dbX AWS Setup Guide

Rev 2.1
Mar 2014

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

www.xtremedata.com

Overview

This document describes the process to launch an XtremeData dbX system on Amazon Web Services. A multi-node dbX system comprises one “head” node and multiple “data” nodes which launched through the AWS console or a Cloud Formation script. It may also be deployed as a single node system with a virtual data node on the head node.

As sample cloud formation template is available at:
https://s3.amazonaws.com/xtremedata-dbx-public/dbx_cluster.template

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

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.

[This procedure represents one method to launch a dbX system. Other methods are possible, such as running a script or launching from Cloud Formation...]

This document describes the procedure for setting up AWS instances, creating a cluster from these instances and starting the dbX database engine. Stopping and re-starting of the cluster is also covered.

Single-node setup

Step 1: Set up a security group to enable the administration tool
Step 2: Configure and launch the head node
Step 3: Initialize cluster
Step 4: Start cluster and dbX

Multi-node Setup

The steps involved in launching a multi-node dbX system on AWS are:

Step 1: Set up a security group to enable all nodes to communicate with each other
Step 2: Configure and launch the head node and data nodes
Step 3: Initialize cluster
Step 4: Start cluster and dbX

Step 1: Setup Security Group

The starting point is the EC2 management console:

In order to run a dbX cluster, all nodes in the cluster must belong to a security group that allows the nodes to communicate with each other over any IP port. To create a new security group, go to the EC2 console and click on the "Security Groups" tab.

Clicking on "Create Security Group" will bring up a dialog box. Fill in the name and description of your choice. In this example, we have chosen "dbxcluster" as the name of the security group:

The next step is to select the security group and change the settings on the "inbound" tab in the lower window and select "edit".

Set the port range to "22" and leave the source as 0.0.0.0/0 then click "Add Rule".

Next set the port range "0-65535" and set the source to the Group ID of your security group then click "AddRule".

Set the port range to "2400" and the source as 0.0.0.0/0 then click "Add Rule". Note that this step is optional and is only used for the GUI (xdAdmin) tool. For security, an alternate method to access the GUI is by tunneling port 2400 over ssh.

When finished with adding the rules, click "Save". Now you have set up the security group, next step is to launch the head node.

Step 2: Launch Node(s)

The head node will be launched first, then the data nodes.

Instance type
Select "Instances" under the "INSTANCES" menu on the left side of the screen. Then select "Launch Instance".

Select the "AWS Marketplace" button, and enter "dbx" into the search window.

Select the option titled "XtremeData dbx (HVM)".

Choose the instance type and hit "Review and Launch".

Security Group
Select "Edit security groups"

Change the selection to "Select an existing security group"

Check the box for the security group created earlier and hit "Review and Launch".

Number of instances and zone
Select "Edit instance details".

Choose the number of instances (choose "1" for head or single node system).

Choose the availability zone (note: all nodes in a multi-node cluster should reside in the same zone).

* **User Data is needed for multi-node clusters only.** This name must be the same for all nodes in the cluster.

Under “Advanced Details” enter the cluster name (For example, enter “cluster1”) in the “User data” box as text. If this is a data node in a multi-node cluster, then also add the IP address of the head node. Hit “Review and Launch”.

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 can be used. Select “Edit storage” and add storage devices. All storage devices that are made available before initialization of the cluster 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). For each device, select “Add New Volume”. When finished, select “Review and Launch”.

Add a Tag
As a convenience, tag your nodes in the AWS console with a name. Select “Edit tags”. Add a value after “Name”. Hit “Review and Launch”.

Select “Launch”. You will be prompted for your key pair. Make the appropriate selection and hit “Launch instances”.

The launch status screen will be displayed. Hit “View Instances”.

For multi-node clusters: Note the availability zone and capture the private IP address – these will be needed for the data nodes. [*Alternately, you can associate an Elastic IP with the head node and save its Public DNS name. Note that using an Elastic IP will make the process of restarting the cluster much easier because the data nodes can use the same IP each time.*]

Repeat above steps to launch data nodes – note that all the data nodes can be launched together (by setting the number of instances under “Edit instance details”).

Step 3: Initialize cluster

WARNING: Initialization of the cluster is done once when the cluster 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 de-attached 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"

There are two ways to initialize the cluster. One destructively initializes all EBS storage (-i option), the other (-n option) assumes that the user has already taken care of the storage and mounted it to /volumes/data for dbx to use. In both cases, ephemeral storage is initialized and mounted to /volumes/temp/dbx-temp for dbx temporary files.

Initialize cluster by typing; "cluster_init [-p] -i *#nodes*", where *#nodes* = number of data nodes you launched in Step 3. Note that "-p" is an option to make the head node also function as a data node. A single node configuration would use "–p –i 0"

- This will initialize all storage devices that are attached and make them available for dbX.
- Nodes will be enumerated and final configuration will be performed.
- This is a DESTRUCTIVE process – no previous data or configuration will be preserved!

Or initialize cluster by typing; "cluster_init [-p] -n *#nodes*", where *#nodes* = number of data nodes you launched in Step 3. Note that "-p" is an option to make the head node also function as a data node. A single node configuration would use "–p –i 0"

- This will only initialize ephemeral storage devices that are attached and make them available for dbX.
- Nodes will be enumerated and final configuration 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"

Step 4: Start cluster & dbX

Start the cluster by typing "cluster_start"

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

Now your instances are initialized and running as a cluster with one head and multiple data nodes. Next step is to start the dbX database engine.

Type "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. To do this, at a system prompt, type "passwd"

You are now up and running! You can now login to the head as user "dbxdba" to create servers and databases. 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".

Stopping the cluster

To stop the cluster, login to the head node as "ec2-user" and type "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 all the nodes in your cluster and under "Instance Actions" select "Stop".
Note: if your data is on ephemeral storage (no EBS drives were configured), it will be lost!

At the prompt, confirm the nodes to be stopped.

Re-starting the cluster

To restart the cluster, select only the head node, and under Instance Actions select "Start".

Select "Yes, Start".

For a multi-node cluster, wait for the head node to boot to capture its new IP address – please note that this will change each time you stop and re-start. Select and copy the IP address.

Select each data node instance individually and under "Instance Actions" select "View/Change User Data".

Alter the user data to reflect the new head node IP address.

Select "Yes, Change".

Once each data node has the new head node IP, select all the data nodes and start them.

Select "Yes, Start" when prompted.

Once all instances have 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, as you did in Step 3, otherwise all data will be lost! Only repeat Step 4:

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

Start cluster by typing "cluster_start".

Start the dbX database engine by typing "dbx_start".

User-Data: Alternate Ways of Configuring

If user-data is not populated at instance launch, or it is desirable to change user-data without stopping the nodes, this can be done at runtime using a number of command line methods. Note that changing user-data via the command line creates a local user-data repository on the node and does not alter the user-data maintained in Amazon's meta-data repository. Each node will check both repositories to determine which one contains the last modified data and will use that data. The local repository of user-data can also be deleted.

On each node, its user-data can be changed via the command line call "set_cluster_data" (shown with -h for help):

```
set_cluster_data -h
usage:  set_cluster_data [options] [cluster-name [head-host-name]]

options:  -d   delete data that has been set, revert to AWS user-data
          -e   use $DBX_CLUSTER_NAME and $DBX_CLUSTER_HEAD
          -l   don't set anything, display current cluster data
          -v   verbose
          -q   quiet
          --   end of options

If cluster-name and head-host-name are not provided on command line and
-e is not used, cluster-name and head-host-name are read from stdin,
one per line
```

For example, the head node only needs the cluster name:

```
set_cluster_data my-cluster-name
```

Each data node will need the cluster name and the head host name:

```
set_cluster_data my-cluster-name ec2-xx-xx-xx-xx.compute-1.amazonaws.com
```

If there are numerous data nodes in the system, it may be more convenient to run a group command on the head node to populate all the data nodes with the same user-data. Note that this method requires an ssh agent to be running with the proper key loaded and agent forwarding enabled. Also, the cluster name in the head node user-data must be set first (since it is implicitly propagated to the data nodes with this command).

```
set_cluster_data_nodes -h
usage: set_cluster_data_nodes [-q] [-H head-host] {[-f hosts-file]|host host ...}
-q  quiet
-H  head node host name (optional)
-f  file with data node host names (1 per line)
```

The head host name is usually not needed as it is collected from the AWS meta-data repository. This also works with an Elastic IP.

[End of Document]