWS-*existVPC_f2mdbX_cluster.template* (or its scaled-up version) to deploy cluster in an already existing VPC, dialog displayed in step 8 is extended further to provide parameters of existing VPC.

    If the existing VPC template is for a single node cluster, dialog extension is as below.



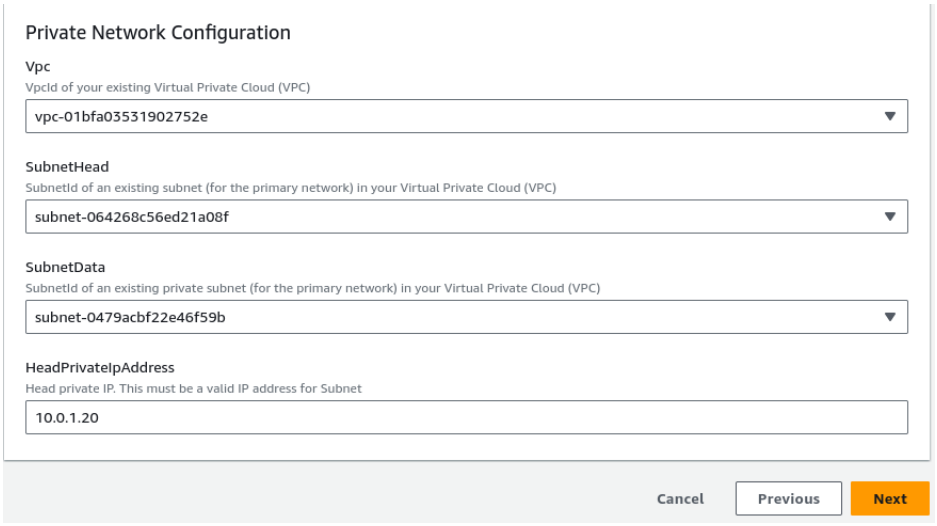
    (a) Select one of the available and existing VPCs in your account by clicking on combo-box labeled *Vpc*.
    (b) The selected VPC must have a public and private subnet; select the subnet by its public ID (*not* private ID) in combo-box labeled *SubnetHead*. If selected VPC does not have a public subnet or you choose a private subnet, *ssh* from remote systems may not be possible to your cluster. For existing VPC deployment, remotely accessible public IP will be generated for cluster head node only if checkbox *HeadPublicIpAddress* of step 8 is enabled and a public subnet is chosen for *SubnetHead*.
    (c) You must now specify a Head private IP for the subnet in edit box labeled *HeadPrivateIpAddress*. The IP specified here must not already be used by the subnet and be within the subnet range.

    If the existing VPC template is for a multi-node cluster dialog extension is as shown below. You may

    (d) Select an existing VPC as in step 12.(a) above.

(e) For a multi-node cluster two subnets have to be chosen: one for the head node and one for the data nodes. As in step 12.(b) above, choose a public subnet for head node *SubnetHead*. If you intend to do metadata processing in a hybrid environment using source data in clouds of vendors other than AWS, then choose a private subnet for data nodes under *SubnetData*. If both subnets are chosen as public subnets, your data nodes cannot access object stores of other cloud platforms except AWS

(f) Specify an IP for *HeadPrivateIpAddress* as in 12.(c) above.



When there is no accessible public IP for head node, either by disabling *HeadPublicIpAddress* or by deployment in an existing VPC using private subnet, you may have to use port forwarding or other techniques suggested by your network admin.

13) Once you have specified all the required values for dialog box displayed in step 8, click on button labeled *Next*.

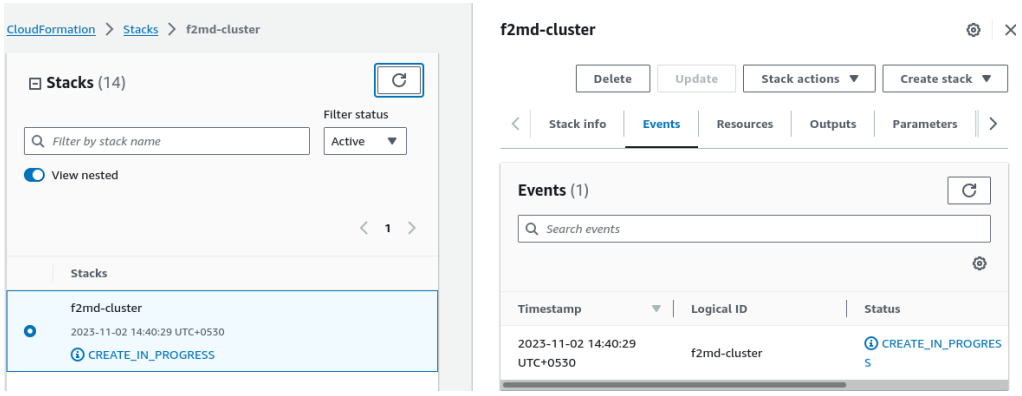
14) A new dialog box for *Configure stack options* and *Advanced Options* is displayed. There is nothing to be modified in this dialog; click on *Next*.

15) Yet another dialog box for you to review your selections and specified values appears. If you are satisfied with your input data, nothing is to be modified in this dialog; scroll down in the dialog and click on button labeled *Submit* at the end of the dialog page.

16) The screen is updated as shown below to display creation status of AWS cluster stack *f2md-cluster*. Cluster stack creation may take a few minutes for all nodes of a multi-node cluster to be created. The time depends on several factors including the type of instance that you have deployed, load on AWS zone, resource availability, etc.

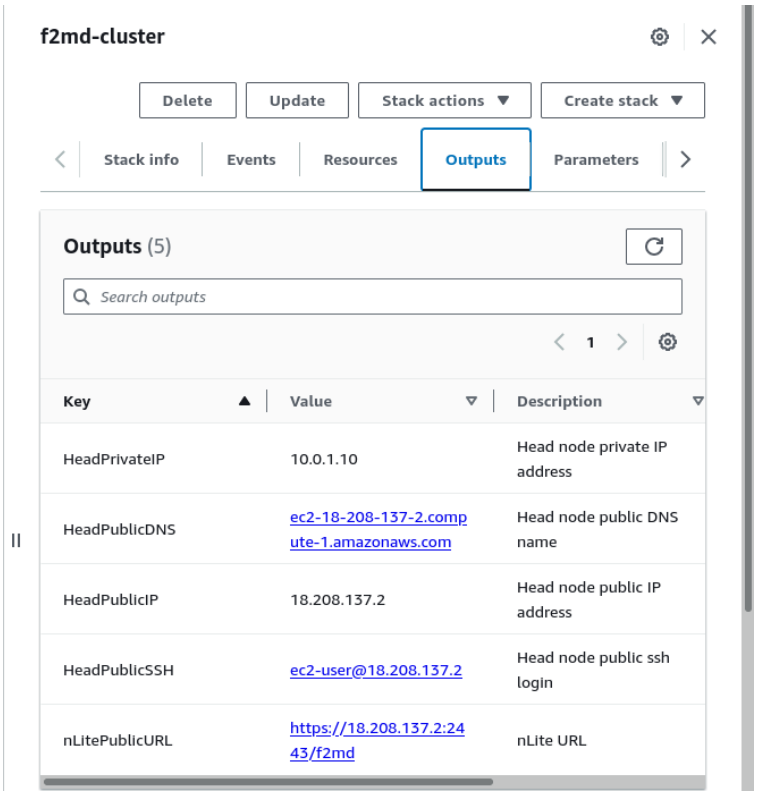
Cluster creation passes through several stages displayed in column labeled *Status* in web page shown below, after a click on *Events* tab.

You may monitor the status of cluster creation by periodically clicking on the *Refresh* button of the page.



17) On completion of cluster creation, a click on *Refresh* icon displays the status *CREATE COMPLETE* in column labeled *Status* for all nodes of your cluster.

18) If you have enabled *HeadPublicIpAddress* in step 8, you may note the public IP by clicking on tab labeled *Outputs*.



Note down the IP listed for *HeadPublicIP* (18.208.137.2 in above figure, and also available in *EC2 Instances* page) as this is essential for remote access to head node of your AWS cluster, browser URLs of GUI tools and for subsequent cluster management.

You can connect to head node of running AWS cluster using the Linux user *ec2-user* and head node IP, namely 18.208.137.2, as *ec2-user@18.208.137.2*. In a multi-node cluster, you cannot connect to data nodes of f2mdbX cluster as they

have no public IPs. On every restart of the cluster after stopping the cluster, the public IP of head node would be changed.

The AWS cluster is now created with tools of *nLite f2md Platform* installed on it and is ready for use, with two Linux users: sudo privileged user *ec2-user* by AWS and ordinary user *dbxdba* by XtremeData. Once the cluster is running, you may verify the status of the installation as discussed in section 6. In order to verify cluster status, you must first connect to head node of the cluster as described in section 5.

5 Remote Connection to Cluster

You are now ready to connect remotely to the cluster from your desktop using either

- *ssh* from Linux desktops or
- *PuTTY* from Windows desktops

The private key of RSA key pair used for cluster creation in step 11 of section 4 must be available in either PEM (for Linux desktop) or PPK (for Windows desktop through PuTTY) formats.

a) If you are using a Linux desktop, and connecting through ssh:

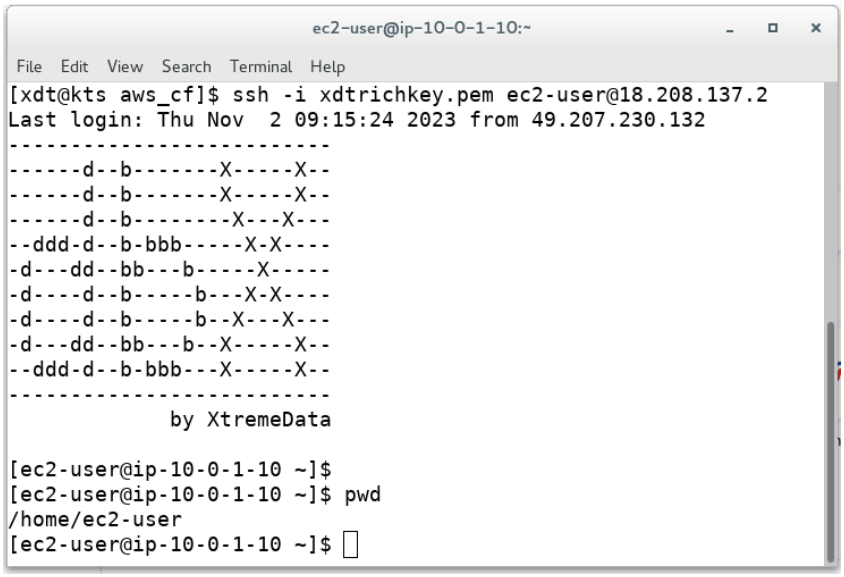
- The RSA key file with extension pem must be made read only:

```
chmod 400 <full_path>/<key_filename>.pem
```

- To connect to head node of running AWS cluster, at Linux shell prompt type the following ssh command:

```
ssh -i <full_path>/<key_filename>.pem ec2-user@<head IP>
```

The IP address *<head_IP>* is the one obtained at step 18 of section 4. The figure below shows connecting to AWS cluster with public IP 18.208.137.2 from a Linux desktop using private key *xdtrichkey.pem* through *ssh*. On successful connection, your terminal becomes a Linux shell window on AWS cluster head node of user *ec2-user* as below.



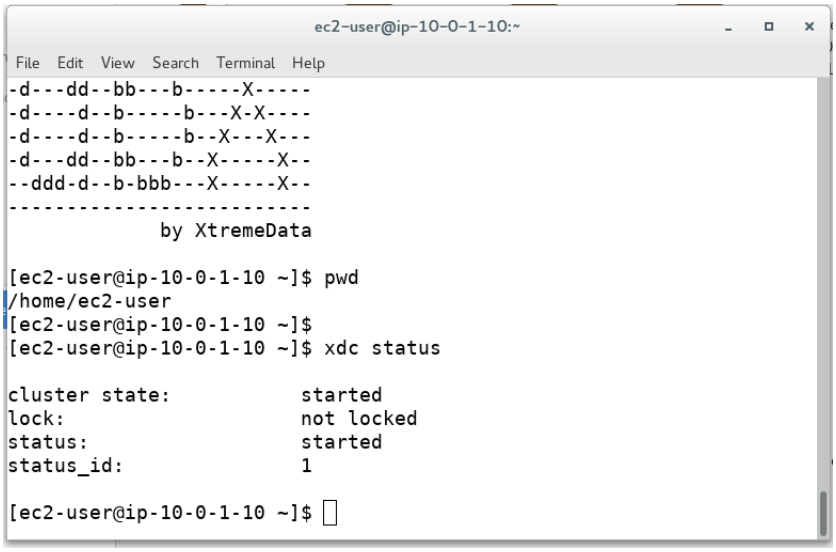
b) If you are using a Windows desktop and connecting through PuTTY:

- Set the PPK format key file in PuTTY using *Session*, *Auth* and *Browse* commands.
- After selecting the right key file, type in the IP of cluster head node (<*head_IP*> of step 18 of section 4) in edit control of *Session* dialog.
- Click on button *Open*.
- PuTTY dialog closes and a new terminal appears. Type in *ec2-user* in the new terminal for login Id and hit *RETURN*.

A shell window similar to Linux ssh is displayed by PuTTY connection.

6 Cluster Status

XtremeData supports a cluster services command *xdc*, to examine the status of the cluster, with a suitable option *status*, at shell prompt of *ec2-user*. The final state of a successfully created and running cluster must be *started*.



During cluster creation and launch, the cluster undergoes several state transitions such as *new*, *stopped*, *configured*, *transitional*, *failed*, etc. If you connect to the head node, during or after creation of head node but before completion of creation, *xdc status* command may not be the same as *started*, which is the final and ready state.

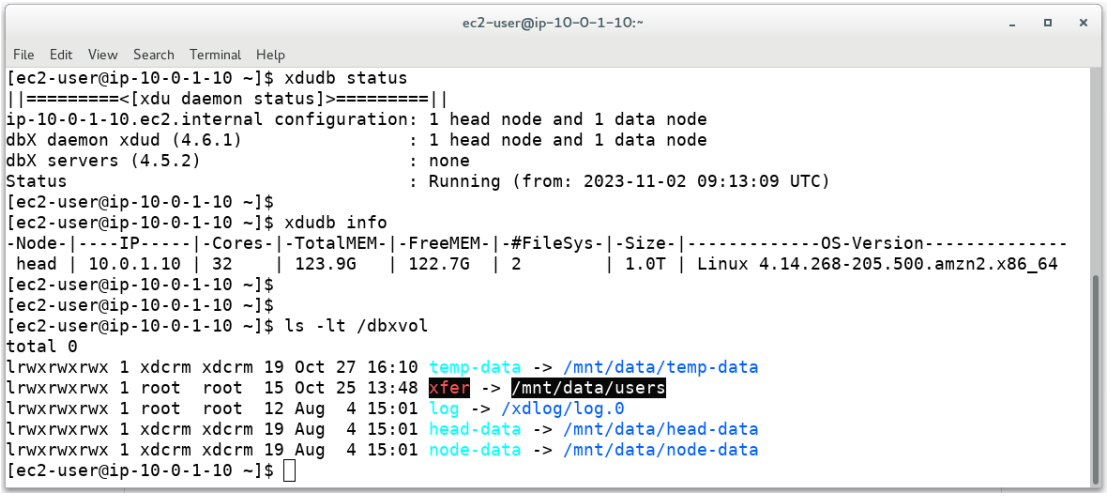
Instead, you may see the status in one of its intermediary states. To reach the final ready state, the system may take a while and this duration may vary according to the number of cluster nodes and EBS volume size. Repeat *xdc status* command at shell prompt until *status* becomes *started*.

On successful completion of a multi-node cluster, all nodes in the cluster must be in state *started* for command *xdc status*. Based on cloud resource constraints, it may take a while for all nodes to reach status *started*.

If all nodes do not reach such a state, or some node reports *failed*, the recommended course of action is to delete the current deployment and deploy a new stack.

Once the cluster is successfully started, minimal daemon services on the cluster are also up and running. As shown in the figure below, you may obtain

- status from daemon *xdu* using f2mdbX command *xdudb status*
- information on nodes of cluster using f2mdbX tool command *xdudb info*
- examine paths set up for f2mdbX server using Linux command *ls -lt*



Note that none of the status commands shown above will work correctly until *xdc status* reports *started*. As no f2mdbX server has been set up on the new cluster, *xdudb status* command shows *none* for any f2mdbX server on the cluster.

7 f2mdbX Server Creation

Once the cluster is successfully deployed with *nLite f2md* product suite installed on it through procedures of section 2 or section 4, the first step before using tools of the product suite is to create a f2mdbX server to run your commands through GUI tools or shell commands.

XtremeData provides a command line tool, *setupserver*, to automate creation of a properly configured f2mdbX server based on the resources of the cluster with EBS storage. Without creating a server through *setupserver*, you cannot use any of the metadata tools of *nLite f2md Platform*. As a first step, *setupserver* must be run at the shell prompt of the privileged user *ec2-user* created by AWS.

Initial access to the cluster to run *setupserver* is a remote connection through ssh or Putty as described in section 5. Once you are connected to the cluster you would be at the shell prompt of *ec2-user*. Type the command *setupserver* at Linux shell prompt. You would be prompted for input of the following items:

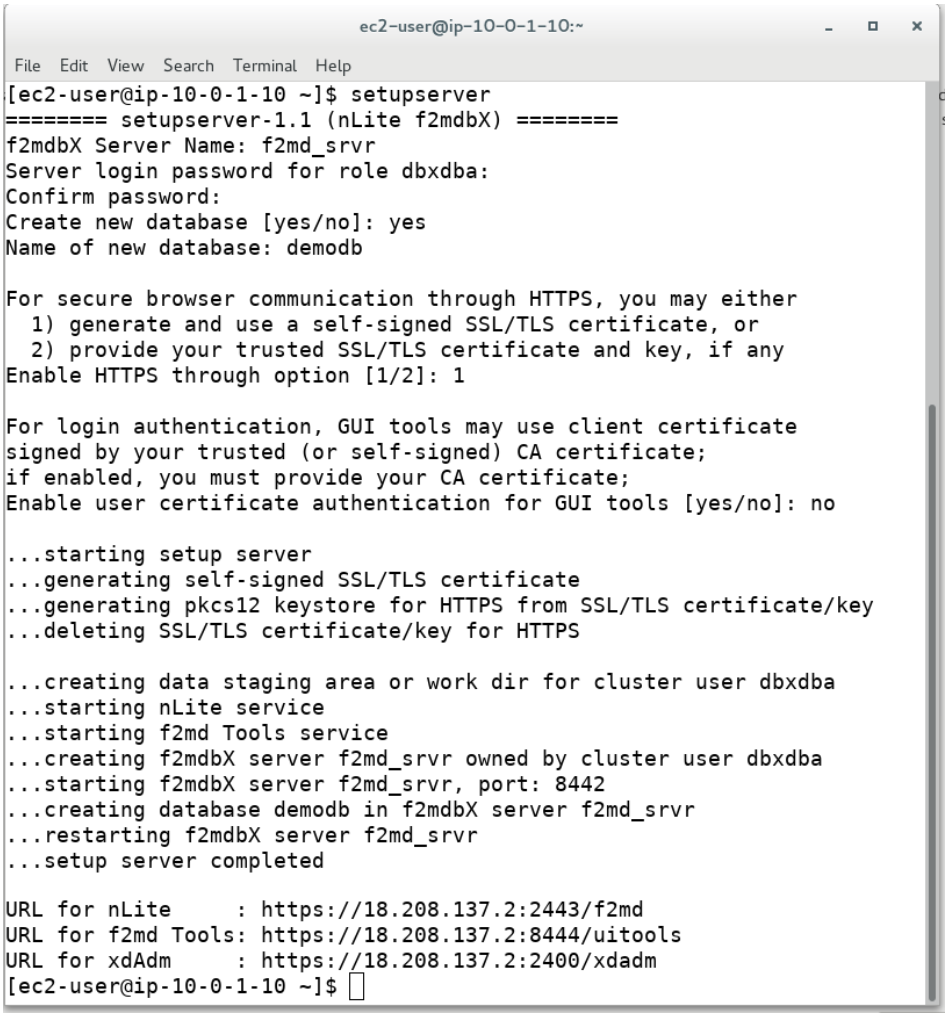
a) name of the new f2mdbX server to be created
b) server login password for role *dbxdba*, owner of the server
c) optionally, name of the new database to be created in the server
d) a certificate and its private key for secure browser communication through https; you may provide your own trusted certificate or generate a self-signed SSL/TLS certificate for https
e) optionally, your CA certificate for client login authentication of GUI tools

Items (c) and (e) are optional and may be skipped. Items (a) and (b) are mandatory, and you must choose one of the two supported options for (d): your own trusted certificate or on-the-fly generation of a self-signed certificate.

The password provided for (b) is essential for connecting to server through GUI tools or shell commands. If lost, the password cannot be retrieved; it may be changed after role login. Password should be 5 to 30 chars long and must be confirmed by retyping.

Any certificate/key provided for items (d) or (e) must be in PEM format. Prior to running *setupserver*, using Linux command *scp* or Windows PuTTY from your local system, the appropriate files of the certificate/key must be transferred to the cluster for storage in a path accessible to *ec2-user*.

Any certificate and key provided for (d) are not stored in the system by *setupserver*, but used by it to create a *keystore* file for https. If you choose to enable client login authentication using certificates, then item (e) is stored in encrypted form. For authentication at the time of GUI tool login, you must provide to the client a user certificate signed (e.g., through *openssl*) using same CA certificate provided for (e).



The screen image of *setupserver* execution given above shows the creation of f2mdbX server with the following attributes

- ✓ server named *f2md_srvr*
- ✓ server login password for role *dbxdba* (not displayed on screen)
- ✓ an empty database in f2md_srvr named *demodb*
- ✓ generation of self-signed SSL/TLS certificate for secure https communication
- ✓ disabling client authentication using certificates

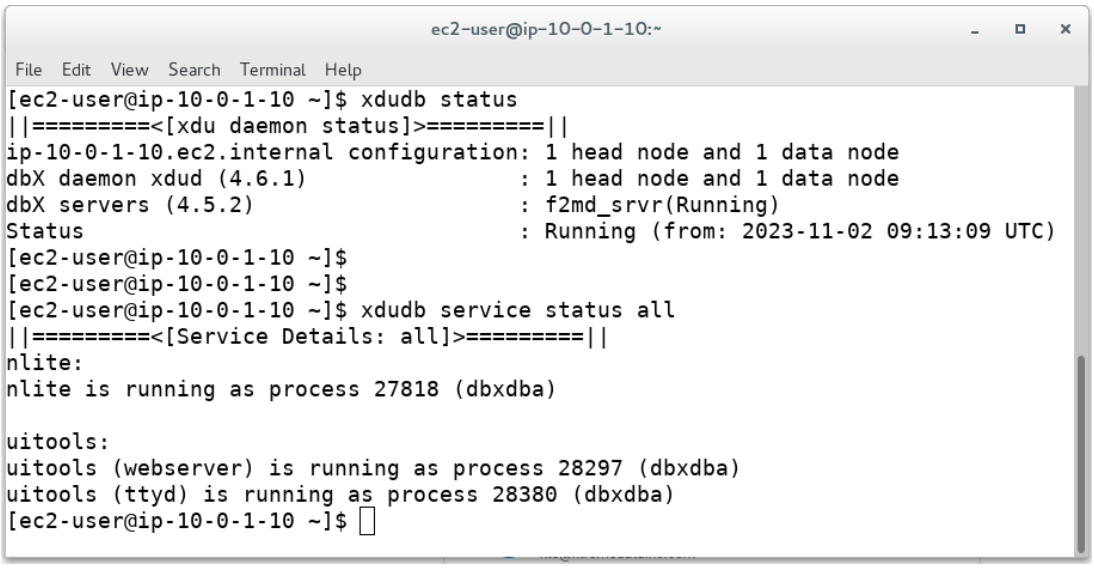
Once the input is provided, *setupserver* creates the server with specified attributes and displays suitable messages about the steps of server creation. Finally, three URLs for the three GUI tools of *nLite f2md Platform* are displayed and the newly created server is started for your use. The above screen image assumes that the new VPC template with a public IP was used, as it displays head IP in URLs to access the tools. If the cluster was deployed in an existing VPC with a private subnet for head node, remote access through IP based URL may be forbidden.

With regards to running of *setupserver* at shell prompt for f2mdbX server creation

- you must be logged in to the cluster as *ec2-user*; no other Linux user, including *root* or *dbxdba*, can run *setupserver*
- *setupserver* does not create multiple f2mdbX servers on the cluster
- after successful server creation, a second server may be created through *setupserver* only after dropping the existing server
- any certificate or key file that you provide must be valid and be in PEM format

7.1 Server Status

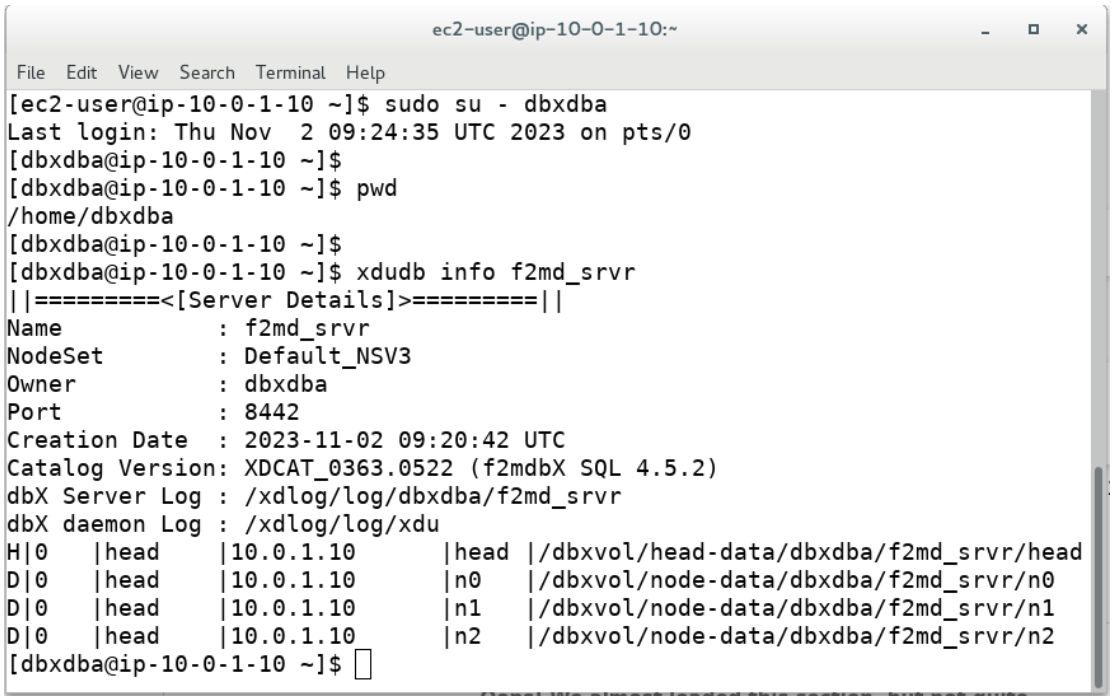
After server creation, logged in as *ec2-user*, you may obtain status of f2mdbX services running on the cluster using *xdudb* commands as shown below. Unlike section 6, name of the f2mdbX server, *f2md_server*, is displayed for command *xdudb status*. The second command *xdudb service* displays status of services running for tools *nLite* (section 8.1) and *uitools* (section 8.6).



Details of server *f2md_srvr* created and launched on the cluster may be obtained using *info* command of *xdudb*. The command cannot be executed as *ec2-user* and you must become the ordinary user, *dbxdba* using *sudo* command of Linux as shown in the figure given below.

When you are logged in as *ec2-user*, you may change the default password of Linux user *dbxdba* using the following Linux command at shell prompt.

sudo passwd dbxdba

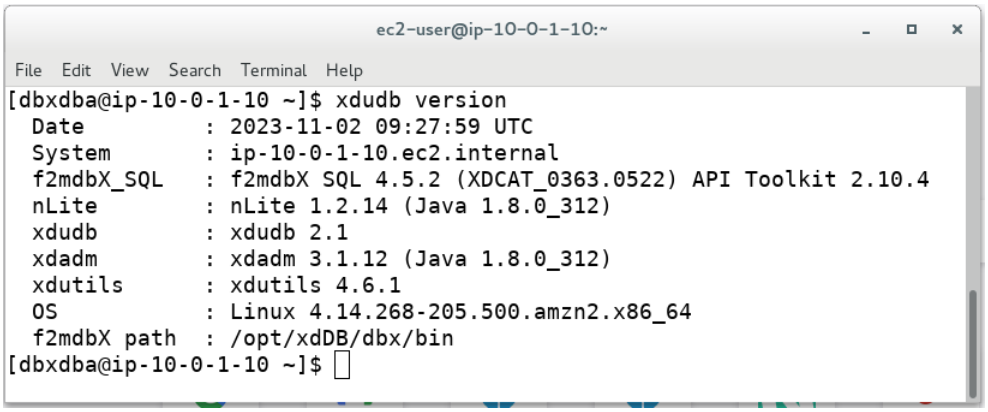


8 f2mdbX Tools

Once XtremeData product installation from AMI, its set up and configuration through *setupserver* (section 7) are completed, you may use metadata services of the product. Three types of services are supported by tools included in *nLite f2md Platform*

- *Metadata Tools*: for metadata discovery, analysis and SQL querying either through GUI tool *nLite* or command line tool *sqldbx*
- *Server Tools*: to administer, manage and monitor f2mdbX servers through command line tool *xdudb* or GUI tool *xdAdm*
- *Cluster Management:* to monitor and administer XtremeData products installed on the cluster

The figure below shows versions of product tools reported by f2mdbX tool *xdudb* (section 8.3) through its command *version*.



Sections 8.1 to 8.5 briefly outline the functionality and use of these tools on the cluster. For more details refer to appropriate tool documentation of XtremeData. Section 8.6 outlines browser methods to access the tools.

8.1 Metadata Tools: nLite

nLite is a browser-based GUI product, that supports a *no-code* approach to derive *insights* on source datasets, residing across hybrid or multi-cloud platforms, for managing data quality, data correctness, migration, data pipelines, privacy, data governance, analytics, machine learning, AI and also augment existing data catalogs.

It supports *full fidelity metadata* (*f2md*) for generation of metadata on source data residing in data lakes, object stores, HDFS, MAPR, web and on-premise files, database tables and queries. Source data may be in formats of TEXT, CSV, Parquet, ORC and AVRO.

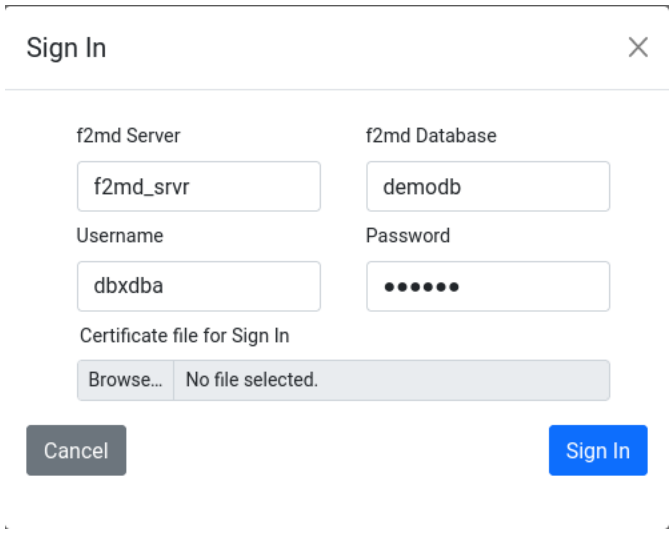
Features of the GUI product include drill down navigation, generation of graphical charts, PDF reports and automated scripts for long running production jobs, offers the full functionality repertoire of its backend engine, and more: data quality dimensional analysis and metrics, file metadata processing for dedupe classification, object store data preview, background script runs, etc.

The product *nLite* may be run through a browser using the URL displayed by *setupserver* during f2mdbX server creation (section 7); an example:

https://18.208.137.2:2443/f2md

The above URL assumes that the public IP of your cluster is 18.208.137.2 and 2443 is the port for communication. An alternative method to using the above URL to run *nLite* is given in section 8.6.

A dialog box for logging in to f2mdbX server running on the cluster, specifying server name, database name, role name, role password and user certificate (if any) is displayed. Refer to *nLite User Guide* for product documentation on *nLite*.



8.2 Metadata Tools: sqldbx

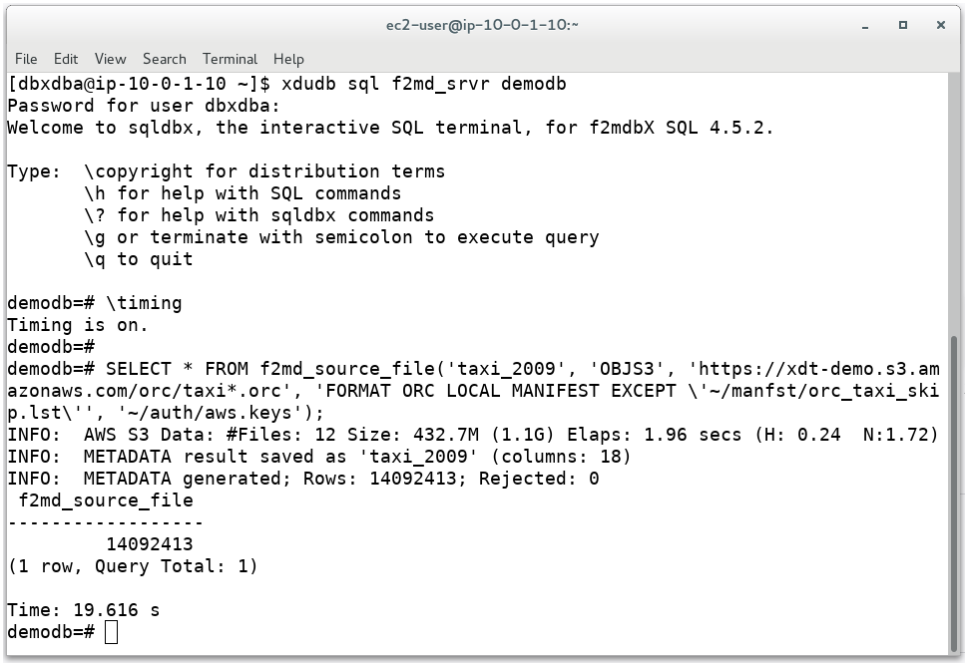
sqldbx is an interactive command line utility for running *f2md Toolkit API* functions and SQL queries for metadata processing. The utility must be run on head node of the cluster after connecting to the head node, or as outlined in section 8.6. *sqldbx* cannot be run when you are logged in as *ec2-user*, and you must switch to become one of the non-root users (e.g.: *dbxdba*) with a DB role.

The figure below runs *sqldbx* on cluster head node using *xdudb* (section 8.3), as user *dbxdba*, connecting to database *demodb* with login password of role *dbxdba*. The window below runs a query to display f2mdbX version deployed on the cluster and the command *\d* to list any relations found in *demodb*.

```
ec2-user@ip-10-0-1-10:~
File  Edit  View  Search  Terminal  Help
[dbxdba@ip-10-0-1-10 ~]$ xdudb sql f2md_srvr demodb
Password for user dbxdba:
Welcome to sqldbx, the interactive SQL terminal, for f2mdbX SQL 4.5.2.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with sqldbx commands
       \g or terminate with semicolon to execute query
       \q to quit

demodb=# select f2mdbx_version();
                                                                    f2mdbx_version

------------------------------------------------------------------------------------
-----------------------------------------------------
 f2mdbX SQL 4.5.2 (XDCAT_0363.0522) API Toolkit 2.10.4 on 4.14.268-205.500.amzn2.
x86_64 #1 SMP Wed Mar 2 18:38:38 UTC 2022 x86_64 Linux
(1 row, Query Total: 1)

demodb=# \d
No relations found.
demodb=#
```

sqldbx may also be used to interactively run metadata discovery and analysis commands by executing *f2md Toolkit API* calls (refer to *API Toolkit Reference*) in a SQL SELECT query. The example below discovers profile metadata on ORC files of New York 2009 taxi data in AWS S3 private bucket using a MANIFEST file to skip some of the files in the bucket. AWS S3 authorization key and the MANIFEST file are assumed

to be stored in home path of user *dbxdba*. The files may be transferred using *File Transfer* tool discussed in section 8.6 or using Linux command *scp* with your AWS cluster authorization key.

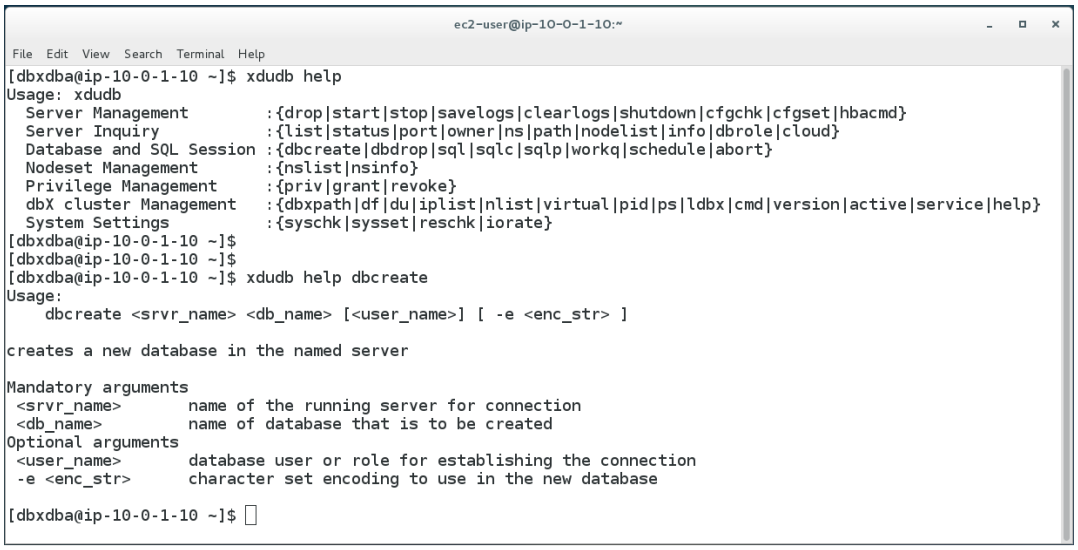


Help on commands of *sqldbx* may be obtained by typing *\?*. All SQL queries typed in at command prompt must be terminated by a semicolon and followed by *RETURN* for query execution. Queries in a SQL script file may be executed using *\i* command, and output may be redirected to a file or screen by *\o* command. *^C* at command prompt will abort running queries or clear syntax errors. You may terminate the interactive SQL session by typing *\q* at command prompt.

8.3 Server Tools: xdudb

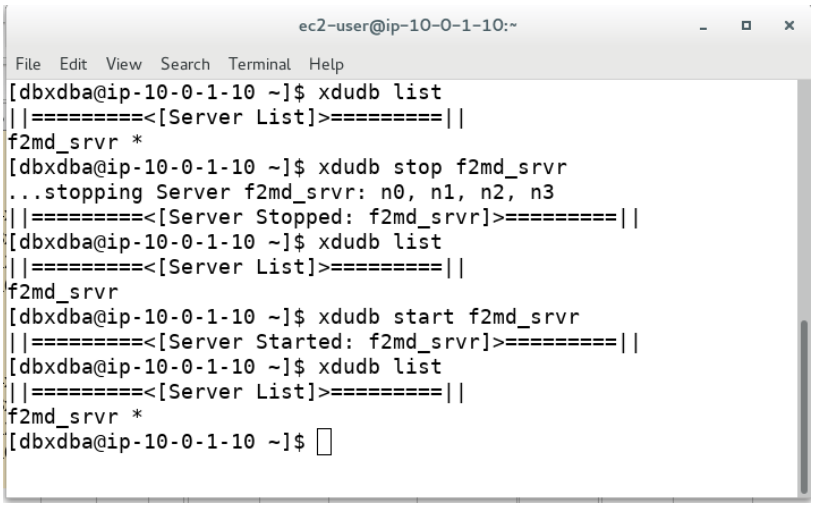
xdudb is a command line utility run at Linux shell prompt to administer, manage and monitor f2mdbX servers; some of the *xdudb* commands cannot be run when you are logged in as *ec2-user*, and you must switch to become one of the non-root users (e.g., *dbxdba*) that can execute all *xdudb* commands.

Command options supported by *xdudb* are shown below using its *help*; some of these commands have already been used in previous sections.



xdudb may also be used to start interactive SQL query tool *sqldbx* through its command option *sql* (as given in section 8.2), or used to run a SQL script in batch mode, either in foreground or background mode, using its command option *sqlc*.

After creation of a f2mdbX server (section 7) the server *f2md_srvr* is running on the cluster. You may list the running server, stop it and restart it as shown in the figure below using *xdudb* commands. When you list the servers on the cluster, *xdudb* displays a * against the name of the currently running server.



A currently running server cannot be started again. When stopping a running server, it is recommended that all active connections should be gracefully terminated else transactions may abort leading to rollback actions when the server is restarted.

8.4 Server Tools: xdAdm

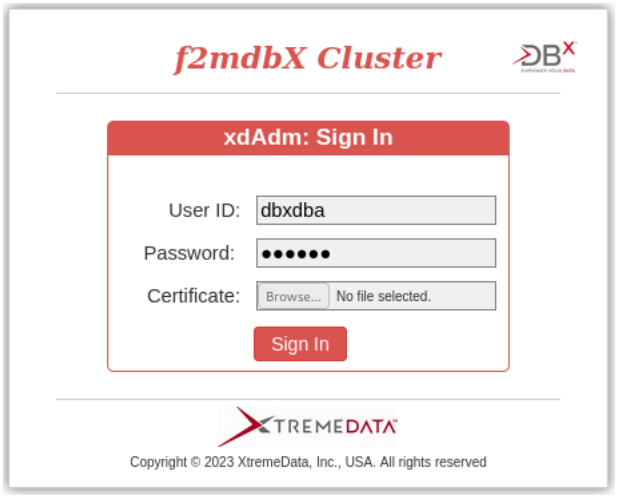
xdAdm is a browser enabled GUI tool to administer, manage and monitor f2mdbX servers remotely through GUI based commands.

The product *xdAdm* may be run through a browser using the URL displayed by *setupserver* during f2mdbX server creation (section 7); an example:

https://18.208.137.2:2400/xdadm

The above URL assumes that the public IP of your cluster is 18.208.137.2 and 2400 is the port for communication. An alternative method to using the above URL to run *xdAdm* is given in section 8.5. Refer to *xdAdm User Guide* for product documentation on *xdAdm*.

You must sign into *xdAdm* with Linux user *dbxdba* and its password, after starting it:



8.5 Cluster Management: xdc

xdc is a Linux command line command line utility to monitor and administer XtremeData products installed on the AWS cluster. All *xdc* commands may be run only at the shell prompt of AWS created user *ec2-user*.

xdc status

The *status* command of *xdc* to monitor the progress at the time of cluster creation and launch is described in section 7. You may use the *status* command later too.

xdc stop

On AWS cluster startup, f2mdbX services on the cluster are also automatically started, and may be stopped using *xdc* command, with the option *stop*, at shell prompt of *ec2-user*. The *stop* command does not stop the AWS cluster (section 9), but only stops f2mdbX services on the cluster including any servers that may be running on the cluster. The command may take a while to stop a multi-node cluster, and periodically updates the screen about its actions.

xdc start

If you manually stop f2mdbX services using *xdc stop*, you must also manually restart f2mdbX services later using *xdc start*, else none of the f2mdbX tools will run on the cluster.

8.6 f2md Tools: Remote Access

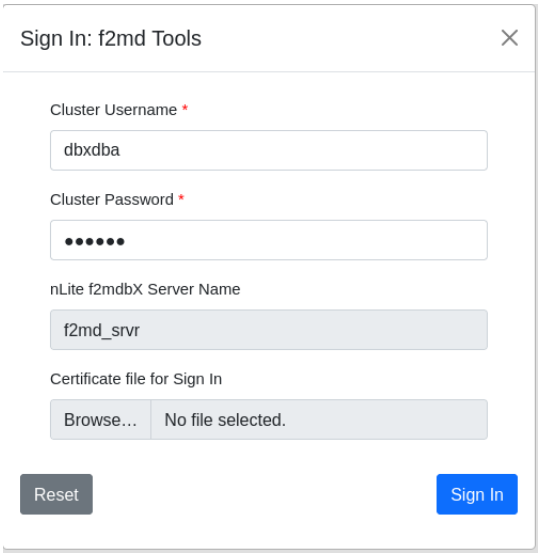
Two of the four tools of *nLite f2md Platform*, *sqldbx* and *xdudb*, are command line utilities that require a login window on the AWS cluster through ssh or Putty. Such a

login requires the AWS private key, and once you are logged in you may run these tools at the command line as shown in the figures of sections 8.2 and 8.3.

As an alternative remote access method, XtremeData supports a browser based GUI tool *uitools* that provides a terminal window after validation of credentials: Linux user name, password and client authentication certificate (if any). In order to run *uitools* use the URL displayed in section 7.

https://18.208.137.2:8444/uitools

A dialog box for logging in as a Linux user (*dbxdba*) of the cluster is displayed. If the server created in section 7 is configured for client authentication, you must also provide a certificate in addition to Linux user password.

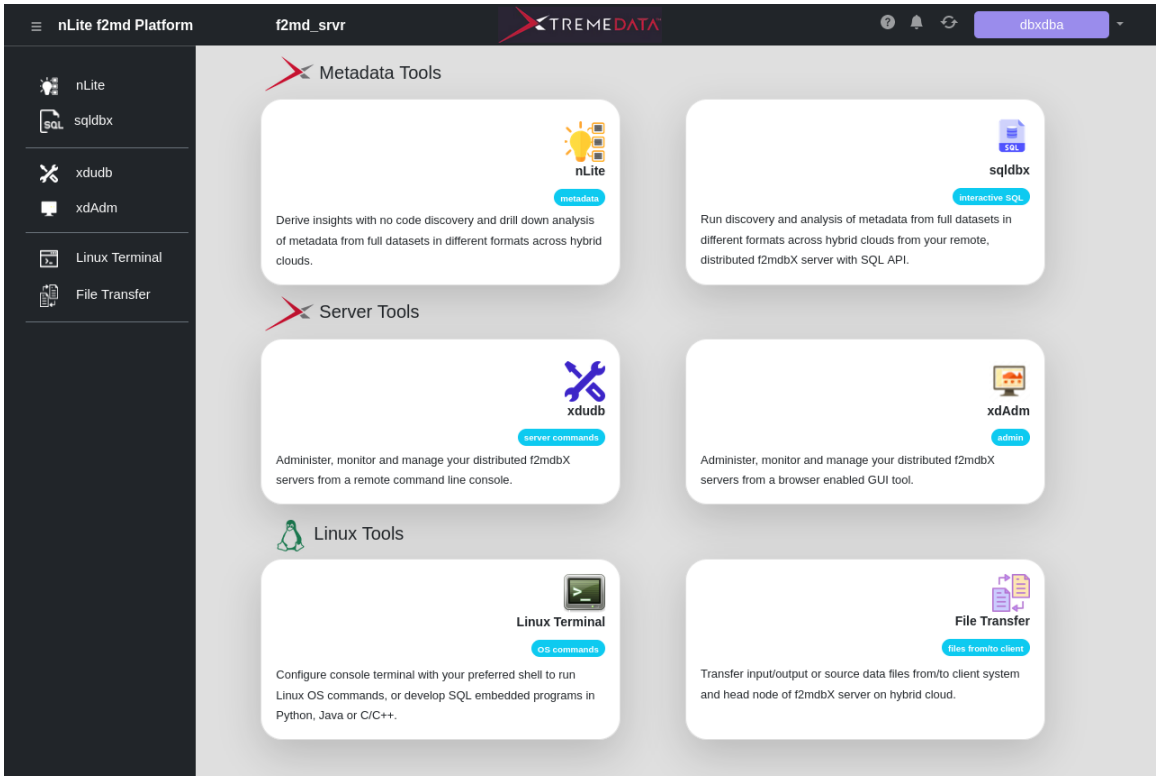


After successful login, *uitools* displays six icons as shown in the figure given below. Click on question mark icon (*?*) in title bar for *Help*. Exit the utility by clicking on purple button displaying Linux user name and choosing *Sign Out*. You cannot create multiple tab sessions for the same tool in the browser window.

You may run the GUI tools *nLite* and *xdAdm* by clicking on the first or fourth icon respectively, without having to specify the URLs listed in sections 7, 8.1 or 8.4.

Clicking on the second icon *sqldbx* displays a terminal window prompting you for login password of f2mdbX role. Using this icon, you may run only *sqldbx* (section 8.2) commands and cannot run any other command. The terminal window is closed when you exit *sqldbx* with *\q* and then type *y* on prompt.

Clicking on third icon *xdudb* displays a terminal window prompting you for Linux login password of current user. Using this icon, you may run only *xdudb* (section 8.3) commands and cannot run any other tool. The terminal window is closed when you exit *xdudb* with *exit*.



The interface displays two more icons, one of which is to create a Linux terminal to run Linux OS commands. Clicking on the fifth icon labeled *Linux Terminal* prompts you for Linux password of current user and provides a shell window for executing Linux commands in home path of the user.

The sixth icon labeled *File Transfer* facilitates transfer of files from, or to, your client system running the browser and folder *data* under home path of Linux login user (for user *dbxdba*, path is */home/dbxdba/data*). In order to transfer files to or from cluster, login user must have necessary privileges and also provide the login password.

9 AWS Cluster: Stop

The running AWS cluster may be stopped when compute resources of the cluster are not being used, as an idle server also consumes cloud resources and incurs cost.

As f2mdbX server stores persistent data on EBS, the storage is detached when a cluster is stopped and both your data and metadata are not lost. You may later restart the cluster and get access to your data or metadata on detached EBS volumes. You would incur a cost from AWS for detached EBS storage during cluster stoppage period.

Stopping of your AWS cluster must be done from the *EC2 Instances* page of your AWS login account and you must be logged in to your AWS account. Prior to stopping the AWS cluster, it is a good practice to ensure that all users are logged out. It is not essential to stop an idle f2mdbX server.

In the *EC2 Instance* page, the list of instances of your account are displayed with instance name, state, type, etc. Select the check box against a stopped AWS instance that you wish to stop; for a multi-node cluster select all nodes of the cluster. In order to stop an instance, the instance state must be displayed as *Running*.

You may stop the selected running instance by clicking on *Stop Instance* command from the pull-down menu of *Instance state* button, or from right click menu on instance details. A confirmation dialog listing all instances selected for stopping is displayed. Click on button *Stop* of the dialog box to stop the selected instances.

AWS updates the status of instances selected for stopping, and you may view the updates by clicking on *Refresh* icon of the page. Stopping all instances of a multi-node cluster may take a while; eventually all instances would be updated to status *Stopped* on the *EC2 Instances* page.

If you do not wish to reuse the data in EBS volumes of your cluster, you may terminate the instance with *Terminate Instance* command instead of *Stop Instance.* As *Terminate Instance* would destroy all data on EBS, you do not incur any cost for EBS volumes after termination. Later, if you restart the cluster the data stored in cluster EBS is lost. However, the cloud formation stack itself is not deleted by *Terminate Instance*.

10 AWS Cluster: Restart

An AWS cluster that has been stopped may be restarted at a later point of time. No resource cost (except for EBS volumes) is incurred on a stopped AWS cluster. f2mdbX servers use detachable EBS volumes attached to cluster instances for persistent data, which is not lost and remains on the storage during stoppage period of the cluster.

When you restart the cluster, EBS volumes of same size assigned at the time of cluster creation but with your last saved persistent data, are attached to the newer instance nodes. You may therefore continue to use f2mdbX servers that were used on the cluster before it was stopped.

Navigate to *EC2 Instances* page on AWS console after signing in to your AWS account. The list of instances of your account are displayed with instance name, state, type, etc. Select the check box against a stopped AWS instance that you wish to start; for a multi-node cluster select all nodes of the cluster. In order to start or restart an instance, the instance state must be displayed as *Stopped*.

You may start or restart the selected stopped instance by clicking on *Start Instance* command from the pull-down menu of *Instance state* button or from right click menu on instance details. AWS initiates start of the selected cluster instances.

AWS updates the status of instances selected for starting, and you may view the updates by clicking on *Refresh* icon on the page. Starting all instances of a multi-node cluster may take a while; eventually all instances would be updated to status *Running* on the *EC2* page.

On a restart of the f2mdbX cluster, though you would get back access to persistent data of f2mdbX databases, the head node public IP of the cluster will change. Note the public IP of head node of the new cluster for logging in to the head node, or remotely accessing dbX services with GUI or other tools; inform new IP to all your cluster users.

After restarting a stopped cluster, you do not have to set up (section 7) the f2mdbX server again, but may have to start f2mdbX server using the following command, as Linux user *dbxdba*, after ensuring that the cluster is successfully restarted (section 6).

xdudb start f2md_srvr

11 AWS Cluster Stack: Delete

Even after termination, the stack that you generated using steps of section 4 to create a cluster remains in your AWS account. You may discard and delete the cluster creation stack permanently using *Delete* command on your stack identified by its name.

Navigate to *CloudFormation* services page on AWS console after signing in to your AWS account. The list of cloud formation stacks of your account is displayed. Select the stack to be deleted, and click on button labeled *Delete* above the stack list.

A dialog box to confirm permanent deletion of selected stack is displayed. Click on button labeled *Delete* of dialog box. The selected cluster creation stack is deleted from your AWS account.

Support

Contact XtremeData at support@xtremedata.com for any further assistance that you may need.