dbX MNT AWS Setup Guide

Rev 4.0
June 2018

XtremeData, Inc.
999 Plaza Dr., Ste. 570
Schaumburg, IL 60173

www.xtremedata.com

1. Overview 3
1.1 Architectural Diagram 4
1.2 Architectural Elements 5
2. Single-node Setup 6
2.1 Configure & Launch dbX instance 6
2.2 Initialize 6
2.3 Start cluster & dbX 6
2.4 Stopping the cluster 7
2.5 Re-starting the cluster 7
3. Multi-node Setup 8
3.1 Enable the Software 8
3.2 Launch dbX template 8
3.3 Login the first time 8
3.4 Stopping the cluster 9
3.5 Re-starting the cluster 9
4. Upgrading the software 10
5. Efficient Use of Resources 11
5.1 Cluster Size 11
5.2 EC2 Instance Selection 11
5.3 S3 Object Store Usage 11
5.4 Cluster Cost Estimation 11
5.5 Temporary Cluster Shutdown to Save Cost 12
5.6 Instance Resizing 12
6. Security and Access Control 15
6.1 AWS Security Group 15
6.2 Host Based Client Authentication 15
7. Binary Backup and Restore/Clone 16
7.1 Procedure for backing up a database server 16
7.2 Procedure for restoring a database server 18
7.3 Procedure for cloning a database server 20
8. System Monitoring 21
8.1 Database Connections 21
8.2 Disk Space Usage 21
8.3 Query History 22
8.4 Other Monitoring Utilities 22
9. Resiliency 22
10. Maintenance and Support 22

1. Overview

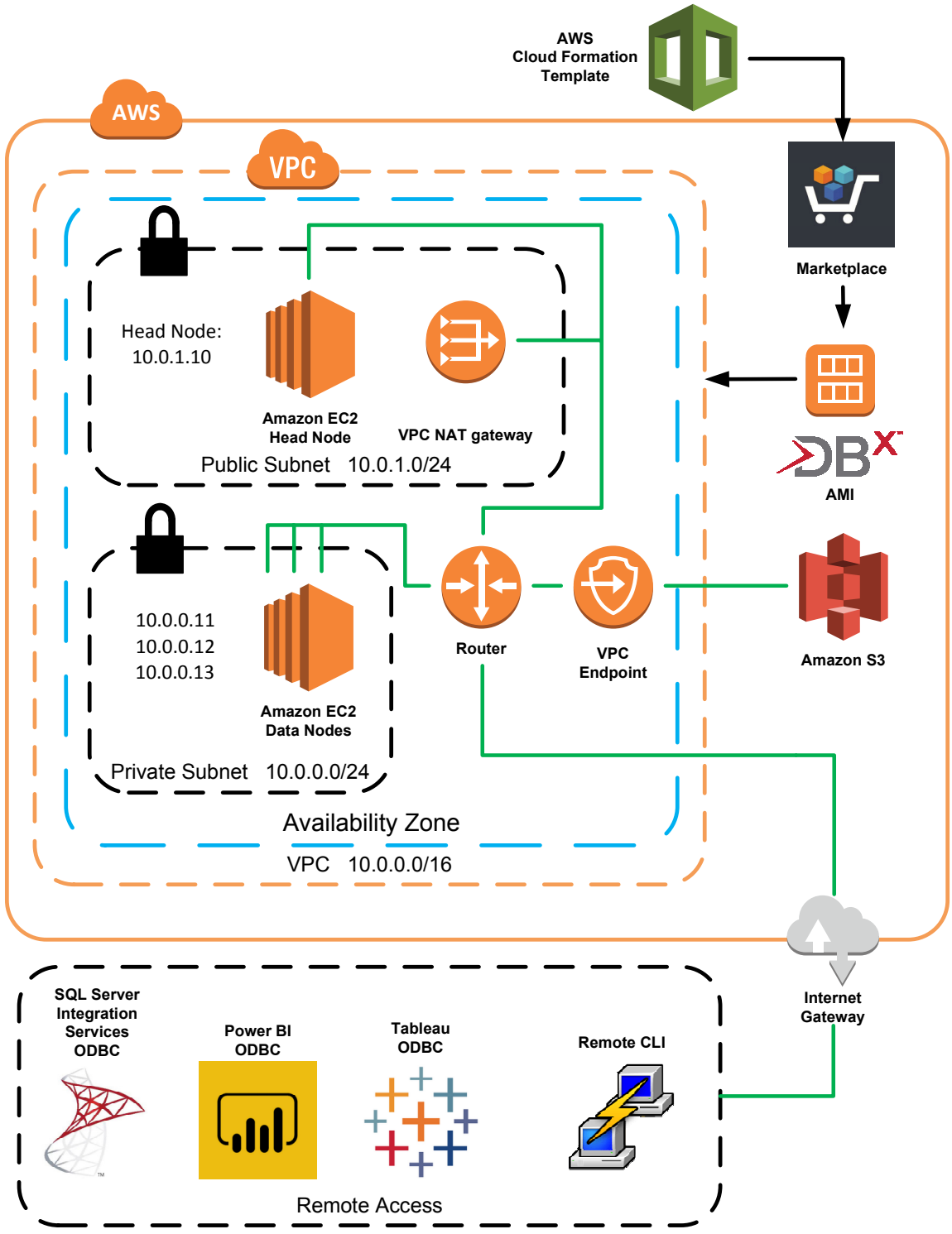
This document describes the process to launch an XtremeData dbX system on Amazon Web Services. Stopping and re-starting of the cluster is also covered.

dbX may be deployed as a single node system with a combined virtual head and data node via the AWS marketplace console. A multi-node dbX system comprises one “head” node and multiple “data” node instances which are launched through an appropriate Cloud Formation template. Both configurations are referred to as “clusters”. Launching the cluster from a template is the preferred method of deployment. Launching a single node from the marketplace is discussed just for completeness.

It is assumed that the user has some familiarity with AWS and knows how to set up an account and establish keys for communication with the running instances.

Please refer to dbX documentation for detailed product descriptions, user guides and SQL reference manuals.

1.1 Architectural Diagram



1.2 Architectural Elements

Elements outside the AWS Cloud environment

AWS Cloud Formation Template - This template is provided by XtremeData and works through AWS Marketplace to access the latest DBx Amazon Machine Image to create the cluster in the cloud. This template can be customized for each customer's needs.

Remote CLI - PuTTY Command Line Tool - This terminal uses an ssh public/private key pair to gain remote access to the cluster via XtremeData's version of PSQL command line tool.

Third Party ODBC Access - Third party tools like SQL Server Integration Services, Power BI, and Tableau can access a DBx cluster via a standard ODBC or JDBC interface.

AWS Cloud - The solid orange line encompasses all features residing in the AWS Cloud environment.

Marketplace - The dbX software image can be accessed through AWS Marketplace where supported EC2 instances and costs are defined.

dbX AMI - The dbX Amazon Machine Image resides in the AWS Marketplace. Registered customers are made aware of updates when they occur.

Amazon S3 – No S3 buckets or objects are created, but for convenient data access, data files can be stored there and should be in the same AWS region as the cluster.

Internet Gateway - Access to the external internet.

VPC - Virtual Private Cloud - The orange dashed line encompasses all features created by the Cloud Formation Template and is considered to be the "database cluster".

Availability Zone - The Blue dashed line encompasses the cluster. All elements of a cluster should be within a single availability zone.

VPC Endpoint - This enables access to the S3 object store.

Router - Contains the routing tables.

Public Subnet - Enables external network access to the cluster.

Amazon EC2 Head Node - By default the cluster head node is the only node accessible from the external network.

VPC NAT Gateway - Network gateway to the outside world.

Private Subnet – All data nodes in the cluster can access each other using the private subnet.

2. Single-node Setup

2.1 Configure & Launch dbX instance

Marketplace Image
Search the "AWS Marketplace" for "dbx".

Select the image titled "XtremeData dbx MNT (HVM)".

Security Group
The security group defaults should be used to allow ssh access over port 22 and GUI administrative access via port 2400.

Block storage (EBS)
The instance can run with only ephemeral drives. However, if the instance is ever stopped, all state and data will be lost. To avoid this, EBS drives should be used. All storage devices that are made available before initialization will be used for dbX. If additional storage is needed for other purposes, it should be added after the initialization process is complete. If EBS is used, it is recommended to have at least 4 devices with a Volume Size that is a minimum of 100GB each (they should all be the same size). The recommended class of storage is General Purpose SSD.

2.2 Initialize

WARNING: Initialization is done once after the instance is created. This is a destructive operation that will wipe-out all pre-existing DB servers and data.

Any EBS storage devices that are to be used by dbx should be attached before proceeding. If there are EBS devices that should not be formatted and used by dbx, they should be detached before proceeding. All local ephemeral drives will be formatted and used by dbx.

Using your preferred terminal window (e.g. putty), login to the head node as "ec2-user".

Initialize the cluster:

```
[ec2-user] $ cluster_init -p -i 0
```

Note that "`-p`" is an option to make the head node also function as a data node. The "`-i 0`" option indicates that there are no additional data nodes (correct for a single-node configuration).

This will initialize all storage devices that are attached and make them available for dbX.

- Final configuration tasks will be performed.
- This is a DESTRUCTIVE process – no previous data or configuration will be preserved!

When finished, the command will report: "cluster_init done"

2.3 Start cluster & dbX

Start the cluster:

```
[ec2-user] $ cluster_start
```

When finished, the command will report: "cluster_start done"

Now your instance is initialized and running. Next step is to start the dbX database engine:

```
[ec2-user] $ dbx_start
```

You will see some messages displayed during the start-up of dbX. Take note of the message “starting dbx nodes, may take several minutes”. This is normal.

When finished, the command will report “dbx startup done”

Upon starting dbX for the first time, your “ec2-user” keys will be copied to user “dbxdba”. User “dbxdba” is preconfigured to allow the creation and management of all DB servers. This user will also be used to log into the administration tool.

Important note: You need to setup a system password for user “dbxdba” in order to use the administration tool. Set the password:

```
[ec2-user] $ sudo passwd dbxdba
```

You are now up and running! You can now login to the head as user “dbxdba” to create servers and databases or login to the administration tool at: https://<IP address>:2400/xdadm

Please refer to dbX user documentation volumes I – IV for information on creating and managing servers and databases. Note that any references in the documentation to tasks managed by user “xdAdm” are performed by user “dbxdba”.

2.4 Stopping the cluster

To stop the cluster, login to the head node as “ec2-user” and stop it:

```
[ec2-user] $ dbx_stop
```

All the running database servers will be stopped and main daemon “xdu” will be stopped. Wait until the stop process is complete.

Then in the AWS console select the instance and stop it.
Note: if your data is on ephemeral storage (no EBS drives were configured), it will be lost!

2.5 Re-starting the cluster

To restart the cluster, select the instance and start it. Note that the public IP address will probably change unless you have attached an Elastic IP.

Once the instance has been re-started, you need to re-start the cluster and dbX. NOTE: since this is a re-start, and your database might already contain data, you should NOT initialize storage, otherwise all data will be lost!

Using your preferred terminal window, login to the head node as “ec2-user”

Start cluster:

```
[ec2-user] $ cluster_start
```

Start the dbX database engine:

```
[ec2-user] $ dbx_start
```

3. Multi-node Setup

Multi-node setup is done automatically with the proper template. Once deployed, the only additional step is to login to the head node and assign an administrative password.

Sample cloud formation templates:

For deployment into a new VPC with NAT and S3 endpoint:
https://s3.amazonaws.com/xtremedata-dbx-public/Marketplace-dbx_cluster-New_VPC-EBS_GP_SSD.template

For deployment into an existing VPC:
https://s3.amazonaws.com/xtremedata-dbx-public/Marketplace-dbx_cluster-Exist_VPC-EBS_GP_SSD.template

Note that the above templates implement 1 head node and 1 data node. To create a new template that deploys more data nodes, use the following Python-based scaling tool on one of the templates:
https://s3.amazonaws.com/xtremedata-dbx-public/dbx_template_scale_aws

```
$ ./dbx_template_scale_aws
usage: dbx_template_scale_aws [-h] [--ifile IFILE] [--ofile OFILE]
                              [--tab [TAB]] [--debug [DEBUG]]
                              nodes

$ ./dbx_template_scale_aws --ifile= Marketplace-dbx_cluster-New_VPC-
EBS_GP_SSD.template --ofile=my4nodes.template 4
```

Note that the number of nodes specified is the number of data nodes in addition to the head node.

3.1 Enable the Software

Before using the template, you must enable the software in Marketplace. Select the image titled "XtremeData dbx MNT (HVM)" and proceed like you plan to deploy it from Marketplace. You will be prompted to agree to the usage terms. After you have accepted the agreement, you can abort the deployment and then use the template. This process will only need to be done once for your account. If it is not done, you will get a deployment failure.

3.2 Launch dbX template

The multi-node template will configure the number of nodes and storage, run initialization and automatically start dbX. Note that this scenario utilizes a management framework for this automation. The framework will automatically shutdown dbx if there is a problem with the cluster and automatically bring it back up once the cluster is in a working state. This automated management supersedes any manual management of the service daemons associated with dbx, including xdadm and xdu.

3.3 Login the first time

Using your preferred terminal window (e.g. putty), login to the head node as "ec2-user".

Check the status of the cluster:

```
[ec2-user] $ xdc status
```

It should report: started. When the cluster is first launched, there may be several state transitions that may occur, including New, Stopped, Configured, Transitional or even Failed. This is normal and the cluster should transition to the "Started" state in a few minutes. If it does not reach that state, it may be due to a resourcing issue in the cloud infrastructure and may take some time to resolve. If it never reaches the "Started" state, the best course would be to delete the deployment and deploy a new stack.

Important note: You need to setup a system password for user “dbxdba” in order to use the administration tool. Set the password:

```
[ec2-user] $ sudo passwd dbxdba
```

Upon starting dbX for the first time, your “ec2-user” keys will be copied to user “dbxdba”. User “dbxdba” is preconfigured to allow the creation and management of all DB servers. This user will also be used to log into the administration tool.

You are now up and running! You can now login to the head as user “dbxdba” to create servers and databases or login to the administration tool at: https://<IP address>:2400/xdadm

Please refer to dbX user documentation volumes I – IV for information on creating and managing servers and databases. Note that any references in the documentation to tasks managed by user “xdAdm” are performed by user “dbxdba” in this environment.

3.4 Stopping the cluster

To stop the cluster, login to the head node as “ec2-user” stop it:

```
[ec2-user] $ xdc stop
```

All the running database servers will be stopped and main daemon “xdu” will be stopped. Wait until the stop process is complete.

Then in the AWS console select the instances belonging to the cluster and stop them.
Note: if your data is on ephemeral storage (no EBS drives were configured in the template), it will be lost!

3.5 Re-starting the cluster

To restart the cluster, select the instances and start them.

Once all the instances have been re-started, the software will restart automatically.

Using your preferred terminal window, login to the head node as “ec2-user” to check the status of the cluster:

```
[ec2-user] $ xdc status
```

4. Upgrading the software

It is always recommended that when launching a new cluster and periodically thereafter, that the software be upgraded to the latest version.

For the manually configured single node deployment, stop any running DB servers and run:

```
[ec2-user] $ sudo yum update
```

For the multi-node deployment, upgrade the entire cluster from the head node as follows:

```
[ec2-user] $ xdc stop
```

To upgrade all software packages, including the kernel:

```
[ec2-user] $ xdc upgrade --all=yes --verify=no
```

To upgrade only the DB software:

```
[ec2-user] $ xdc upgrade --verify=no
```

Once the upgrade completes, run:

```
[ec2-user] $ xdc verify --auto=yes
```

On some systems there may be dependency errors that need to be manually addressed:

```
Required packages dependencies not met for:selinux-policy 3.7.19-292

System state or settings are invalid. Xdc exit code:112
```

In this case, manually upgrade the dependency:

```
[ec2-user] $ xdc upgrade selinux-policy --verify=no

[ec2-user] $ xdc verify --auto=yes
```

If the kernel was upgraded, all the nodes need to be rebooted and they will automatically start the software. Otherwise run:

```
[ec2-user] $ xdc start
```

5. Efficient Use of Resources

5.1 Cluster Size

By default the provided templates attach 2TB of EBS drives to each physical node. XtremeData recommends that you scale your cluster size to only use 50% of the available disk space at the beginning to allow for expansion. We also recommend that you create compressed column store tables, which will have an ASCII input to disk size compression ratio of about 2:1 .

data nodes = (ASCII Input Data Size in TB) / 2 TB

So for a 15 TB ASCII database we would recommend at least an 8 node cluster.

5.2 EC2 Instance Selection

dbX software performance is dependent on the number of CPU cores, disk bandwidth, the amount of memory, and network bandwidth. The AWS Marketplace and Cloud Formation Templates will limit your selection to only those instances which have been qualified by XtremeData, but that still leaves a large range of choices. The good news is that the instance type can be changed even after the cluster is created and the database has been loaded, so the initial instance choice is not permanent.

We recommend using the **i3** series:
i3.2xlarge for cost sensitive applications
i3.8xlarge for high performance applications

There are **i3.4xlarge** and **i3.16xlarge** instances as well.

5.3 S3 Object Store Usage

Source and backup data should be stored in S3 object store in the **same region as the cluster**. Compression of source files should be used to reduce long term storage costs and speed up the data transfer between object store and cluster.

5.4 Cluster Cost Estimation

Cluster cost has two components, the EC2 instances (which include the software cost) and the persistent EBS drives.

EC2 Instance	Cost per hour
i3.2xlarge	$1.264
i3.4xlarge	$2.528
i3.8xlarge	$5.056
i3.16xlarge	$10.112

EBS SSD (gp2) drives cost $0.10 per GB-month, so:
2 TB = 2,000 GB * $0.10 per GB-month = $200 / month
or
$200 / month * (month / (30 * 24 hours)) = $0.278 / hour-node

Elastic Block Storage Drives per Node	Cost per hour per node
4 x 500 GB EBS SSD (gp2) drives	$0.278 / hour-node

Example: 8 node cluster of i3.8xlarge instances
Cost while running:
8 * ($5.056 + $0.278) = $42.67 / hour

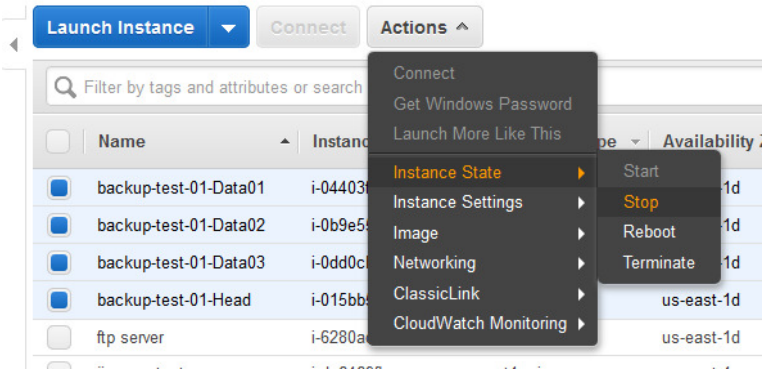
Cost while EC2 instances stopped:
8 * $0.278 = $2.22 / hour

5.5 Temporary Cluster Shutdown to Save Cost

To save money while not in use the cluster can be shut down without losing the existing database. This is possible because the database files reside on the non-volatile EBS drives which persist through a cluster shutdown. As seen in the previous section, cost savings up to 95% can be realized during shutdown.

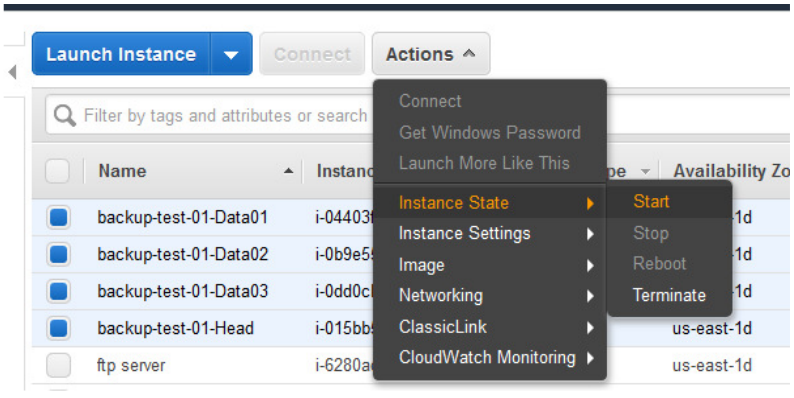
To put the cluster in shutdown mode the administrator should first make sure that all queries have finished. It is not necessary to shutdown the database server, but shutting down the database server is recommended and is a sure way to know that no queries are running when the cluster is stopped.

Log into the AWS GUI, and navigate to the **EC2 Instances** screen.
Select all nodes in the cluster.
Select **Actions → Instance State → Stop**



Wait a few minutes to make sure that the shutdown process has begun. If not, then refresh the screen and repeat the process.

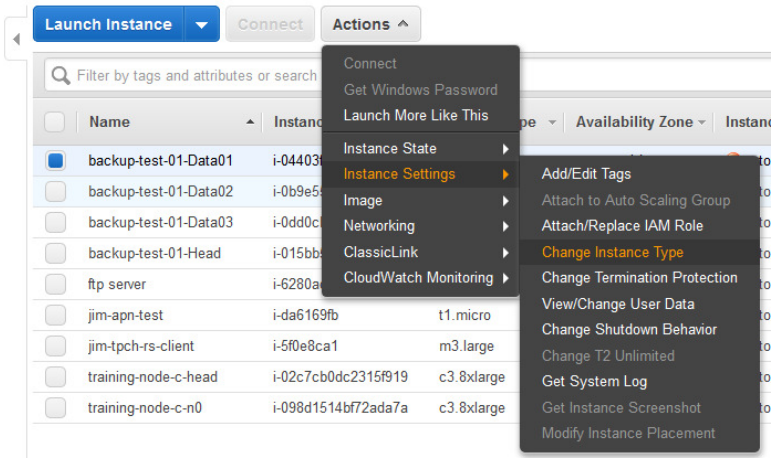
To restart select all nodes.
Select **Actions → Instance State → Start**
Wait until the dbX software has had time to boot (check with **`[ec2-user]`** `xdc status`). If the DB server was manually shutdown, it will need to be restarted (`[dbxdba] xdudb start <db_server>`) to begin queries.



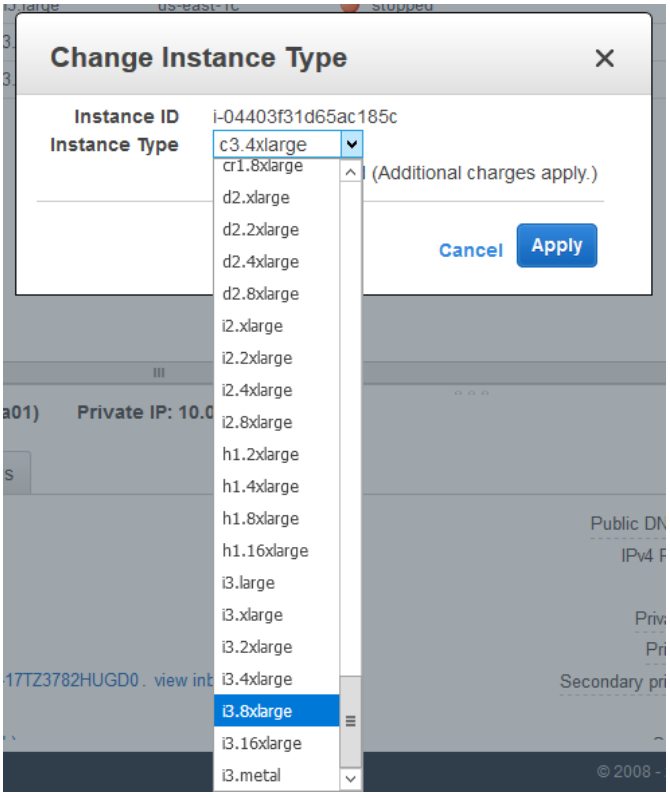
5.6 Instance Resizing

To change the EC2 instance type you should first stop all cluster nodes as described in the previous section.

Then select each instance individually and …
Select **Actions → Instance Settings → Change Instance Type**



In the pop-up window select the new instance type



Select **Apply**
Repeat this procedure for nodes in the cluster.
Start all nodes.
Wait for dbX software to boot.

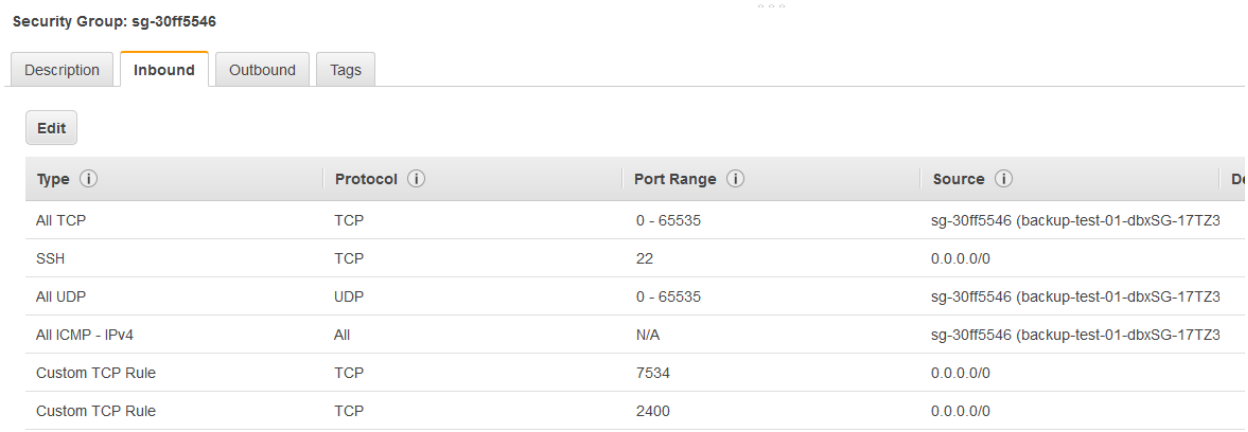
NOTE: There are database cache settings which make use of the available memory. It the amount of memory in the new instance is less than the memory of the original instance, you may have to set new cache settings before restarting the database server.

6. Security and Access Control

There are two layers of security which control access to the dbX Cluster. The outer layer is controlled by the AWS Security Group, which controls external access to the cluster. The inner layer is the dbX software Client Authentication which controls access to the database.

6.1 AWS Security Group

The cluster is built within an AWS Security Group which must have specific ports open to allow communication to the dbX server via the head node. Below is an example of a Security Group.



Type	Protocol	Port Range	Source	D
All TCP	TCP	0 - 65535	sg-30ff5546 (backup-test-01-dbxSG-17TZ3	
SSH	TCP	22	0.0.0.0/0	
All UDP	UDP	0 - 65535	sg-30ff5546 (backup-test-01-dbxSG-17TZ3	
All ICMP - IPv4	All	N/A	sg-30ff5546 (backup-test-01-dbxSG-17TZ3	
Custom TCP Rule	TCP	7534	0.0.0.0/0	
Custom TCP Rule	TCP	2400	0.0.0.0/0	

The line with type SSH and **port 22** allows terminal access via SSH key pairs. A public key is specified when the cluster is built allowing any user with the private key to gain command line access to the cluster over this port.

Port 2400 must be opened to allow browser based GUI access to the administration dbX software. When this port is used, a specific Linux user, usually dbxdba, will be granted password based access.

Port 7534 is the database port for this example. To enable third party ODBC access, this port must be opened. ODBC access to the database will only be granted if the ODBC credentials pass the host based client authentication described in the next section.

In this example these ports are open to all source hosts, but security can be restricted to specific hosts.

6.2 Host Based Client Authentication

dbX employs a host based client authentication methodology which grants specific hosts access to databases based on the rules shown below. This is a screen shot of the browser based xdadm tool.

Client Authentication: Server demo_finmkt_server (Host Records)

Select	Type	Database	Role	IP Address	Method	Edit State
○	host	all	all	127.0.0.1/32	password	--
○	host	all	all	::1/128	password	--
◉	host	all	all	40.78.102.242/32	**md5**	Altered
○	host	all	all	96.84.117.220/32	**trust**	Altered
○	host	all	**"dbxdba"**	40.78.16.19/32	password	Altered
○	host	**"demo_finmkt_db"**	all	54.208.194.103/32	password	Altered

Append Insert Alter Drop Save Cancel

127.0.0.1 is granted access to all databases as all roles using password authentication.
40.78.102.242 is granted access to all databases as all roles using md5 password authentication.
96.84.117.220 is granted access without a password.
40.78.16.19 can only login as the dbxdba role.
54.205.194.103 can only access the demo_finmkt_db database.

The last column Edit State shows that there are pending changes that will not take effect until the database server is stopped and restarted.

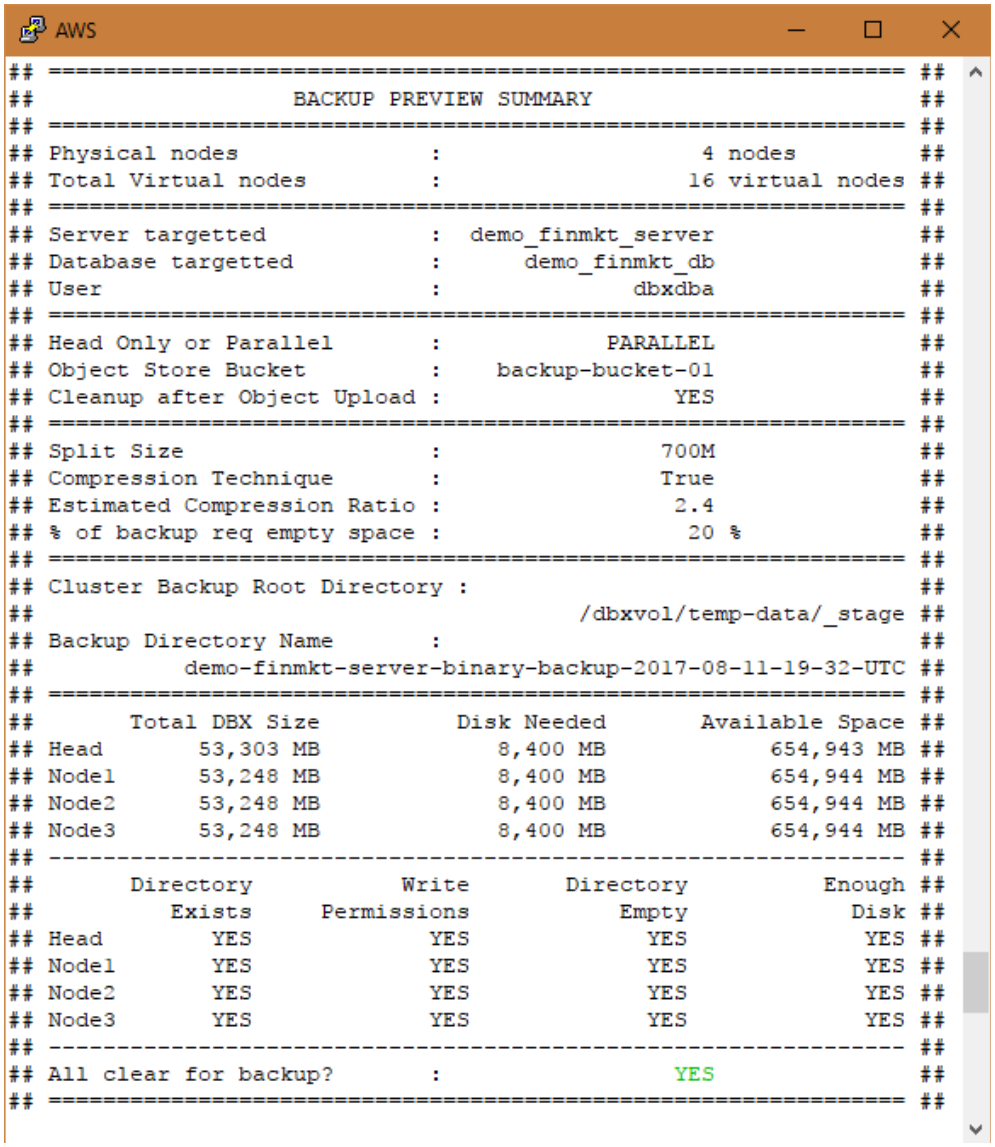
7. Binary Backup and Restore/Clone

XtremeData provides a backup utility which allows you to quickly back up large database binaries to object store by using all cluster nodes in parallel. This utility tars, compresses, splits, and uploads the database server binaries to object store using a very small disk footprint on each node. To the user, it will appear to be streaming the binaries to object store.

7.1 Procedure for backing up a database server

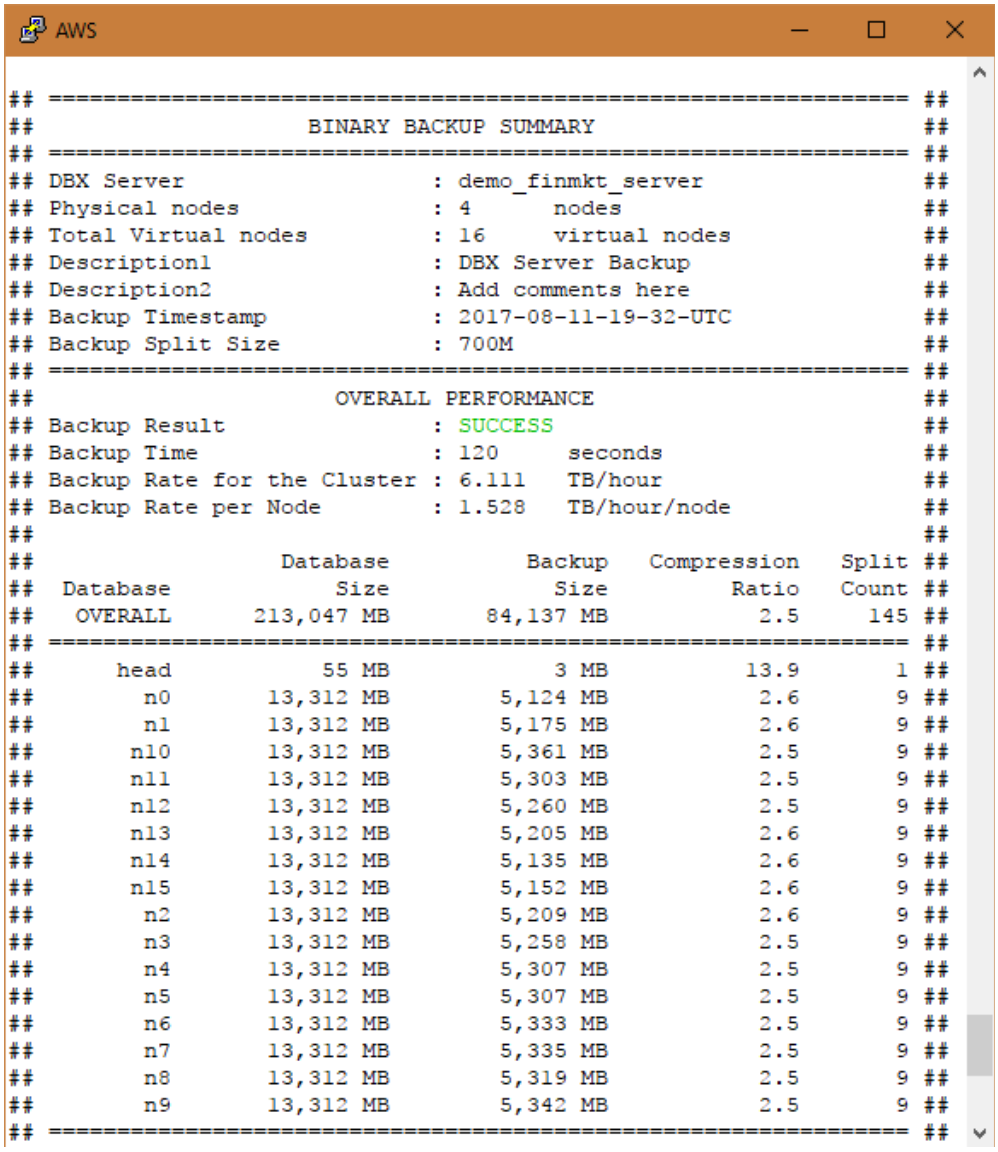
- Get terminal access to the head node using a port 22 SSH PuTTY window.
- Login as the system administrator dbxdba
- Stop the database server
  `[dbxdba] $ xdudb stop <database server>`
- Generate a backup configuration file template
  `[dbxdba] $ dbx-backup --gen`
- Modify the **backup-sample.conf** file to include all parameters relevant to this backup. Save it as **backup.conf** for this example. Refer to the user's guide for details.
- Execute the backup
  `[dbxdba] $ dbx-backup --config backup.conf`

You will see a screen like the one below



The backup utility checks the entire cluster to make sure that all nodes are prepared to do the backup before the backup begins. If any check fails, it will appear in red on this screen and the backup will exit out. If all checks pass the backup begins immediately.

After the backup is complete the following summary will appear.



If everything backed up correctly, then a green SUCCESS will appear. If any splits failed to upload, then a red FAIL will be reported and the backup will need to be retried. In this example the backup rate was just over 1.5 TB/hour/node.

Backups can be run in the background using the **`--json`** option.

7.2 Procedure for restoring a database server

- Get terminal access to the head node using a port 22 SSH PuTTY window.
- Login as the system administrator dbxdba
- If you are restoring a database server to a cluster with an existing server with the same name then you will have to drop that server before restoring, or chose a new server name and port number during the restore process.
  `[dbxdba] $ xdudb drop <database_server>`
- Generate a restore configuration file by using the backup configuration file as a template.
  `[dbxdba] $ dbx-restore --gen --template backup.conf`

- Edit the resulting **restore-sample.conf** if necessary and save it as **restore.conf** for this example.
- Execute the restore.
  [dbxdba] $ dbx-restore --config restore.conf
  All available backups for the specified server will be listed.

```
$ dbx-restore --config restore.conf
## ================================================================ ##
##                      CLUSTER SUMMARY                             ##
## ================================================================ ##
## Physical nodes              :            4 nodes                  ##
## ---------------------------------------------------------------- ##
## Node Name                   : Available Space                     ##
## head                        :    1,922,236 MB                     ##
## node00                      :    1,919,532 MB                     ##
## node01                      :    1,918,553 MB                     ##
## node02                      :    1,921,149 MB                     ##
## ================================================================ ##
##                        BACKUPS FOUND                             ##
## ================================================================ ##
## Select Server Name                      Time Stamp                ##
##    (0) demo_finmkt_server               2017-08-01-22-46-UTC      ##
##        Object store bucket :                      backup-bucket-01/ ##
##           demo-finmkt-server-binary-backup-2017-08-01-22-46-UTC ##
##        Description : DBX Server Backup                           ##
##                    : Add comments here                           ##
##        Physical    Virtual         Total                         ##
##           Nodes      Nodes          Size       Nodeset     Port ##
##               4         16       213,047 MB      NSVx4     7533 ##
## ---------------------------------------------------------------- ##
## Select Server Name                      Time Stamp                ##
##    (1) demo_finmkt_server               2017-08-02-21-02-UTC      ##
##        Object store bucket :                      backup-bucket-01/ ##
##           demo-finmkt-server-binary-backup-2017-08-02-21-02-UTC ##
##        Description : DBX Server Backup                           ##
##                    : Add comments here                           ##
##        Physical    Virtual         Total                         ##
##           Nodes      Nodes          Size       Nodeset     Port ##
##               4         16        51,043 MB      NSVx4     7533 ##
## ---------------------------------------------------------------- ##
## ================================================================ ##
Select a backup to restore [0:1],[2] to exit : 1
Backup 1 selected for restore.
Restore = "r", Delete backup from object store = "d", anyone other
character exits [r:d] : r  ← Select "r" for restore
```

- Select one of the backups, and then select "r" for restore as shown above.
  The backup will begin.

- When the backup completes the following summary will appear.

```
## ============================================================== ##
##                    BINARY RESTORE SUMMARY                      ##
## ============================================================== ##
## Restore Result              : SUCCESS                          ##
## ============================================================== ##
## DBX Server                  : demo_finmkt_server_a             ##
## DBX NodeSet                 : NSVx4                            ##
## DBX Port                    : 7536                             ##
## Physical nodes              : 4      nodes                     ##
## Total Virtual nodes         : 16     virtual nodes             ##
## Description1                : DBX Server Backup               ##
## Description2                : Add comments here               ##
## ============================================================== ##
##                       OVERALL PERFORMANCE                      ##
## Restore Time                :          165 seconds             ##
## Restore Database Size       :      213,047 MB                  ##
## Restore Rate per Cluster    :        4.431 TB/hour             ##
## Restore Rate per Node       :        1.108 TB/hour/node        ##
## ============================================================== ##
## Object Store Bucket         : backup-bucket-01                 ##
## Object Store Directory      :                                  ##
##           demo-finmkt-server-binary-backup-2017-08-01-22-46-UTC ##
## ============================================================== ##
##                     Total   Downloaded       Failed     Download ##
##                    Splits       Splits       Splits      Retries ##
##   OVERALL             145          145            0            0 ##
## ============================================================== ##
##       head                1            1            0            0 ##
##         n0                9            9            0            0 ##
##         n1                9            9            0            0 ##
##        n10                9            9            0            0 ##
##        n11                9            9            0            0 ##
##        n12                9            9            0            0 ##
##        n13                9            9            0            0 ##
##        n14                9            9            0            0 ##
##        n15                9            9            0            0 ##
##         n2                9            9            0            0 ##
##         n3                9            9            0            0 ##
##         n4                9            9            0            0 ##
##         n5                9            9            0            0 ##
##         n6                9            9            0            0 ##
##         n7                9            9            0            0 ##
##         n8                9            9            0            0 ##
##         n9                9            9            0            0 ##
## ============================================================== ##
```

This restore was a success. There were no download retries, and the restore occurred at the rate of just over 1 TB per hour per node.

7.3 Procedure for cloning a database server

- Build an identical cluster.
- Restore a backup to that cluster.
- Done.

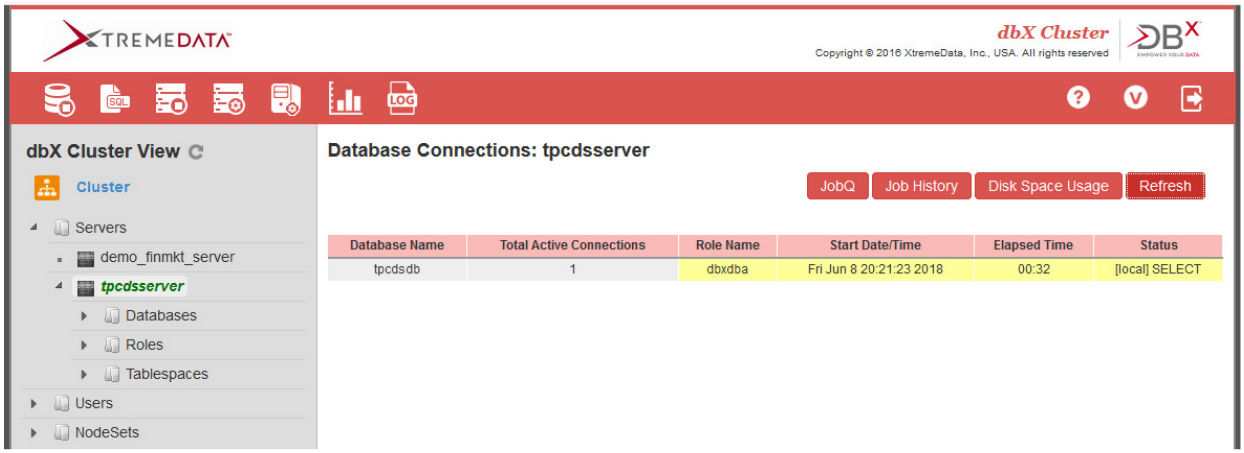
Refer to the dbx_backup_ug.pdf document for details on the backup and restore feature.

8. System Monitoring

The dbX xdadm browser-based interface enables the user to monitor system performance in real time.

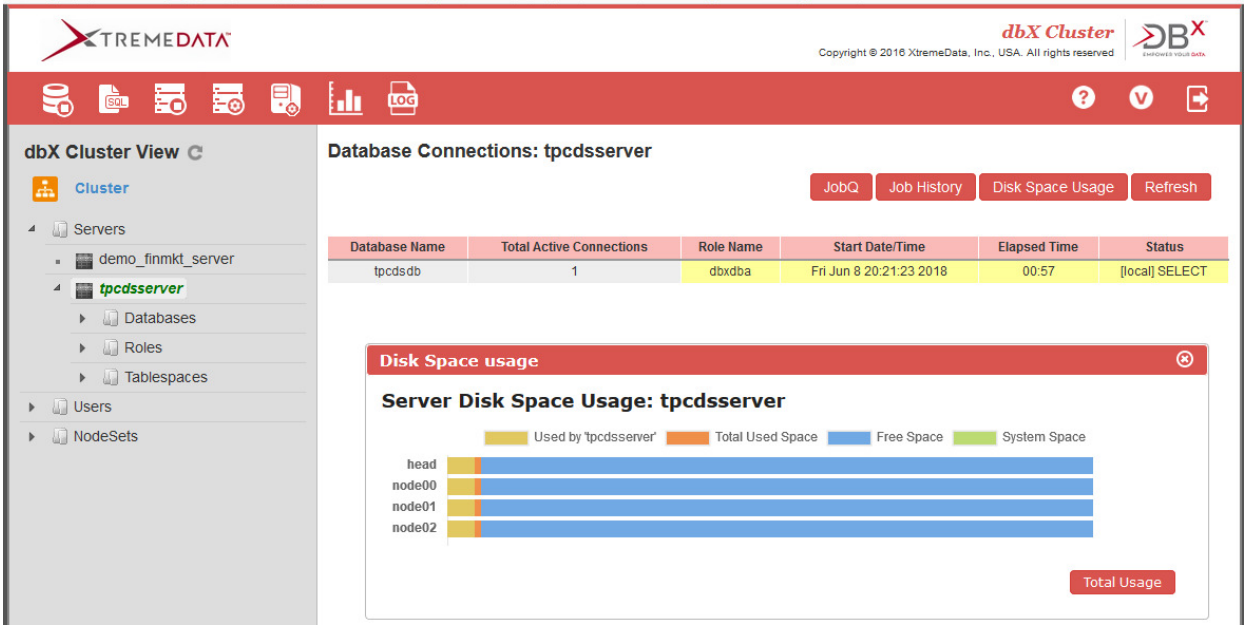
8.1 Database Connections

Select **Monitor Connections**
All database server connections are shown.



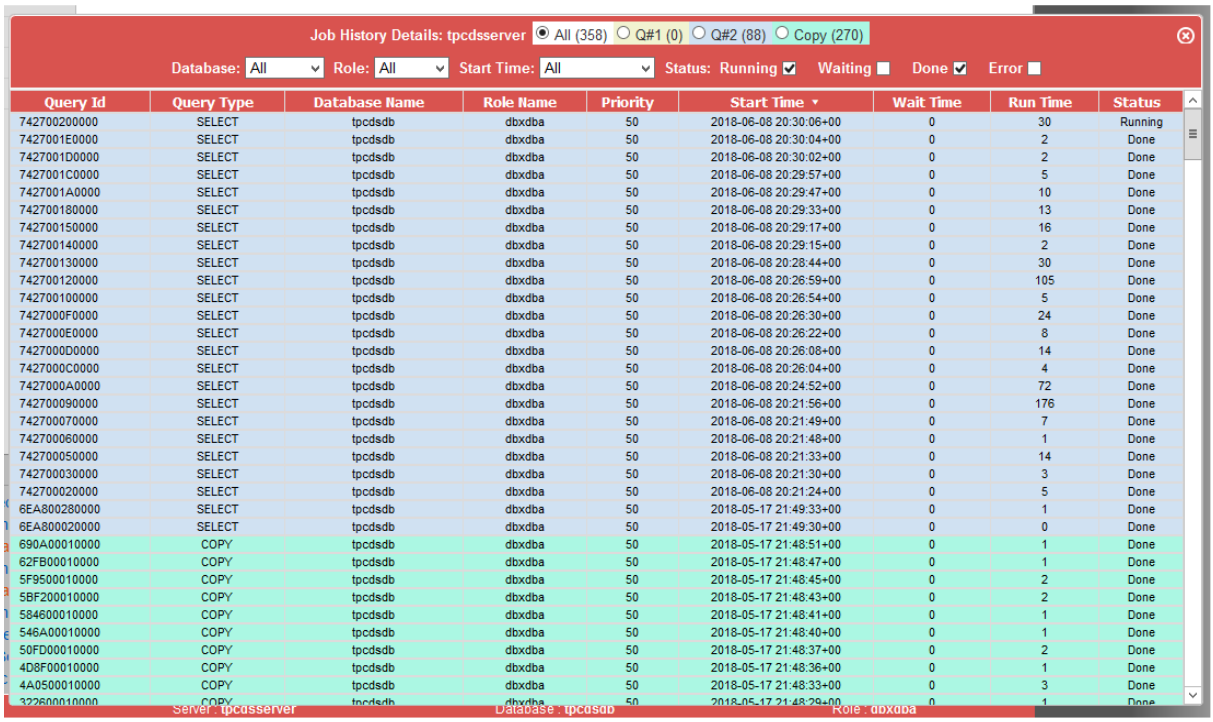
8.2 Disk Space Usage

After connecting to the database server select **Monitor connections → Disk Space Usage**
The disk space usage for every EC2 instance is shown in graphical form.



8.3 Query History

Select **Job History**
All running queries along with the most recently completed queries are displayed.



Query Id	Query Type	Database Name	Role Name	Priority	Start Time	Wait Time	Run Time	Status
742700200000	SELECT	tpcdsdb	dbxdba	50	2018-06-08 20:30:06+00	0	30	Running
7427001E0000	SELECT	tpcdsdb	dbxdba	50	2018-06-08 20:30:04+00	0	2	Done
7427001D0000	SELECT	tpcdsdb	dbxdba	50	2018-06-08 20:30:02+00	0	2	Done
7427001C0000	SELECT	tpcdsdb	dbxdba	50	2018-06-08 20:29:57+00	0	5	Done
7427001A0000	SELECT	tpcdsdb	dbxdba	50	2018-06-08 20:29:47+00	0	10	Done
742700180000	SELECT	tpcdsdb	dbxdba	50	2018-06-08 20:29:33+00	0	13	Done
742700150000	SELECT	tpcdsdb	dbxdba	50	2018-06-08 20:29:17+00	0	16	Done
742700140000	SELECT	tpcdsdb	dbxdba	50	2018-06-08 20:29:15+00	0	2	Done
742700130000	SELECT	tpcdsdb	dbxdba	50	2018-06-08 20:28:44+00	0	30	Done
742700120000	SELECT	tpcdsdb	dbxdba	50	2018-06-08 20:26:59+00	0	105	Done
742700100000	SELECT	tpcdsdb	dbxdba	50	2018-06-08 20:26:54+00	0	5	Done
7427000F0000	SELECT	tpcdsdb	dbxdba	50	2018-06-08 20:26:30+00	0	24	Done
7427000E0000	SELECT	tpcdsdb	dbxdba	50	2018-06-08 20:26:22+00	0	8	Done
7427000D0000	SELECT	tpcdsdb	dbxdba	50	2018-06-08 20:26:08+00	0	14	Done
7427000C0000	SELECT	tpcdsdb	dbxdba	50	2018-06-08 20:26:04+00	0	4	Done
7427000A0000	SELECT	tpcdsdb	dbxdba	50	2018-06-08 20:24:52+00	0	72	Done
742700090000	SELECT	tpcdsdb	dbxdba	50	2018-06-08 20:21:56+00	0	176	Done
742700070000	SELECT	tpcdsdb	dbxdba	50	2018-06-08 20:21:49+00	0	7	Done
742700060000	SELECT	tpcdsdb	dbxdba	50	2018-06-08 20:21:48+00	0	1	Done
742700050000	SELECT	tpcdsdb	dbxdba	50	2018-06-08 20:21:33+00	0	14	Done
742700030000	SELECT	tpcdsdb	dbxdba	50	2018-06-08 20:21:30+00	0	3	Done
742700020000	SELECT	tpcdsdb	dbxdba	50	2018-06-08 20:21:24+00	0	5	Done
6EA800280000	SELECT	tpcdsdb	dbxdba	50	2018-05-17 21:49:33+00	0	1	Done
6EA800020000	SELECT	tpcdsdb	dbxdba	50	2018-05-17 21:49:30+00	0	0	Done
690A00010000	COPY	tpcdsdb	dbxdba	50	2018-05-17 21:48:51+00	0	1	Done
62FB00010000	COPY	tpcdsdb	dbxdba	50	2018-05-17 21:48:47+00	0	1	Done
5F9500010000	COPY	tpcdsdb	dbxdba	50	2018-05-17 21:48:45+00	0	2	Done
5BF200010000	COPY	tpcdsdb	dbxdba	50	2018-05-17 21:48:43+00	0	2	Done
584600010000	COPY	tpcdsdb	dbxdba	50	2018-05-17 21:48:41+00	0	1	Done
546A00010000	COPY	tpcdsdb	dbxdba	50	2018-05-17 21:48:40+00	0	1	Done
50FD00010000	COPY	tpcdsdb	dbxdba	50	2018-05-17 21:48:37+00	0	2	Done
4D8F00010000	COPY	tpcdsdb	dbxdba	50	2018-05-17 21:48:36+00	0	1	Done
4A0500010000	COPY	tpcdsdb	dbxdba	50	2018-05-17 21:48:33+00	0	3	Done

8.4 Other Monitoring Utilities

XtremeData can provide more system monitoring tools on request.

9. Resiliency

XtremeData database cluster operate within our xdc framework, which monitors all EC2 instances and restarts any that fail to respond. Xdc can be configured to forward failure notifications to administrators. If a node fails, it can be stopped and restarted, forcing a reallocation of the failed resources. The data on EBS drives is stored as redundant copies within the AWS environment, so even if a disk fails, no data will be lost.

10. Maintenance and Support

XtremeData offers 2 hour email response time for reported issues during normal business hours (US Central time), and 24 hour email response time outside of normal business hours. Additional maintenance agreements are available on request. Email: support@xtremedata.com

[End of Document]