TECHNICAL VALIDATION OF BLUEDATA EPIC WITH EMC ISILON

ETD Solution Architecture Compatibility and Performance Testing Validation Brief

ABSTRACT

This document captures details on the technologies, results, and environment used to perform various functionality tests to demonstrate compatibility between BlueData EPIC Software and EMC technologies described herein.

November 3 2015

Revision 1.3

REDEFINE

Technical Guide – ISV Partner Validation

EMC²

To learn more about how EMC products, services, and solutions can help solve your business and IT challenges, contact your local representative or authorized reseller, visit www.emc.com, or explore and compare products in the EMC Store

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

The information in this publication is provided “as is.” EMC Corporation makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

TABLE OF CONTENTS

INTRODUCTION 5

Purpose 5

Bluedata and Isilon joint value proposition 5

Technology Overview 5

Relevant Partner Features 7

Relevant EMC Features 7

VALIDATION SCOPE 7

Partner Technologies 7

EMC Technologies 7

TESTING ENVIRONMENT 7

Architecture Illustration 7

Resources 8

Bluedata EPIC documentation 9

BLUEDATA installation and Configuration 9

EMC ISILON Installation and Configuration 15

Supporting Information and Considerations 21

VALIDATION TESTS 21

Compatibility Testing 21

TEST # 1: Connect to Kerberized Isilon HDFS via BlueData DataTap object. 21

TEST # 2: Connect to Isilon via BlueData NFS DataTap........ 24

TEST # 3: Browse Kerberized Isilon HDFS DataTap from CDH (Cloudera) Compute Cluster 25

TEST # 4: Browse Kerberized Isilon HDFS DataTap from HDP (Hortonworks) Compute Cluster 27

TEST # 5: Run a Hive query against data stored in Isilon Kerberized HDFS via Hue console 28

TEST # 5: Integrate with Cloudera Navigator for metadata search and auditing 31

TEST # 6: Preliminary performance benchmarks using standard development/test hardware in BlueData Labs. 37

APPENDIX 39
Teragen command 39
Terasort command 41
Teravalidate command... 42

INTRODUCTION

PURPOSE

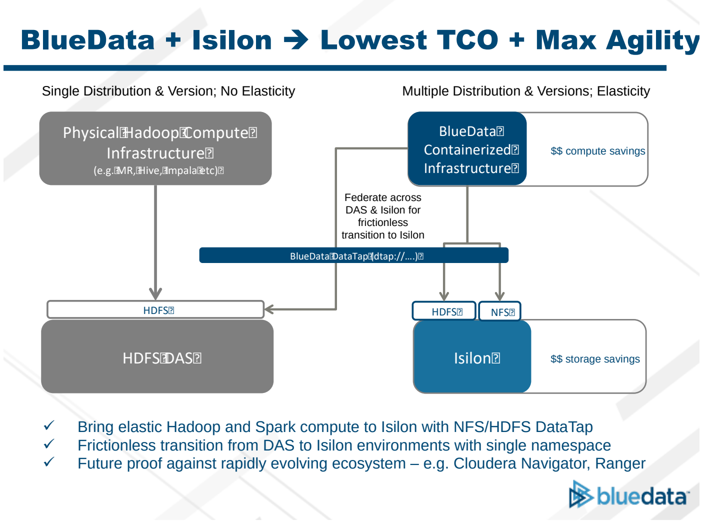
The intent of this document is to capture details pertaining to the technologies and environment used to confirm compatibility between BlueData EPIC Software and EMC Isilon storage system

BLUEDATA AND ISILON JOINT VALUE PROPOSITION

The BlueData EPIC™ (Elastic Private Instant Clusters) software platform solves the infrastructure challenges and limitations that can slow down and stall Big Data deployments. With EPIC software, you can spin up Hadoop or Spark clusters – with the data and analytical tools that your data scientists need – in minutes rather than months. Leveraging the power of containers and the performance of bare-metal, EPIC delivers speed, agility, and cost-efficiency for Big Data infrastructure. It works with all of the major Apache Hadoop distributions as well as Apache Spark. It integrates with each of the leading analytical applications, so your data scientists can use the tools they prefer. You can run it with any shared storage environment, so you don't have to move your data. And it delivers the enterprise-grade security and governance that your IT teams require. With the BlueData EPIC software platform, you can provide Hadoop-as-a-Service or Spark-as-a-Service in an on-premises deployment model.

EMC Isilon Scale-out Storage Solutions for Hadoop combine a powerful yet simple and highly efficient storage platform with native Hadoop integration that allows you to accelerate analytics, gain new flexibility, and avoid the costs of a separate Hadoop infrastructure. BlueData EPIC Software combined with EMC Isilon shared storage provides a comprehensive solution for compute + storage.

BlueData and Isilon share several joint customers and opportunities at leading financial services, advanced research laboratories, healthcare and media/communication organizations. As such, the following diagram describes the key value propositions:



BlueData + Isilon → Lowest TCO + Max Agility

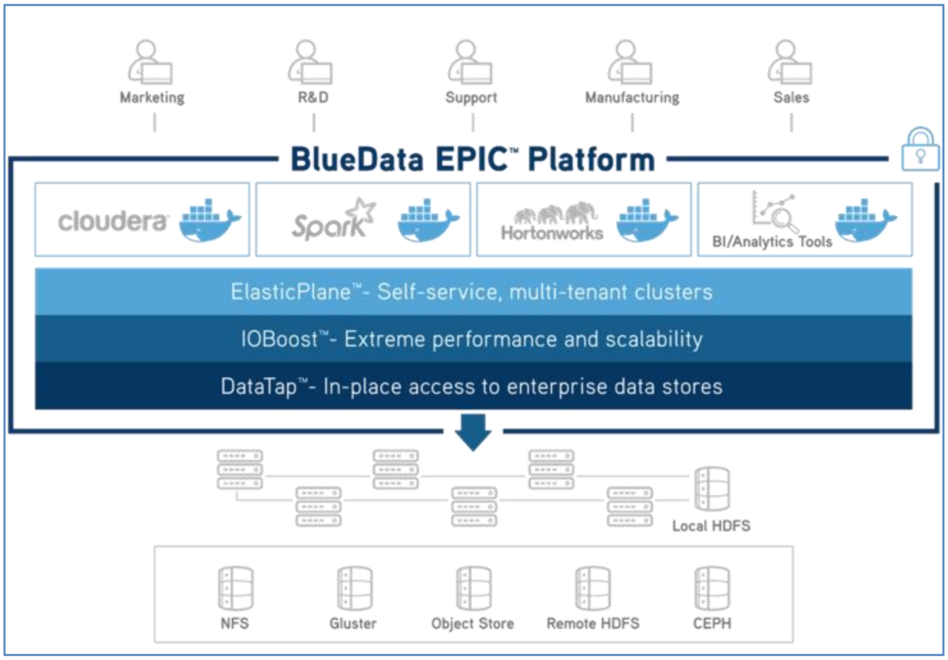
Single Distribution & Version; No Elasticity
Multiple Distribution & Versions; Elasticity
Physical Hadoop Compute Infrastructure
(e.g. MR, Hive, Impala etc)
BlueData Containerized Infrastructure
$$ compute savings
Federate across DAS & Isilon for frictionless transition to Isilon
BlueData DataTap (dtap://....)
HDFS
HDFS DAS
HDFS
NFS
Isilon
$$ storage savings
bluedata

- ✓ Bring elastic Hadoop and Spark compute to Isilon with NFS/HDFS DataTap
- ✓ Frictionless transition from DAS to Isilon environments with single namespace
- ✓ Future proof against rapidly evolving ecosystem – e.g. Cloudera Navigator, Ranger

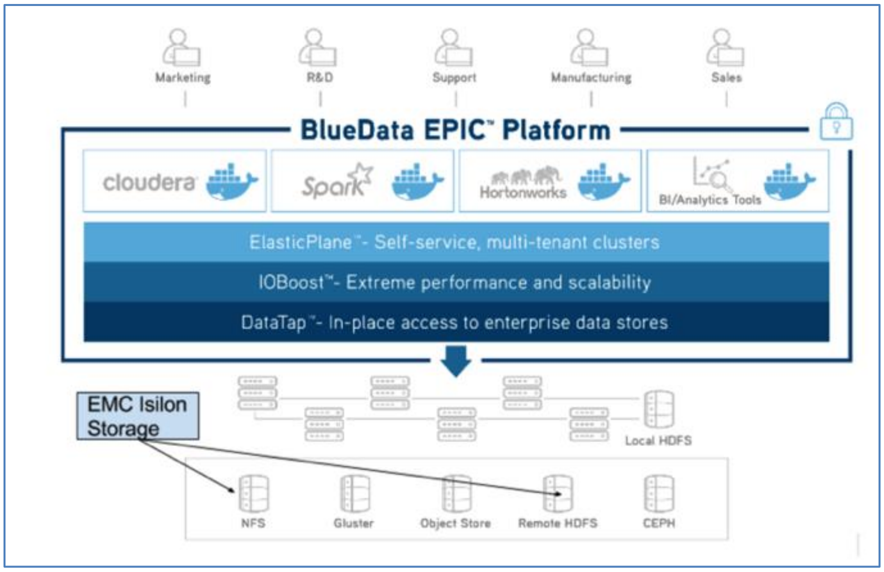
This paper describes the process of validating Hadoop applications running in virtual clusters on the EPIC platform with data stored on the EMC Isilon storage device using either NFS or HDFS data access protocols.

TECHNOLOGY OVERVIEW

The BlueData EPIC platform is an infrastructure software platform, purpose built for to virtualize/containerize Big Data/Hadoop applications to increase agility, significantly reduce infrastructure cost by increasing utilization while maintaining security and control. As such, BlueData software enables Big Data-as-a-service on-premises.



The point of interconnection with EMC Isilon storage is the EPIC DataTap feature. DataTap surfaces an HDFS abstraction layer so that any Big Data application can be run unmodified, but at the same time implements a high-performance connection to remote data storage systems via NFS, HDFS, and object RESTful protocols. The result allows unmodified Hadoop applications to run against data stored in remote NFS, HDFS, and object storage systems without loss of performance.



RELEVANT PARTNER FEATURES

Table 1 List of Partner technology features relevant to the validation consideration

DESCRIPTION	DETAIL
BlueData EPIC 2.0 Software	Infrastructure software that supports virtual 'compute' clusters of CDH, HDP and Spark stand-alone. CDH clusters can be configured with Cloudera Manager (and Cloudera Navigator) while HDP clusters can be configured with Ambari (and other services enabled by Ambari) Another key feature is **DataTap** that provides a HDFS protocol URI dtap:// that allows virtual 'compute' clusters running on BlueData to leverage external storage systems such as Isilon via HDFS or NFS. Note: CDH and HDP clusters are **NOT** configured with Isilon directly. These clusters connect to Isilon via BlueData DataTap functionality

RELEVANT EMC FEATURES

Table 2 List of EMC technology features relevant to the validation consideration

DESCRIPTION	DETAIL
EMC Isilon	HDFS Module, NFS

VALIDATION SCOPE

PARTNER TECHNOLOGIES

Table 3 List of Partner technologies used in the validation process

TECHNOLOGY - MODEL	TECHNOLOGY VERSION	TECHNOLOGY INSTANCE (PHYSICAL/VIRTUAL)
BlueData EPIC	EPIC 2.0	Controller Node: Physical Worker Node: Physical
Cloudera virtual Hadoop cluster	CDH 5.4.3 with Cloudera Manager Enterprise	1 Master Node: Virtual (Docker Container) 3 Worker Node: Virtual (Docker Container
Hortonworks virtual Hadoop cluster	HDP 2.3 with Ambari 2.1	1 Master Node: Virtual (Docker Container) 3 Worker Node: Virtual (Docker Container)

EMC TECHNOLOGIES

Table 4 List of EMC technologies used in the validation process

TECHNOLOGY - MODEL	TECHNOLOGY VERSION	TECHNOLOGY INSTANCE (PHYSICAL/VIRTUAL)
Isilon X210	Isilon OneFS v7.2.1.0 B_7_2_1_014(RELEASE)	Physical (8 node system)

Data Access Methods

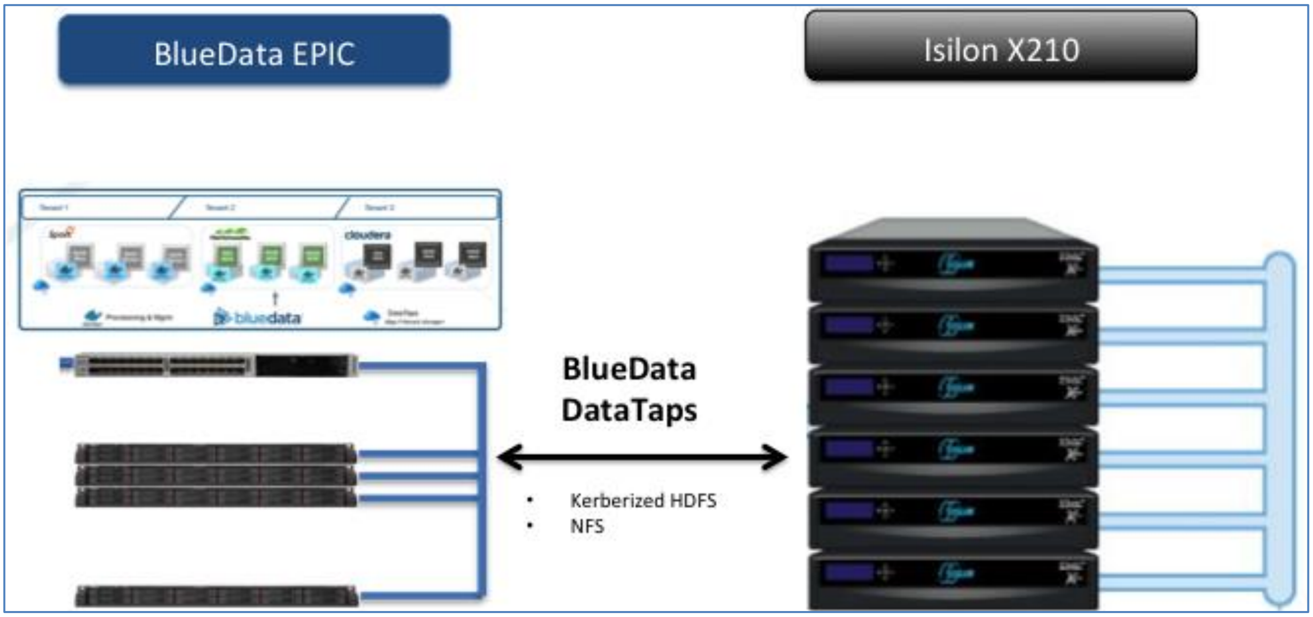
Table 5 List of data access methods and technologies used in the validation process

DATA ACCESS METHOD	DATA ACCESS STRUCTURE	DETAILS
Isilon HDFS	API	Over the wire HDFS protocol implemented in Isilon

TESTING ENVIRONMENT

ARCHITECTURE ILLUSTRATION

Figure 1. Illustration of logical layout of Partner and EMC technologies used for validation



RESOURCES

Table 6 List of resources and technologies used to support the validation process

DESCRIPTION	VERSION	DETAILS & SPECIFICATIONS
HARDWARE		
(6) BlueData Workers		6 workers: 10G NIC, 64GB RAM, 4 cores, 8 cpu, 5 1TB drives 1 controller: Same as the worker, but with 128GB ram CPU: Intel(R) Xeon(R) CPU E5-2609 v2 @ 2.50GHz All boxes have the same 5 drives: 1TB SSDH hybrid drives (ssd + disk)
(1) BlueData Controller		6 workers: 10G NIC, 64GB RAM, 4 cores, 8 cpu, 5 1TB drives 1 controller: Same as the worker, but with 128GB ram CPU: Intel(R) Xeon(R) CPU E5-2609 v2 @ 2.50GHz All boxes have the same 5 drives: 1TB SSDH hybrid drives (ssd + disk)
(8) Isilon X210		8 node system
CDH Virtual Cluster		***Functional testing and validations***
(1) Master Node		Master: XXL - 8 VCPU, 24576 MB RAM
(6) Slave Nodes		Worker: Extra Large - 4 VCPU, 12288 MB RAM

BLUEDATA EPIC DOCUMENTATION

1. About BlueData EPIC
2. Installation Guide
3. User-Administrator Guide

BLUEDATA INSTALLATION AND CONFIGURATION

Note: There are various pre-requisites for BlueData EPIC Software. These are related to Hosts, Operating System, Networking and High Availability. Details can be found here

Phase One: Installing the BlueData EPIC Controller

To install EPIC from the command line:

1. Install Red Hat Enterprise Linux 6.5 or CentOS 6.5 on the hosts that you will use for the Controller and Worker nodes.
2. Log into the host that you will be using as the Controller node using the root account and password.
3. Download the EPIC Enterprise binary (.bin) from BlueData Software, Inc. to the host that you will use as the Controller node. The size of the download will depend on the distribution(s) included and the flavor of the .bin file.
4. Make the .bin file executable by executing the command
5. chmod a+x bluedata-epic-<type>-<os>-<package>-release-<version>-<build>.bin

Where:

- <type> can be either entdoc (for EPIC) or onedoc (for EPIC Lite)

- <os> is the operating system supported by this .bin file. This can be either centos (for CentOS) or rhel (for Red Hat Enterprise Linux).

- <package> is the EPIC package type. This will be either eval (EPIC Lite), full (CentOS with all App Store images and OS packages included), or minimalplus (RHEL with all App Store images included).

- <version> is the EPIC version being downloaded.

- <build> is the specific EPIC build number being downloaded. 5. Run the executable binary from the Linux console as the root user by typing ./<epic>.bin --floating-ip <address/range>, where:

- <epic> is the full name of the .bin file (see Step 4, above).

- <address/range> is the CIDR to use, such as 192.168.25.10/24. This range of addresses allows network access from outside the EPIC platform to the virtual nodes that EPIC will create as part of future clusters.

The installer checks the integrity of the EPIC bundle and then extracts the bundle contents and performs some initial checks.

The installer checks to see which network interface card(s) (NIC) have Internet connectivity.

- If multiple NICs can connect to the Internet, then a prompt appears asking you to select the NIC to use.

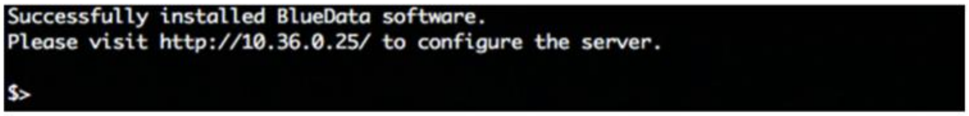
- If only one NIC can connect to the Internet, then the installer bypasses this step.

- If no NICs can connect to the Internet, the installer asks for the IP address of the proxy server being used for your network. If prompted, enter either the number corresponding to the NIC to use to connect to the Internet or the IP address of the proxy server, and then press [ENTER].

The End User License Agreement (EULA) appears. Read through the EULA, pressing [SPACE] to page through the content. Once you have viewed the entire EULA, press y to accept it and continue installing EPIC.

EPIC installs on the Controller node. A series of messages appear during the installation. The following message appears once the installation is complete:

Figure 2.1: Installation complete



```
Successfully installed BlueData software.
Please visit http://10.36.0.25/ to configure the server.

$>
```

Figure 2.1: Installation complete

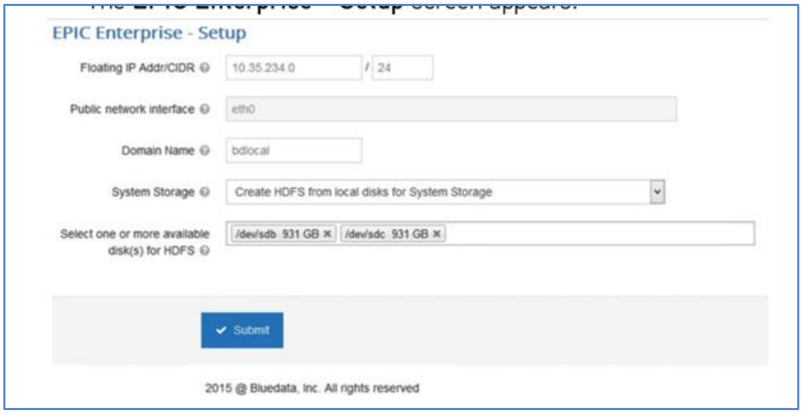
This concludes the first phase of the EPIC installation. Note the URL provided, as you will use this to continue configuring EPIC.

Phase Two: Completing the Installation

The next step of the EPIC installation process is to configure EPIC using a Web browser to access the application. To do this:

1. Open a Web browser and navigate to the URL provided at the end of the command line installation process.

The EPIC Enterprise - Setup screen appears.



EPIC Enterprise - Setup

Floating IP Addr/CIDR: 10.35.234.0 / 24

Public network interface: eth0

Domain Name: bdlocal

System Storage: Create HDFS from local disks for System Storage

Select one or more available disk(s) for HDFS: /dev/sdb 931 GB × /dev/sdc 931 GB ×

2015 @ Bluedata, Inc. All rights reserved

3. The read-only Public network interface field displays the NIC that you selected for Internet access during the command line installation. Each host in the EPIC platform must use the same NIC to access the Internet. For example, if you selected the eth0 NIC on the Controller node, then the eth0 NIC on each Worker node must also be able to reach the Internet.

4. Use the System Storage pull-down menu to select the type of system storage to use for the EPIC platform. The available options are:

If the nodes each have a second or third hard drive as described in the About EPIC Guide and you want to create local system storage using HDFS with Kerberos protections, then select Create HDFS from local disks for system storage.

. To use an existing external HDFS file system as system storage, select Use existing HDFS for System Storage and then enter the parameters for “HDFS”

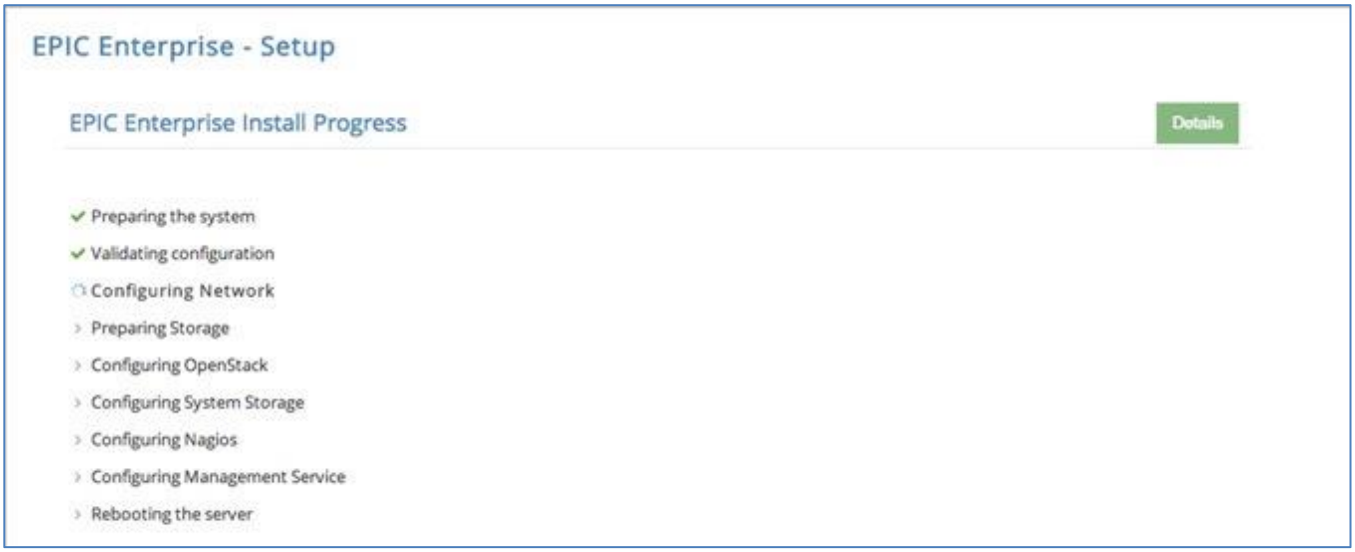
. To use an existing external NFS file system as system storage, select Use existing NFS for System Storage and then enter the parameters described for “NFS”

Note: Configuring system storage does not preclude the ability for clusters running on BlueData to reach out to other storage systems using DataTap. The system storage is just a default scratch space for each tenant and is referred to as Tenant Storage.

If you are creating local HDFS system storage, then select one or more hard drive(s) to use for this storage in the Select one or more available disk(s) for HDFS field.

Click Submit to finish installing EPIC on the Controller node. EPIC displays a popup indicating that the installation process has started successfully. Figure 2.3: Installation Started popup

This popup is soon replaced by a status summary as the installation completes. If you like, you may click the green Details button to open a popup that displays additional information about the installation. Please allow about 20 minutes for this process to complete (actual time will vary depending on various factors).


EPIC Enterprise - Setup

EPIC Enterprise Install Progress

- ✓ Preparing the system
- ✓ Validating configuration
- Configuring Network
- › Preparing Storage
- › Configuring OpenStack
- › Configuring System Storage
- › Configuring Nagios
- › Configuring Management Service
- › Rebooting the server

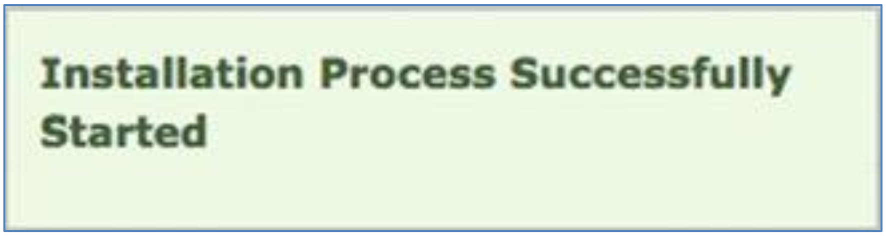
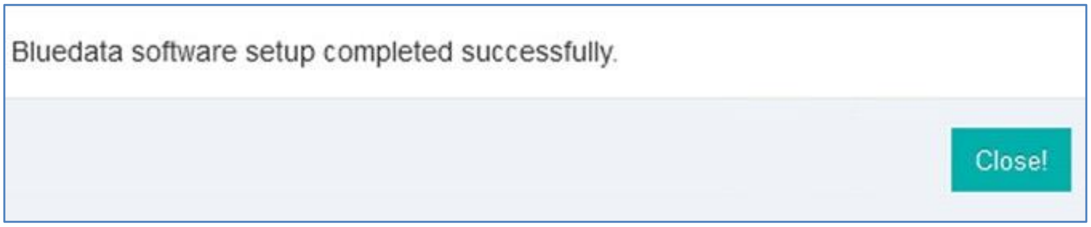

Installation Process Successfully Started

Figure 2.4: Completing the EPIC installation

The Bluedata software setup completed successfully popup appears when the installation process is completed. Click the Close button to exit to the EPIC Login screen,

Figure 2.5: Installation completed


Bluedata software setup completed successfully.

Proceed to the next section to begin adding Worker nodes.

Phase Three: Adding Worker Nodes

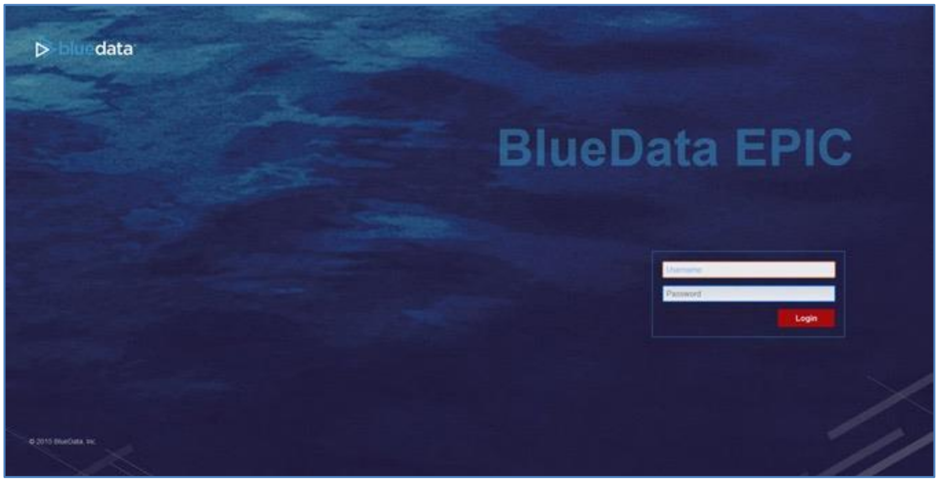
Once you have finished installing EPIC on the Controller node, the final step is to add the Worker nodes to the EPIC platform. As such, workers provide additional CPU and memory so that you can create more virtual nodes. Each Worker node or host must conform to the system requirements listed in the About EPIC Guide and any applicable Deployment Guide.

Note: This section provides a high-level overview intended to help you get up and running with EPIC as quickly as pos- sible. Please see the User/Admin Guide for information about this screen and any applicable Deployment Guide for further directions.

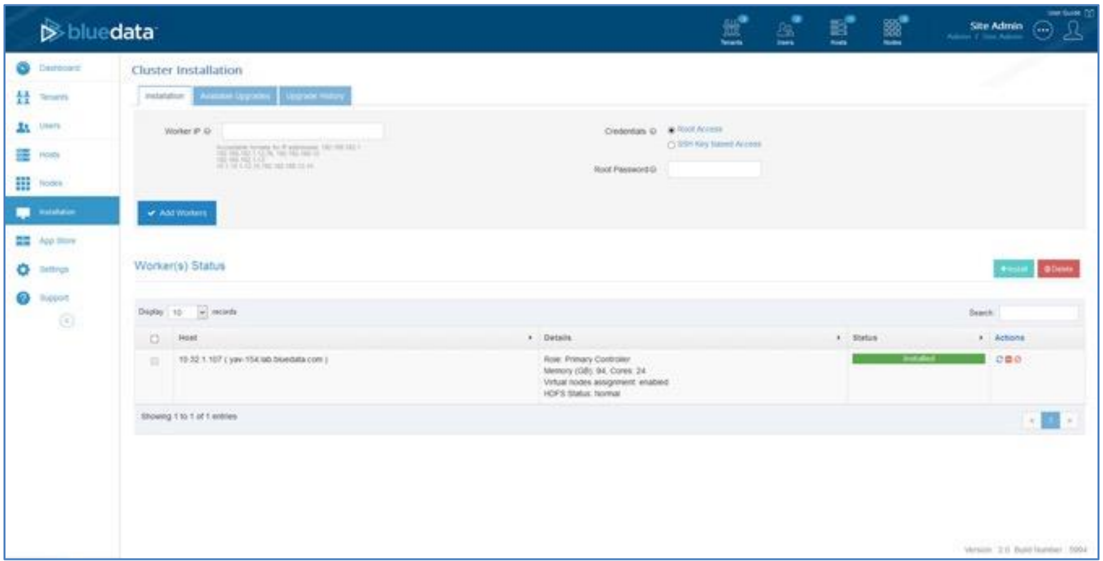
To add Worker nodes:

1. Access the EPIC Login screen by opening a Web browser and navigating to the Controller IP address.

2. Enter your username and password in the appropriate fields and then click the red Login button.



3. In the main menu, select Installation. The Cluster Installation screen appears.



Cluster Installation

Worker IP

Credentials ● Root Access ○ SSH Key based Access

Root Password

Worker(s) Status

Display 10 records Search:

□	Host	Details	Status	Actions
□	10.32.1.107 (yav-154.lab.bluedata.com)	Role: Primary Controller Memory (GB): 94, Cores: 24 Virtual nodes assignment: enabled HDFS Status: normal	Installed	

Showing 1 to 1 of 1 entries

Version: 2.0 Build Number: 5004

4. Enter the IP address(es) of the Worker node(s) that you wish to add to the EPIC platform in the Worker IP field. You may enter IP addresses as follows, being sure not to add any spaces:

- Single IP address: Enter a properly formatted IP address, such as 10.10.1.1.

- Multiple IP addresses: Enter the first three octets of the IP addresses, and then separate each digit of the fourth octet with a commas, such as 10.10.1.1,2,5,8. In this example, EPIC will install four Worker nodes with IP addresses of 10.10.1.1, 10.10.1.2, 10.10.1.5, and 10.10.1.8.

- Multiple IP addresses: Enter multiple IP addresses separated by commas, such as

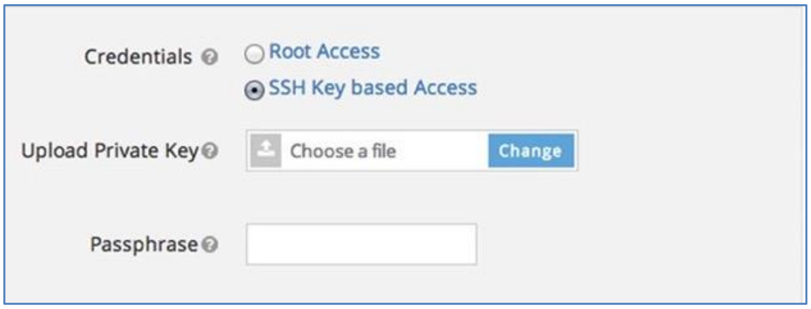
10.10.1.1,10.10.1.2,10.10.1.5,10.10.1.8. In this example, EPIC will install four Worker nodes with the same IP addresses as the previous example.

- IP address range: Enter an IP address range, such as 10.10.1.1-8. In this example, EPIC will install eight Worker nodes with IP addresses from 10.10.1.1 to 10.10.1.8.

- Combination: Use a combination of the above methods, such as 10.10.1.1,10.10.1.2,5,8,10.10.1.9-12.

5. Select how to access the Worker node(s). Your available options are:

cCicking the Private Key field to open a standard File Upload dialog that allows you to browse for and select the key file. If the key requires a pass phrase, enter that phrase in the Passphrase field. The uploaded private key will only be used for initial access to the worker nodes and will not be permanently stored.



Credentials ○ Root Access ● SSH Key based Access

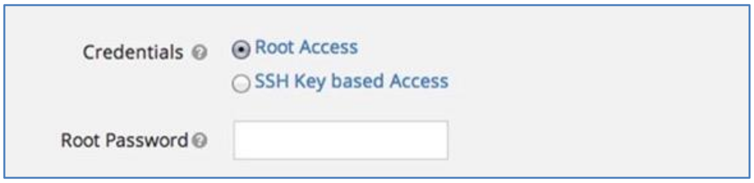
Upload Private Key: Choose a file

Passphrase:

Root access: Check the Root Access radio button and then enter the root password for the Worker node(s) you are installing in the Root Password field..

Figure 2.8: Root Access information

SSH Key: If the Worker node(s) have a public key installed to allow password-free access, then you may check the SSH Key based Access radio button. Upload the private key by



Credentials ● Root Access ○ SSH Key based Access

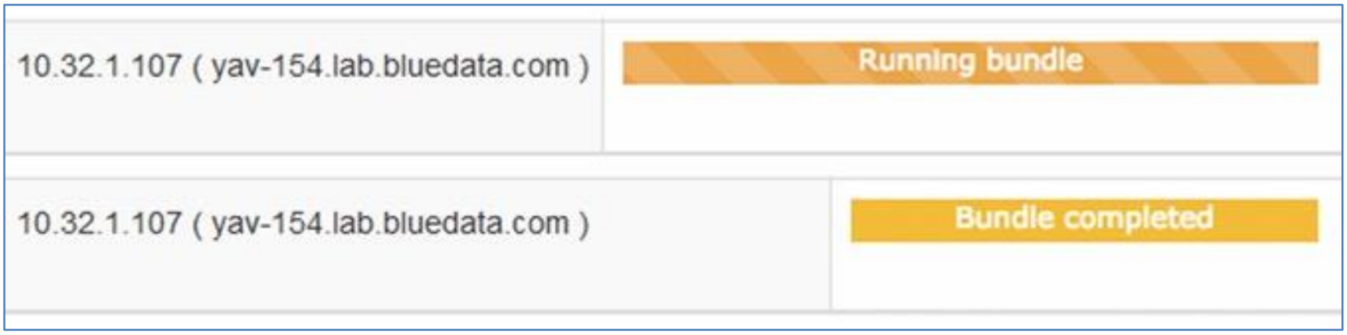
Root Password:

Figure 2.9: SSH Key information

6. Click the blue Add Workers button to install the selected Worker node(s).

EPIC will prepare the selected Worker node(s). A series of colored indicator bars in the Worker(s) Status table will display the installation progress for each Worker node, as follows:

Figure 2.10: Worker node installation progress



10.32.1.107 (yav-154.lab.bluedata.com) Running bundle

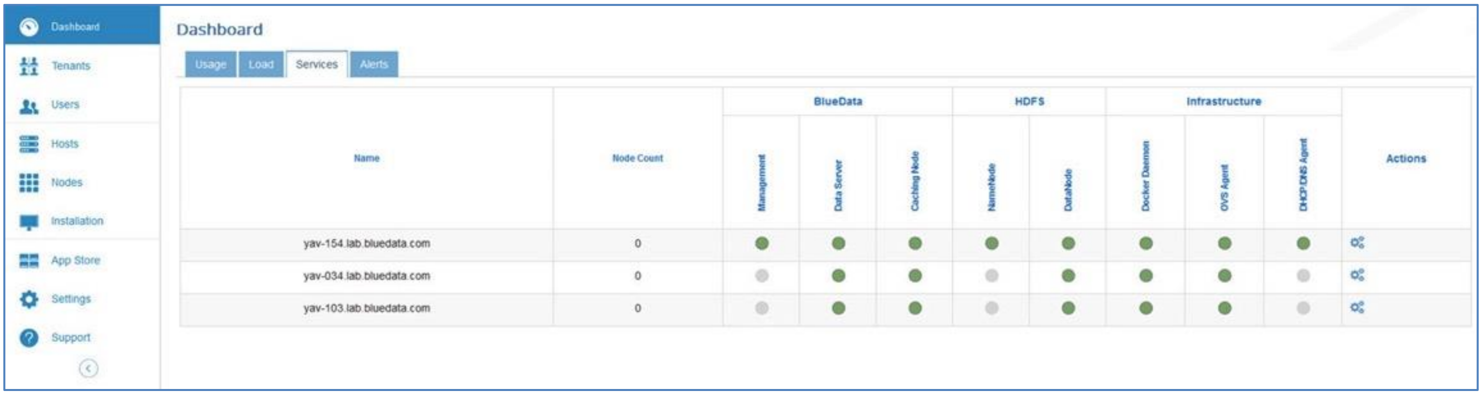
10.32.1.107 (yav-154.lab.bluedata.com) Bundle completed

-

Once you have finished adding Worker node(s), you may view the status of each node by clicking Dashboard in the main menu to open the Dashboard screen, and then selecting the Services tab, which presents detailed status information for each Worker node.

Note: You may only perform one set of Worker node instal- lations to one or more node(s) at once. To save time, con- sider adding all of the Worker nodes at once by entering multiple IP addresses as described above.

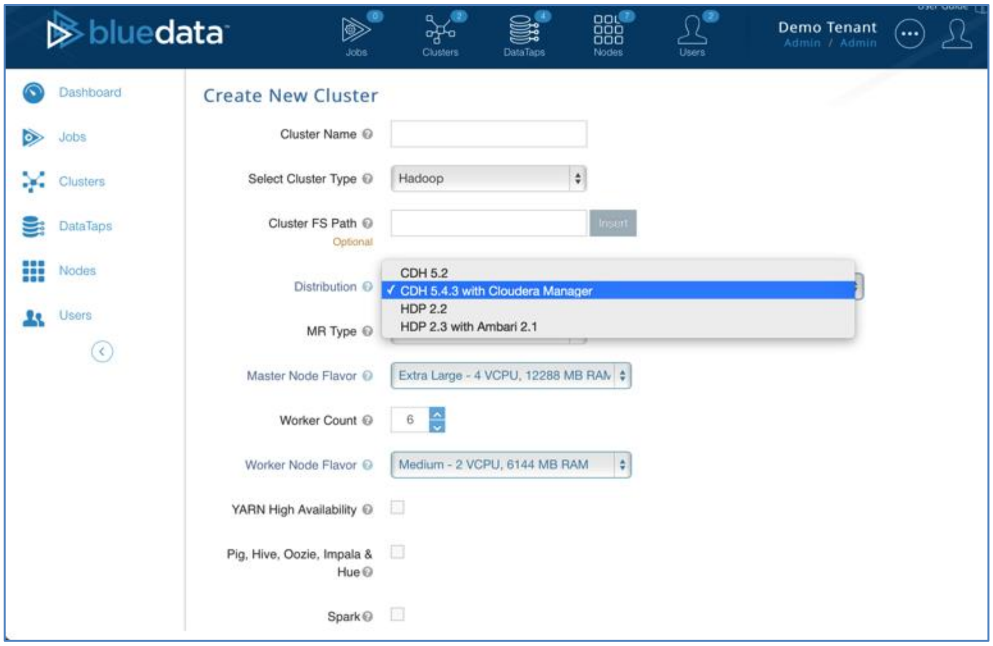
Note: Verify that the Worker node(s) have finished rebooting before attempting to create a virtual cluster. Check the Ser- vice Status tab of the Dashboard screen to ensure that all services are green before proceeding.



Name	Node Count	BlueData			HDFS		Infrastructure			Actions
		Management	Data Server	Caching Node	NameNode	DataNode	Docker Daemon	OVS Agent	DHCP/DNS Agent	
yav-154.lab.bluedata.com	0	●	●	●	●	●	●	●	●	⚙
yav-034.lab.bluedata.com	0	○	●	●	○	●	●	●	○	⚙
yav-103.lab.bluedata.com	0	○	●	●	○	●	●	●	○	⚙

Figure 2.15: Site Admin Dashboard screen - Service Status tab

Once the installation of the workers is complete, you can login to BlueData UI, change to Demo Tenant and create Hadoop or Spark clusters using this simple one page form. BlueData automatically provisions the virtual nodes (Docker containers); does the networking and isolation; deploys the software for the relevant distribution and configures the applications (YARN, M/R, Hive etc).



Create New Cluster

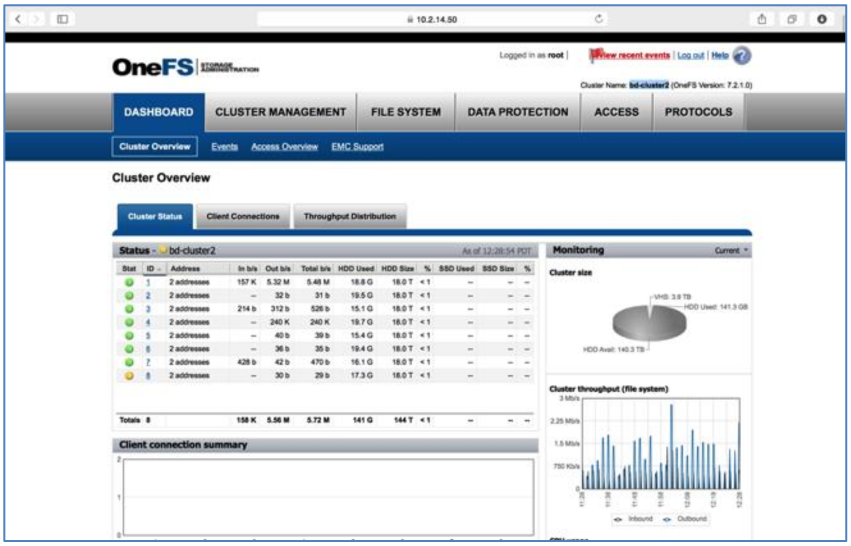
Cluster Name	
Select Cluster Type	Hadoop
Cluster FS Path (Optional)	Insert
Distribution	CDH 5.2 ✓ CDH 5.4.3 with Cloudera Manager HDP 2.2 HDP 2.3 with Ambari 2.1
MR Type	
Master Node Flavor	Extra Large - 4 VCPU, 12288 MB RAM
Worker Count	6
Worker Node Flavor	Medium - 2 VCPU, 6144 MB RAM
YARN High Availability	□
Pig, Hive, Oozie, Impala & Hue	□
Spark	□

EMC ISILON INSTALLATION AND CONFIGURATION

Install

The physical assembly and configuration of the Isilon cluster was performed without incident using some basic information provided by email from EMC.

Screenshots of EMC Isilon Set Up

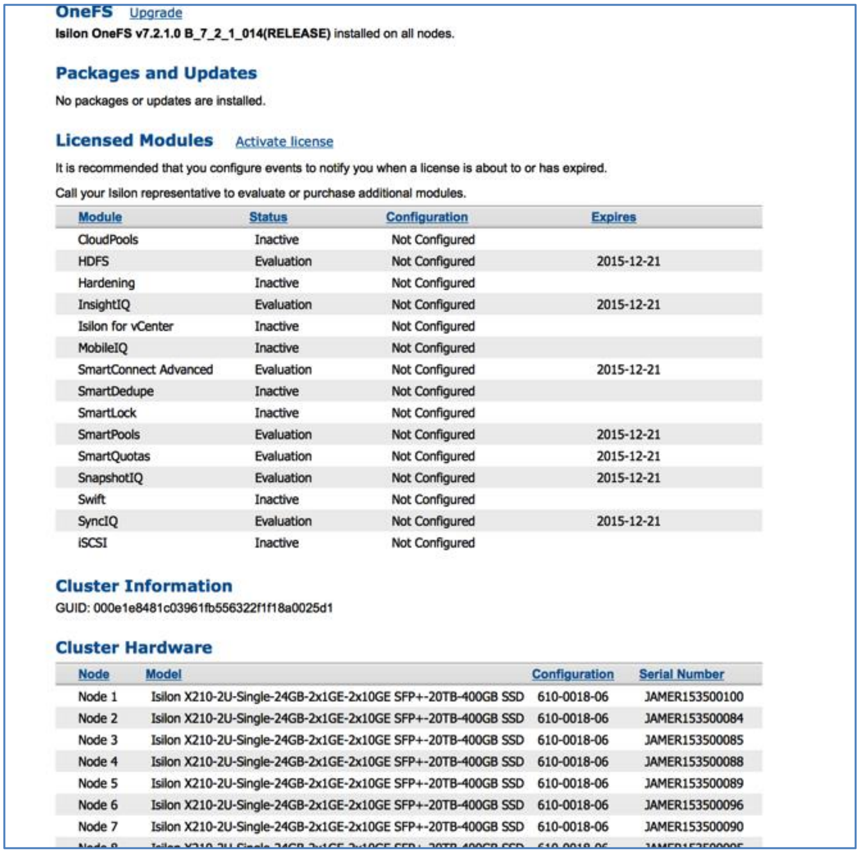


Cluster Name: bd-cluster2 (OneFS Version: 7.2.1.0)

Cluster Overview

Status - bd-cluster2 As of 12:28:54 PDT

Stat	ID	Address	In b/s	Out b/s	Total b/s	HDD Used	HDD Size	%	SSD Used	SSD Size	%
●	1	2 addresses	157 K	5.32 M	5.48 M	18.8 G	18.0 T	< 1	–	–	–
●	2	2 addresses	–	32 b	31 b	19.5 G	18.0 T	< 1	–	–	–
●	3	2 addresses	214 b	312 b	526 b	15.1 G	18.0 T	< 1	–	–	–
●	4	2 addresses	–	240 K	240 K	19.7 G	18.0 T	< 1	–	–	–
●	5	2 addresses	–	40 b	39 b	15.4 G	18.0 T	< 1	–	–	–
●	6	2 addresses	–	36 b	35 b	19.4 G	18.0 T	< 1	–	–	–
●	7	2 addresses	428 b	42 b	470 b	16.1 G	18.0 T	< 1	–	–	–
●	8	2 addresses	–	30 b	29 b	17.3 G	18.0 T	< 1	–	–	–
Totals	8		158 K	5.56 M	5.72 M	141 G	144 T	< 1	–	–	–



OneFS Upgrade

Isilon OneFS v7.2.1.0 B_7_2_1_014(RELEASE) installed on all nodes.

Packages and Updates

No packages or updates are installed.

Licensed Modules Activate license

It is recommended that you configure events to notify you when a license is about to or has expired.

Call your Isilon representative to evaluate or purchase additional modules.

Module	Status	Configuration	Expires
CloudPools	Inactive	Not Configured	
HDFS	Evaluation	Not Configured	2015-12-21
Hardening	Inactive	Not Configured	
InsightIQ	Evaluation	Not Configured	2015-12-21
Isilon for vCenter	Inactive	Not Configured	
MobileIQ	Inactive	Not Configured	
SmartConnect Advanced	Evaluation	Not Configured	2015-12-21
SmartDedupe	Inactive	Not Configured	
SmartLock	Inactive	Not Configured	
SmartPools	Evaluation	Not Configured	2015-12-21
SmartQuotas	Evaluation	Not Configured	2015-12-21
SnapshotIQ	Evaluation	Not Configured	2015-12-21
Swift	Inactive	Not Configured	
SyncIQ	Evaluation	Not Configured	2015-12-21
iSCSI	Inactive	Not Configured	

Cluster Information

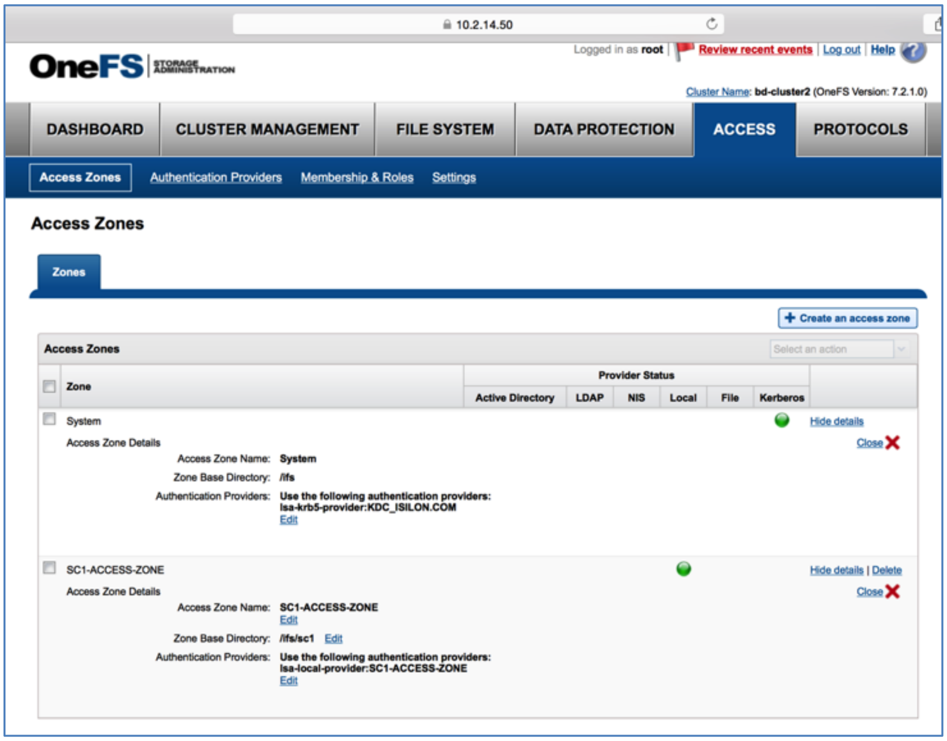
GUID: 000e1e8481c03961fb556322f1f18a0025d1

Cluster Hardware

Node	Model	Configuration	Serial Number
Node 1	Isilon X210-2U-Single-24GB-2x1GE-2x10GE SFP+-20TB-400GB SSD	610-0018-06	JAMER153500100
Node 2	Isilon X210-2U-Single-24GB-2x1GE-2x10GE SFP+-20TB-400GB SSD	610-0018-06	JAMER153500084
Node 3	Isilon X210-2U-Single-24GB-2x1GE-2x10GE SFP+-20TB-400GB SSD	610-0018-06	JAMER153500085
Node 4	Isilon X210-2U-Single-24GB-2x1GE-2x10GE SFP+-20TB-400GB SSD	610-0018-06	JAMER153500088
Node 5	Isilon X210-2U-Single-24GB-2x1GE-2x10GE SFP+-20TB-400GB SSD	610-0018-06	JAMER153500089
Node 6	Isilon X210-2U-Single-24GB-2x1GE-2x10GE SFP+-20TB-400GB SSD	610-0018-06	JAMER153500096
Node 7	Isilon X210-2U-Single-24GB-2x1GE-2x10GE SFP+-20TB-400GB SSD	610-0018-06	JAMER153500090

HDFS with Kerberos Authorization

The creation of the necessary Kerberos Authentication Provider, SPNs, and proxyusers on the Isilon was performed without incident after following the instructions in the OneFS-7.2.0-CLI-Administration-Guide and the Configuring Kerberos with HWX and Isilon white paper.



Access Zones

□ Zone	Provider Status						
	Active Directory	LDAP	NIS	Local	File	Kerberos	
□ System						●	Hide details
□ SC1-ACCESS-ZONE				●			Hide details | Delete

System — Access Zone Details

Access Zone Name: System

Zone Base Directory: /ifs

Authentication Providers: Use the following authentication providers:
lsa-krb5-provider:KDC_ISILON.COM
Edit

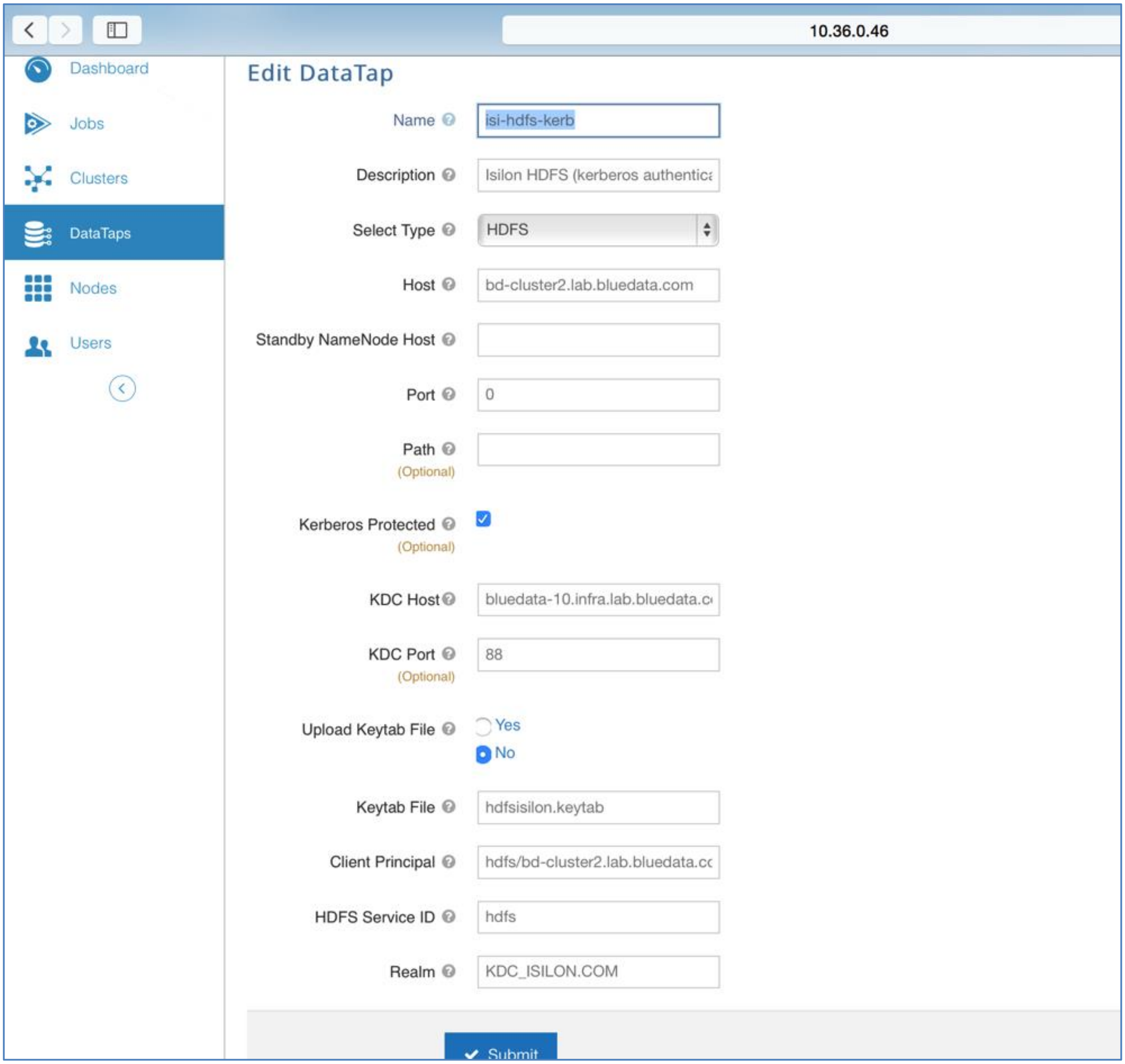
SC1-ACCESS-ZONE — Access Zone Details

Access Zone Name: SC1-ACCESS-ZONE
Edit

Zone Base Directory: /ifs/sc1 Edit

Authentication Providers: Use the following authentication providers:
lsa-local-provider:SC1-ACCESS-ZONE
Edit

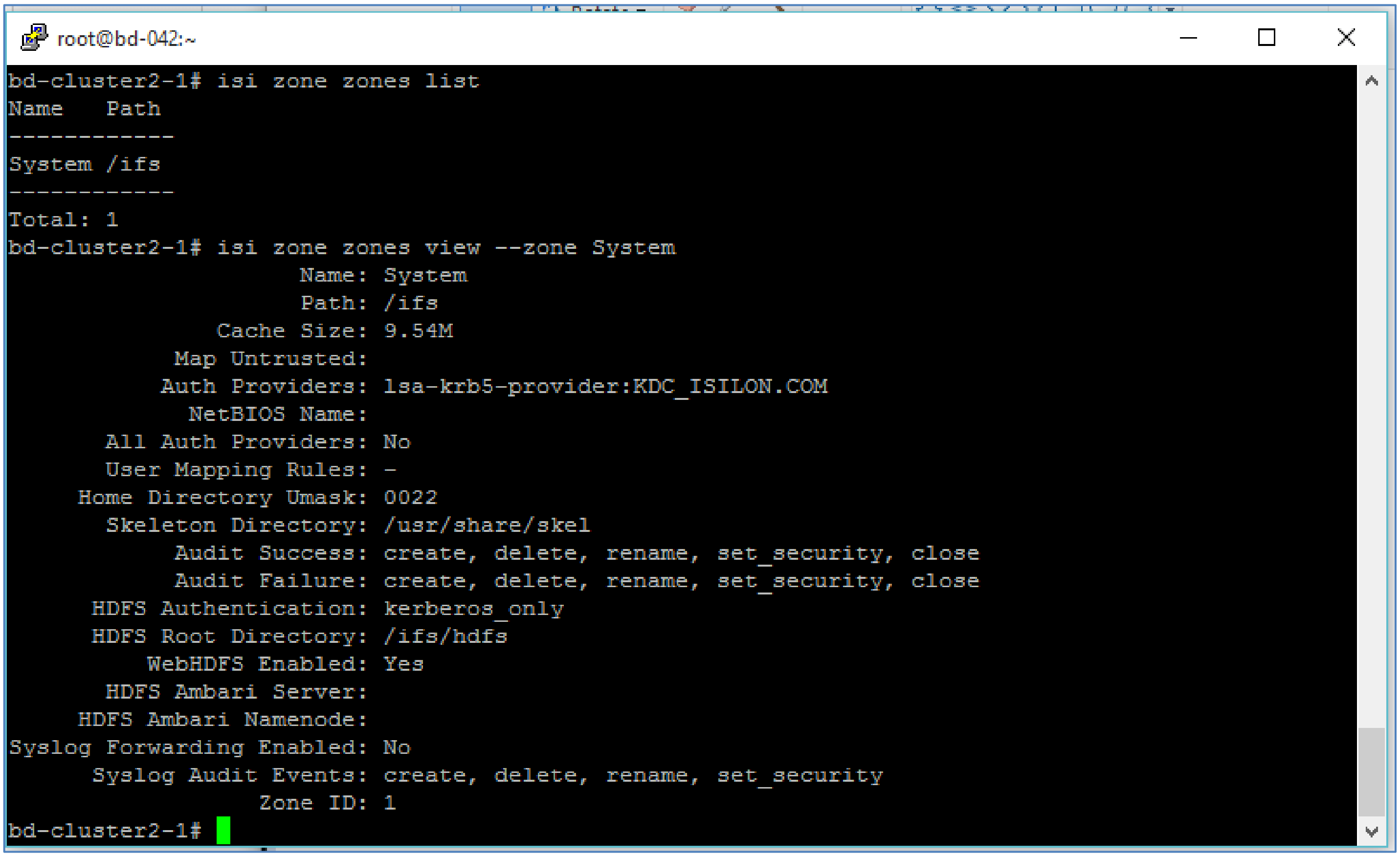
Screenshot of EPIC DataTap Configuration Screen (to Kerberized Isilon HDFS)



Edit DataTap

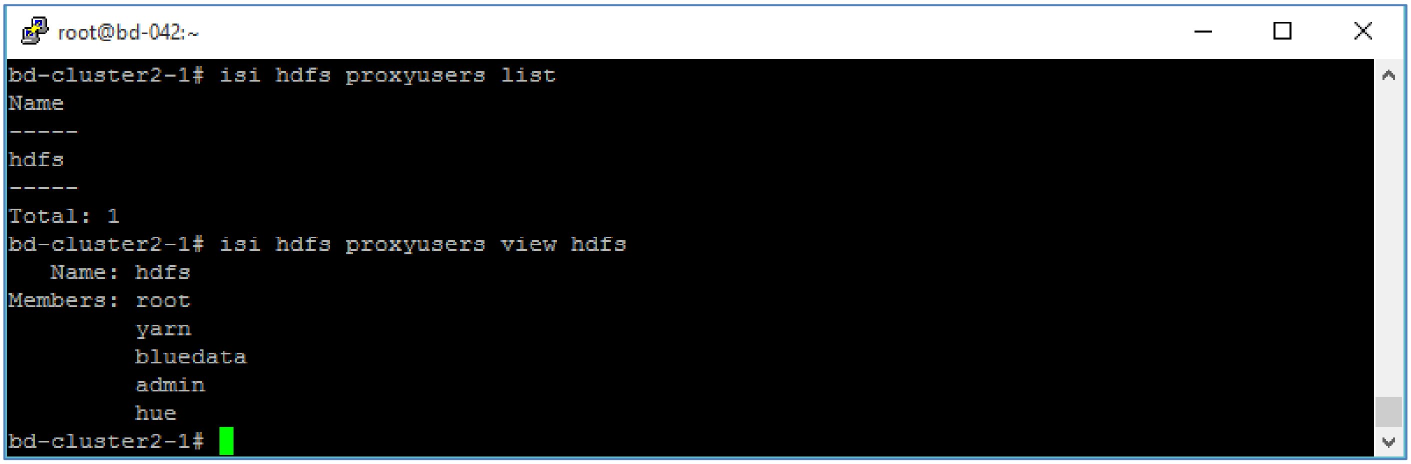
Name	isi-hdfs-kerb
Description	Isilon HDFS (kerberos authentica
Select Type	HDFS
Host	bd-cluster2.lab.bluedata.com
Standby NameNode Host	
Port	0
Path (Optional)	
Kerberos Protected (Optional)	☑
KDC Host	bluedata-10.infra.lab.bluedata.c
KDC Port (Optional)	88
Upload Keytab File	○ Yes ● No
Keytab File	hdfsisilon.keytab
Client Principal	hdfs/bd-cluster2.lab.bluedata.cc
HDFS Service ID	hdfs
Realm	KDC_ISILON.COM

Isilon HDFS Access Zone configuration



```
bd-cluster2-1# isi zone zones list
Name   Path
------------
System /ifs
------------
Total: 1
bd-cluster2-1# isi zone zones view --zone System
                     Name: System
                     Path: /ifs
               Cache Size: 9.54M
            Map Untrusted:
           Auth Providers: lsa-krb5-provider:KDC_ISILON.COM
             NetBIOS Name:
       All Auth Providers: No
       User Mapping Rules: -
     Home Directory Umask: 0022
       Skeleton Directory: /usr/share/skel
            Audit Success: create, delete, rename, set_security, close
            Audit Failure: create, delete, rename, set_security, close
      HDFS Authentication: kerberos_only
      HDFS Root Directory: /ifs/hdfs
          WebHDFS Enabled: Yes
       HDFS Ambari Server:
     HDFS Ambari Namenode:
Syslog Forwarding Enabled: No
      Syslog Audit Events: create, delete, rename, set_security
                  Zone ID: 1
bd-cluster2-1#
```

Isilon HDFS Proxyusers Configuration

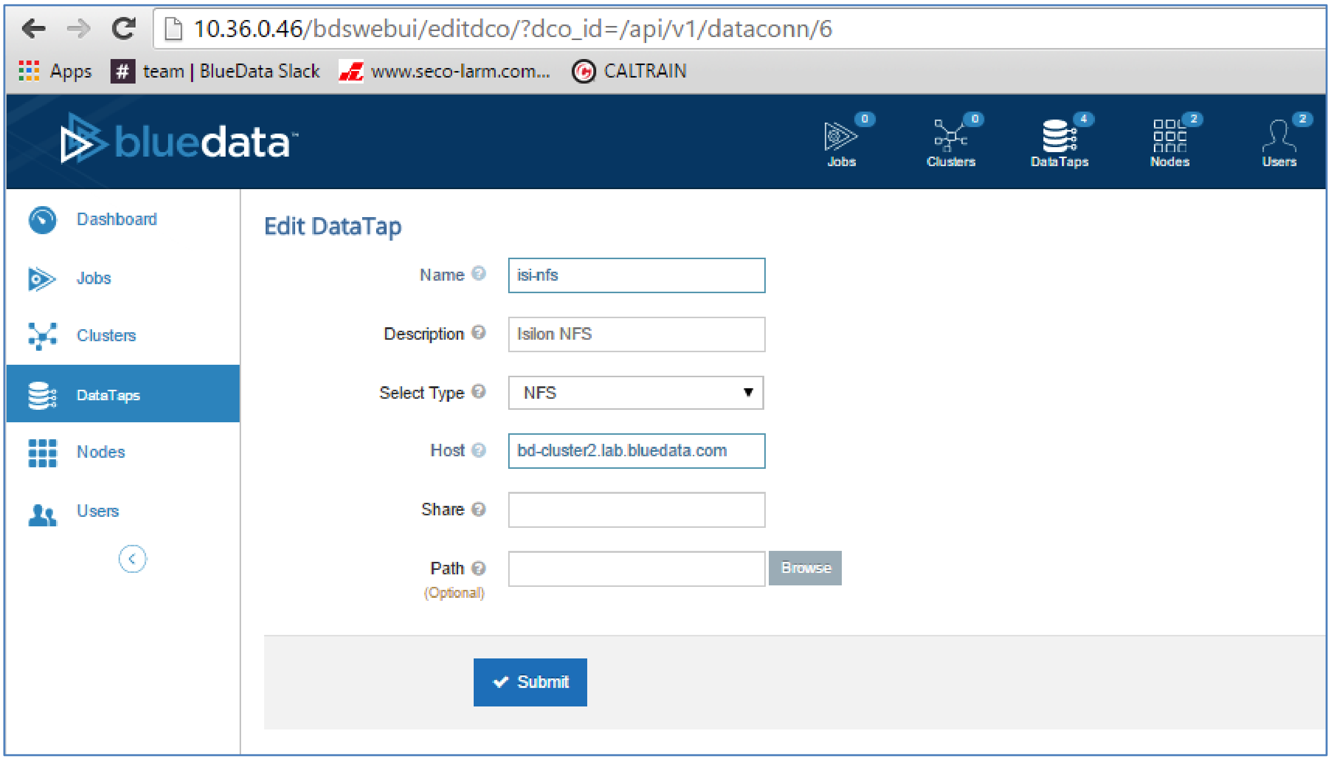


```
bd-cluster2-1# isi hdfs proxyusers list
Name
-----
hdfs
-----
Total: 1
bd-cluster2-1# isi hdfs proxyusers view hdfs
   Name: hdfs
Members: root
         yarn
         bluedata
         admin
         hue
bd-cluster2-1#
```

NFS with Simple Authorization

The creation of an NFS export and associating it with the System Access Zone was performed without incident. The only issue encountered during configuration was that the default setting of the NFS export in Isilon which mapped the “root” user to “nobody” had to be changed in order to permit the “root” user specified by the DataTap to write to the NFS share.

Screenshot of EPIC DataTap Screen (NFS DataTap)



10.36.0.46/bdswebui/editdco/?dco_id=/api/v1/dataconn/6

Edit DataTap

Name: isi-nfs

Description: Isilon NFS

Select Type: NFS

Host: bd-cluster2.lab.bluedata.com

Share:

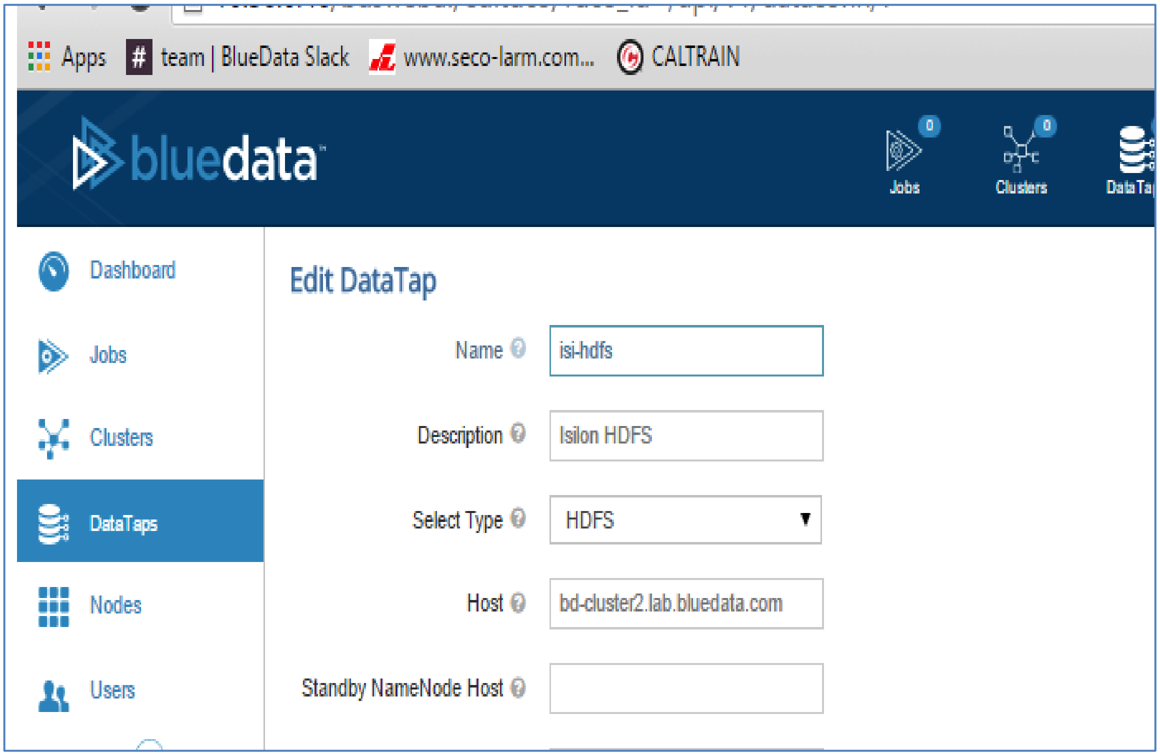
Path (Optional):

HDFS with Simple Authorization

The creation of the Local Authentication Provider with “bluedata”, “hdfs”, and “root” members and configuring it with the System Access Zone was performed without incident.

During basic connectivity testing between DataTap and Isilon using the HDFS protocol, an issue with the DataTap issuing of commands to the DataNode was uncovered. Commands issued by DataTap to the DataNode were returned with a “Lease expired” error. The root issue was that for the Isilon HDFS implementation the contents of the “clientname” fields had to be kept consistent across NameNode and DataNode request packets. This restriction was not observed when running DataTap against a native Apache HDFS instance. The EPIC DataTap code was changed to maintain this clientname consistency.

Screenshot of EPIC DataTap Screen (HDFS Non-Kerberized DataTap)



Edit DataTap

Name	isi-hdfs
Description	Isilon HDFS
Select Type	HDFS ▼
Host	bd-cluster2.lab.bluedata.com
Standby NameNode Host	

Installation of BlueData EPIC was performed using the standard installation procedure. There were no Isilon specific dependencies.

SUPPORTING INFORMATION AND CONSIDERATIONS

The Table below contains pertinent information relating to supporting the testing environment.

Table 7 List of supporting information and considerations relevant to the validation process

DESCRIPTION	DETAIL
Isilon Kerberos Setup	
BlueData Installation Guide	This is an EMC-internal application for automating the execution of benchmarks and collecting performance metrics.

VALIDATION TESTS

COMPATIBILITY TESTING

TEST # 1: Connect to Kerberized Isilon HDFS via BlueData DataTap object.

a. Test Goal: Ability to connect to remote Kerberized Isilon HDFS

b. Pass Criteria:

   i. Browse the directories and files in the Kerberized HDFS Access Zone with trusted user

   ii. Permissions denied message if un-authorized users attempt access to Kerberized Isilon HDFS

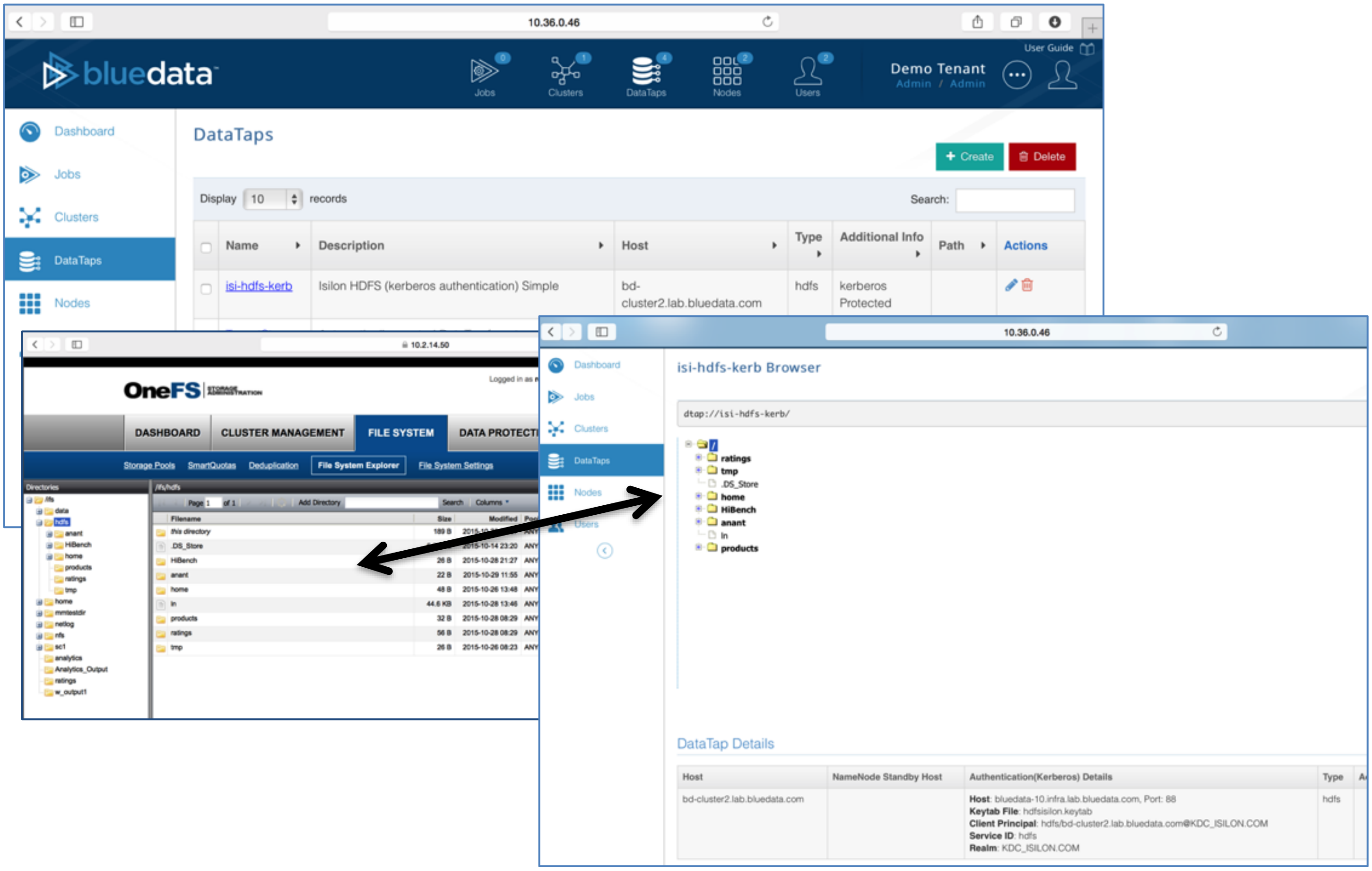
c. Test Preparation:

1. Set up standalone KDC
2. Configure HDFS Access Zone with KDC
3. Generate keytab file for a specific principal (hdfs)

d. Test Plan:

1. Login to BlueData tenant
2. Create new DataTap of type = HDFS
3. Specify Kerberos credentials including keytab file for the relevant user (see screenshot above in installation and configuration section)
4. Click submit
5. Click on the DataTap name to browse the data in Isilon – should match all the directories/folders as seen in Isilon file system browser

e. Result: PASS (see screenshots below)



DataTaps

Name	Description	Host	Type	Additional Info	Path	Actions
isi-hdfs-kerb	Isilon HDFS (kerberos authentication) Simple	bd-cluster2.lab.bluedata.com	hdfs	kerberos Protected		

Filename	Size	Modified	Per
this directory	189 B	2015-	
.DS_Store		2015-10-14 23:20	ANY
HiBench	26 B	2015-10-28 21:27	ANY
anant	22 B	2015-10-29 11:55	ANY
home	48 B	2015-10-26 13:48	ANY
In	44.6 KB	2015-10-28 13:46	ANY
products	32 B	2015-10-28 08:29	ANY
ratings	56 B	2015-10-28 08:29	ANY
tmp	26 B	2015-10-26 08:23	ANY

isi-hdfs-kerb Browser

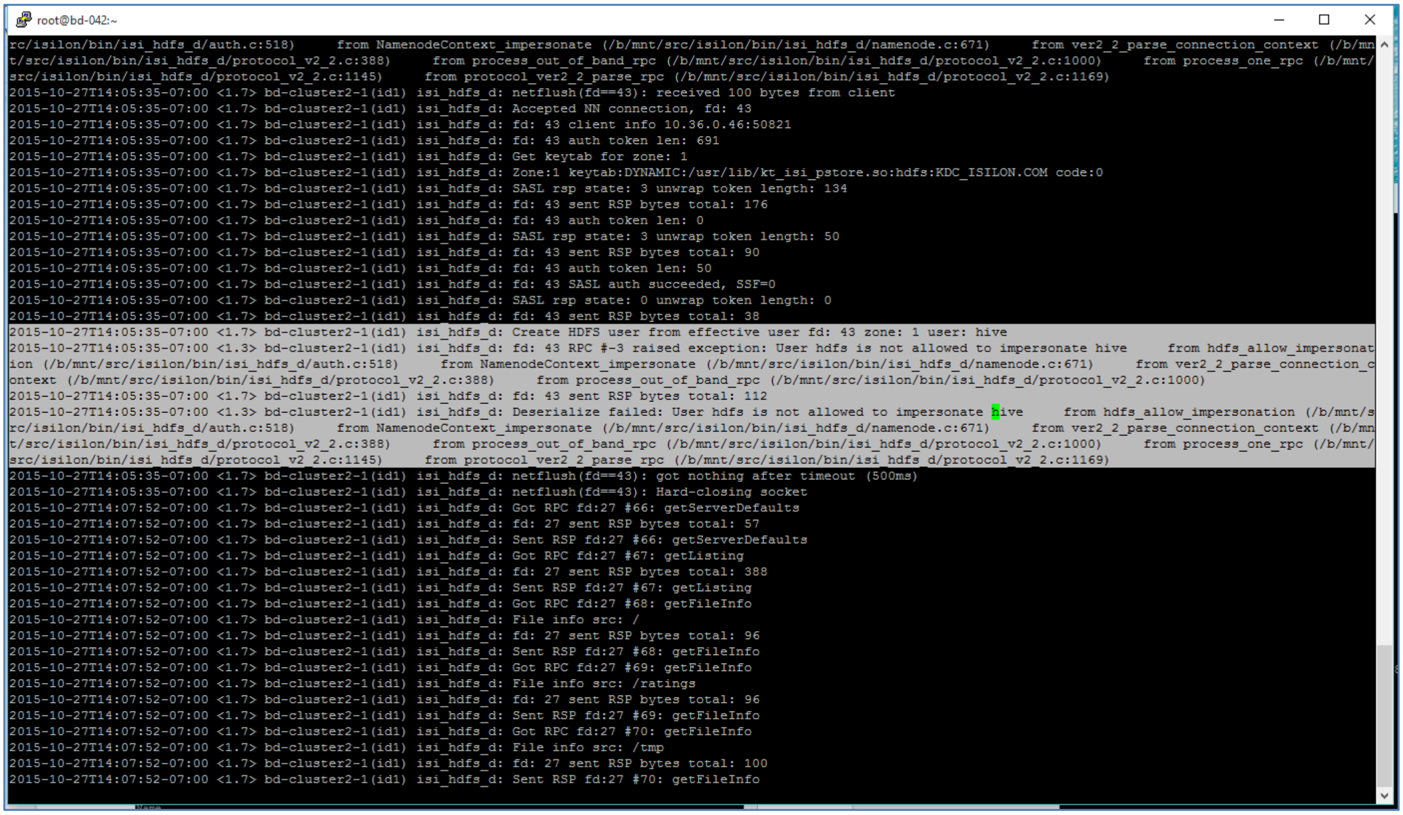
dtap://isi-hdfs-kerb/

```
/
  ratings
  tmp
  .DS_Store
  home
  HiBench
  anant
  In
  products
```

DataTap Details

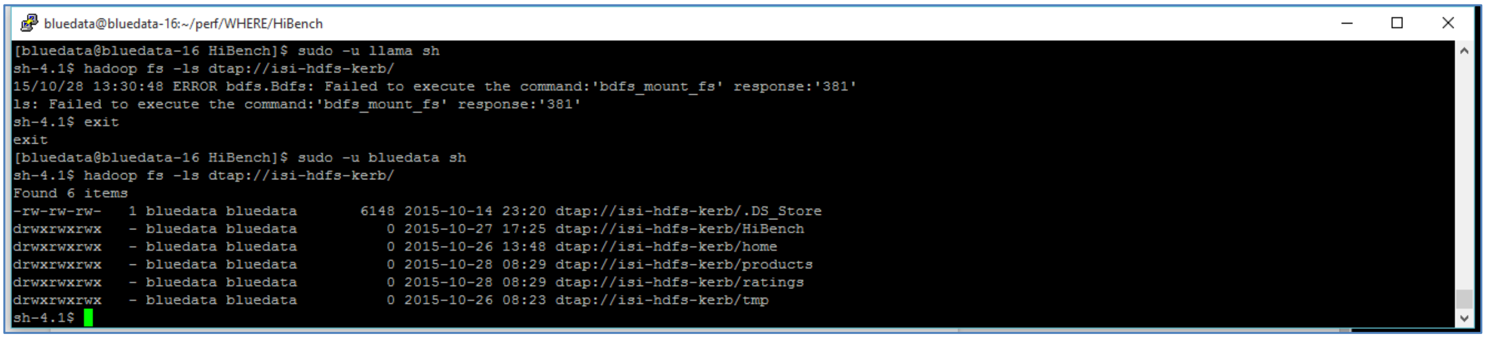
Host	NameNode Standby Host	Authentication(Kerberos) Details	Type	A
bd-cluster2.lab.bluedata.com		**Host**: bluedata-10.infra.lab.bluedata.com, Port: 88 **Keytab File**: hdfsisilon.keytab **Client Principal**: hdfs/bd-cluster2.lab.bluedata.com@KDC_ISILON.COM **Service ID**: hdfs **Realm**: KDC_ISILON.COM	hdfs	

Isilon log showing non-proxyuser being denied DataTap Access



```
rc/isilon/bin/isi_hdfs_d/auth.c:518)     from NamenodeContext_impersonate (/b/mnt/src/isilon/bin/isi_hdfs_d/namenode.c:671)     from ver2_2_parse_connection_context (/b/mn
t/src/isilon/bin/isi_hdfs_d/protocol_v2_2.c:388)     from process_out_of_band_rpc (/b/mnt/src/isilon/bin/isi_hdfs_d/protocol_v2_2.c:1000)     from process_one_rpc (/b/mnt/
src/isilon/bin/isi_hdfs_d/protocol_v2_2.c:1145)     from protocol_ver2_2_parse_rpc (/b/mnt/src/isilon/bin/isi_hdfs_d/protocol_v2_2.c:1169)
2015-10-27T14:05:35-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: netflush(fd==43): received 100 bytes from client
2015-10-27T14:05:35-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: Accepted NN connection, fd: 43
2015-10-27T14:05:35-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: fd: 43 client info 10.36.0.46:50821
2015-10-27T14:05:35-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: fd: 43 auth token len: 691
2015-10-27T14:05:35-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: Get keytab for zone: 1
2015-10-27T14:05:35-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: Zone:1 keytab:DYNAMIC:/usr/lib/kt_isi_pstore.so:hdfs:KDC_ISILON.COM code:0
2015-10-27T14:05:35-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: SASL rsp state: 3 unwrap token length: 134
2015-10-27T14:05:35-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: fd: 43 sent RSP bytes total: 176
2015-10-27T14:05:35-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: fd: 43 auth token len: 0
2015-10-27T14:05:35-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: SASL rsp state: 3 unwrap token length: 50
2015-10-27T14:05:35-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: fd: 43 sent RSP bytes total: 90
2015-10-27T14:05:35-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: fd: 43 auth token len: 50
2015-10-27T14:05:35-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: fd: 43 SASL auth succeeded, SSF=0
2015-10-27T14:05:35-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: SASL rsp state: 0 unwrap token length: 0
2015-10-27T14:05:35-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: fd: 43 sent RSP bytes total: 38
2015-10-27T14:05:35-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: Create HDFS user from effective user fd: 43 zone: 1 user: hive
2015-10-27T14:05:35-07:00 <1.3> bd-cluster2-1(id1) isi_hdfs_d: fd: 43 RPC #-3 raised exception: User hdfs is not allowed to impersonate hive     from hdfs_allow_impersonat
ion (/b/mnt/src/isilon/bin/isi_hdfs_d/auth.c:518)     from NamenodeContext_impersonate (/b/mnt/src/isilon/bin/isi_hdfs_d/namenode.c:671)     from ver2_2_parse_connection_c
ontext (/b/mnt/src/isilon/bin/isi_hdfs_d/protocol_v2_2.c:388)     from process_out_of_band_rpc (/b/mnt/src/isilon/bin/isi_hdfs_d/protocol_v2_2.c:1000)
2015-10-27T14:05:35-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: fd: 43 sent RSP bytes total: 112
2015-10-27T14:05:35-07:00 <1.3> bd-cluster2-1(id1) isi_hdfs_d: Deserialize failed: User hdfs is not allowed to impersonate hive     from hdfs_allow_impersonation (/b/mnt/s
rc/isilon/bin/isi_hdfs_d/auth.c:518)     from NamenodeContext_impersonate (/b/mnt/src/isilon/bin/isi_hdfs_d/namenode.c:671)     from ver2_2_parse_connection_context (/b/mn
t/src/isilon/bin/isi_hdfs_d/protocol_v2_2.c:388)     from process_out_of_band_rpc (/b/mnt/src/isilon/bin/isi_hdfs_d/protocol_v2_2.c:1000)     from process_one_rpc (/b/mnt/
src/isilon/bin/isi_hdfs_d/protocol_v2_2.c:1145)     from protocol_ver2_2_parse_rpc (/b/mnt/src/isilon/bin/isi_hdfs_d/protocol_v2_2.c:1169)
2015-10-27T14:05:35-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: netflush(fd==43): got nothing after timeout (500ms)
2015-10-27T14:05:35-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: netflush(fd==43): Hard-closing socket
2015-10-27T14:07:52-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: Got RPC fd:27 #66: getServerDefaults
2015-10-27T14:07:52-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: fd: 27 sent RSP bytes total: 57
2015-10-27T14:07:52-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: Sent RSP fd:27 #66: getServerDefaults
2015-10-27T14:07:52-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: Got RPC fd:27 #67: getListing
2015-10-27T14:07:52-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: fd: 27 sent RSP bytes total: 388
2015-10-27T14:07:52-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: Sent RSP fd:27 #67: getListing
2015-10-27T14:07:52-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: Got RPC fd:27 #68: getFileInfo
2015-10-27T14:07:52-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: File info src: /
2015-10-27T14:07:52-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: fd: 27 sent RSP bytes total: 96
2015-10-27T14:07:52-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: Sent RSP fd:27 #68: getFileInfo
2015-10-27T14:07:52-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: Got RPC fd:27 #69: getFileInfo
2015-10-27T14:07:52-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: File info src: /ratings
2015-10-27T14:07:52-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: fd: 27 sent RSP bytes total: 96
2015-10-27T14:07:52-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: Sent RSP fd:27 #69: getFileInfo
2015-10-27T14:07:52-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: Got RPC fd:27 #70: getFileInfo
2015-10-27T14:07:52-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: File info src: /tmp
2015-10-27T14:07:52-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: fd: 27 sent RSP bytes total: 100
2015-10-27T14:07:52-07:00 <1.7> bd-cluster2-1(id1) isi_hdfs_d: Sent RSP fd:27 #70: getFileInfo
```

EPIC Running a CDH 5.4.3 virtual cluster returns an error when un-authorized user “llma” attempts to access an Isilon HDFS file system protected by Kerberos authorization via a DataTap. Access to the same HDFS file system via DataTap succeeds when performed by authorized user “bluedata”.



```
[bluedata@bluedata-16 HiBench]$ sudo -u llama sh
sh-4.1$ hadoop fs -ls dtap://isi-hdfs-kerb/
15/10/28 13:30:48 ERROR bdfs.Bdfs: Failed to execute the command:'bdfs_mount_fs' response:'381'
ls: Failed to execute the command:'bdfs_mount_fs' response:'381'
sh-4.1$ exit
exit
[bluedata@bluedata-16 HiBench]$ sudo -u bluedata sh
sh-4.1$ hadoop fs -ls dtap://isi-hdfs-kerb/
Found 6 items
-rw-rw-rw-   1 bluedata bluedata       6148 2015-10-14 23:20 dtap://isi-hdfs-kerb/.DS_Store
drwxrwxrwx   - bluedata bluedata          0 2015-10-27 17:25 dtap://isi-hdfs-kerb/HiBench
drwxrwxrwx   - bluedata bluedata          0 2015-10-26 13:48 dtap://isi-hdfs-kerb/home
drwxrwxrwx   - bluedata bluedata          0 2015-10-28 08:29 dtap://isi-hdfs-kerb/products
drwxrwxrwx   - bluedata bluedata          0 2015-10-28 08:29 dtap://isi-hdfs-kerb/ratings
drwxrwxrwx   - bluedata bluedata          0 2015-10-26 08:23 dtap://isi-hdfs-kerb/tmp
sh-4.1$
```

TEST # 2: Connect to Isilon via BlueData NFS DataTap

a. Test Goal: Ability to connect to remote Isilon via NFS protocol

b. Pass Criteria: Browse the directories and files in the NFS Access Zone

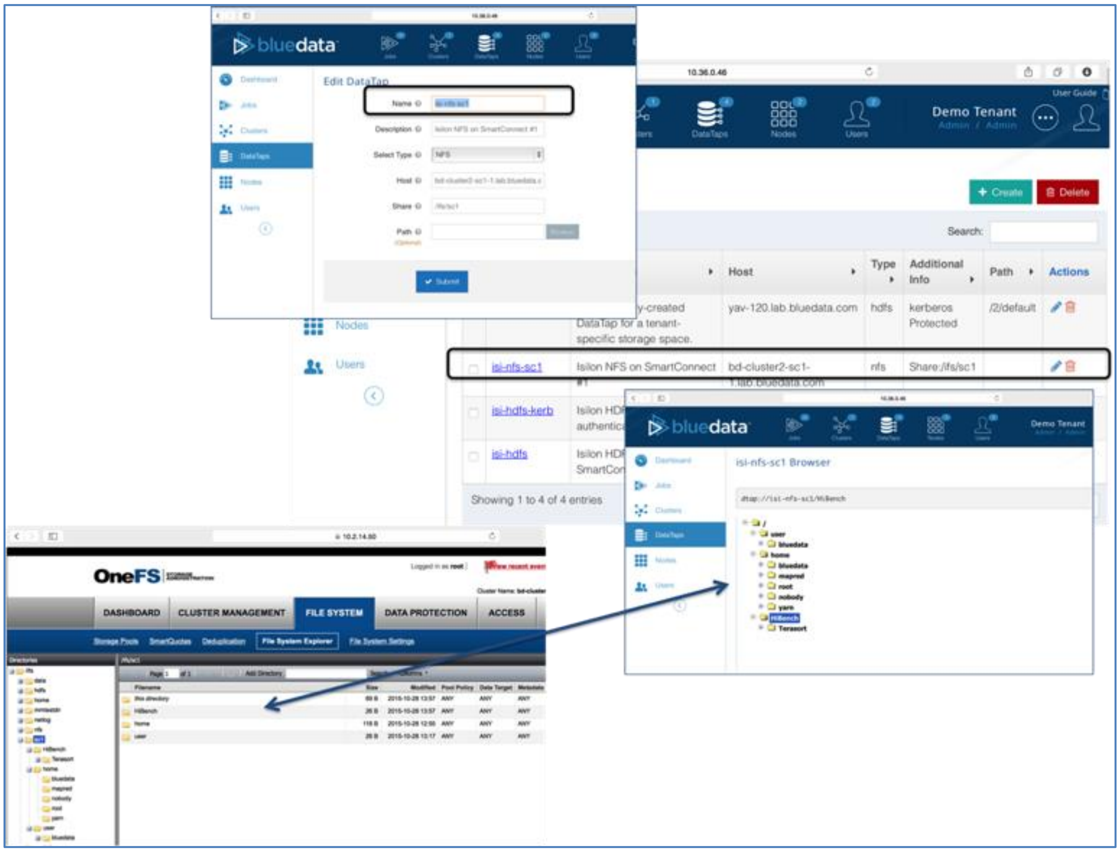
c. Test Preparation:

   1. Configure NFS access zone in Isilon

d. Test Plan:

   1. Login to BlueData tenant
   2. Create new DataTap of type = NFS
   3. Click submit
   4. Click on the DataTap name to browse the data in Isilon – should match all the directories/folders as seen in Isilon file system browser

e. Result: PASS (see screenshots below);



Edit DataTap

Name: isi-nfs-sc1

Description: Isilon NFS on SmartConnect #1

Select Type: NFS

Host: bd-cluster2-sc1-1.lab.bluedata.c

Share: /ifs/sc1

Path (Optional):

		Host	Type	Additional Info	Path	Actions
	…y-created DataTap for a tenant-specific storage space.	yav-120.lab.bluedata.com	hdfs	kerberos Protected	/2/default	
isi-nfs-sc1	Isilon NFS on SmartConnect #1	bd-cluster2-sc1-1.lab.bluedata.com	nfs	Share:/ifs/sc1		
isi-hdfs-kerb	Isilon HD… authentic…					
isi-hdfs	Isilon HD… SmartCon…					

Showing 1 to 4 of 4 entries

isi-nfs-sc1 Browser

dtap://isi-nfs-sc1/HiBench

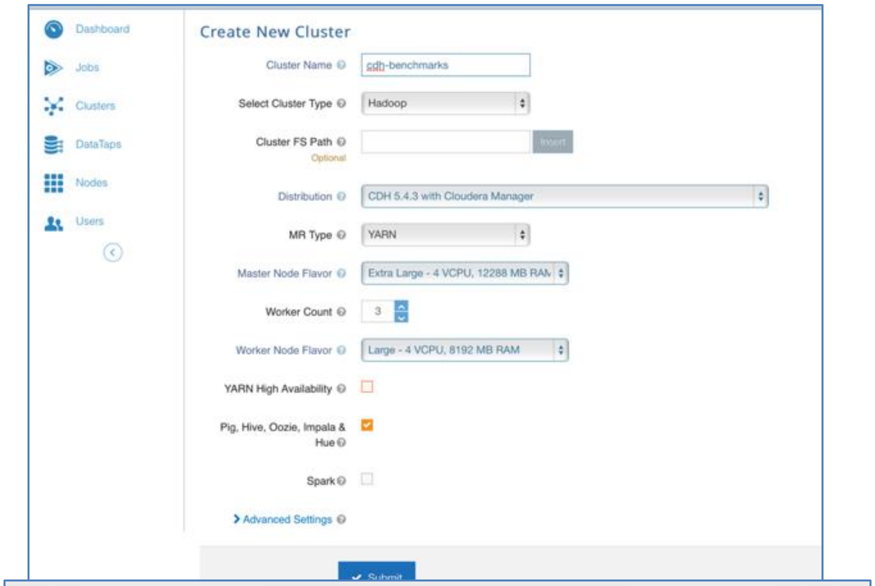
TEST # 3: Browse Kerberized Isilon HDFS DataTap from CDH (Cloudera) Compute Cluster

a. Test Goal: Ability to connect to Kerberized Isilon HDFS DataTap from any Hadoop compute cluster without any specialized configuration (i.e. without any distribution specific integration with Isilon HDFS)

b. Pass Criteria: Login to master node of Hadoop cluster and issue hadoop fs commands against Isilon HDFS DataTap

c. Test Preparation:

1. Create a Cloudera (CDH) virtual cluster using BlueData cluster



Create New Cluster

Cluster Name: cdh-benchmarks

Select Cluster Type: Hadoop

Cluster FS Path (Optional): Insert

Distribution: CDH 5.4.3 with Cloudera Manager

MR Type: YARN

Master Node Flavor: Extra Large - 4 VCPU, 12288 MB RAM

Worker Count: 3

Worker Node Flavor: Large - 4 VCPU, 8192 MB RAM

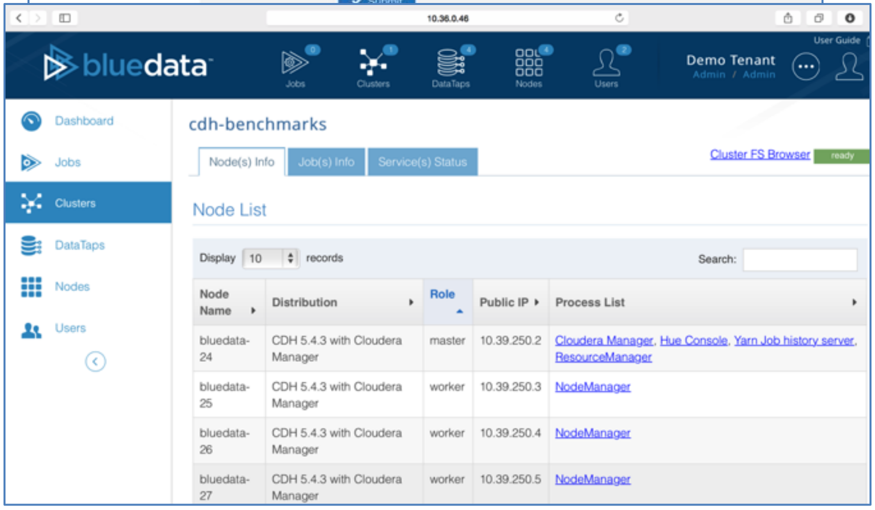
YARN High Availability ☐

Pig, Hive, Oozie, Impala & Hue ☑

Spark ☐

Advanced Settings

Submit



cdh-benchmarks

Node(s) Info | Job(s) Info | Service(s) Status

Cluster FS Browser — ready

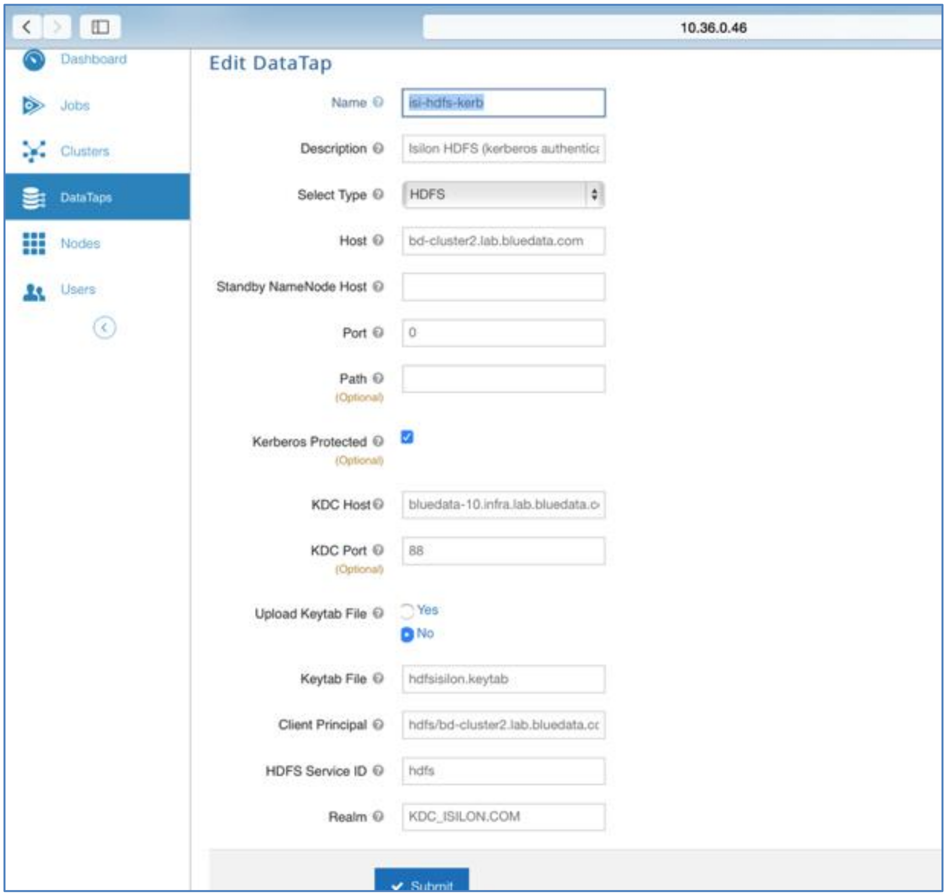
Node List

Display 10 records — Search:

Node Name	Distribution	Role	Public IP	Process List
bluedata-24	CDH 5.4.3 with Cloudera Manager	master	10.39.250.2	Cloudera Manager, Hue Console, Yarn Job history server, ResourceManager
bluedata-25	CDH 5.4.3 with Cloudera Manager	worker	10.39.250.3	NodeManager
bluedata-26	CDH 5.4.3 with Cloudera Manager	worker	10.39.250.4	NodeManager
bluedata-27	CDH 5.4.3 with Cloudera Manager	worker	10.39.250.5	NodeManager

CDH virtual

Utilize the Isilon DataTap (isi-hdfs-kerb)



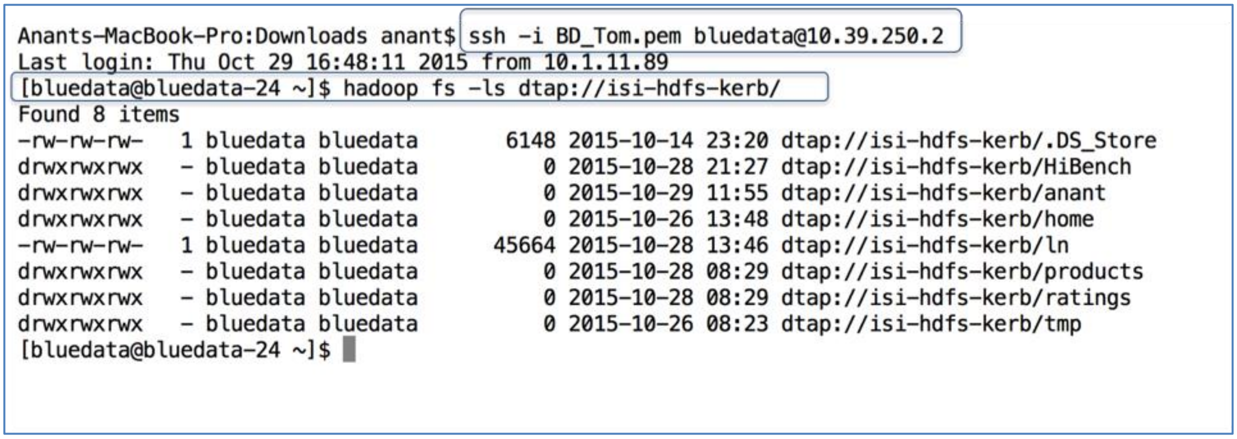
Edit DataTap

Name	isi-hdfs-kerb
Description	Isilon HDFS (kerberos authentic
Select Type	HDFS
Host	bd-cluster2.lab.bluedata.com
Standby NameNode Host	
Port	0
Path (Optional)	
Kerberos Protected (Optional)	☑
KDC Host	bluedata-10.infra.lab.bluedata.c
KDC Port (Optional)	88
Upload Keytab File	○ Yes ● No
Keytab File	hdfsisilon.keytab
Client Principal	hdfs/bd-cluster2.lab.bluedata.cc
HDFS Service ID	hdfs
Realm	KDC_ISILON.COM

d. Test Plan:

1. ssh into master node (10.39.250.2) of virtual Hadoop cluster (see above)
2. Execute command: hadoop fs – ls dtap://isi-hdfs-kerb
3. The directories/folders listed must match Isilon file system browser

e. Result: PASS (see screenshots below);



```
Anants-MacBook-Pro:Downloads anant$ ssh -i BD_Tom.pem bluedata@10.39.250.2
Last login: Thu Oct 29 16:48:11 2015 from 10.1.11.89
[bluedata@bluedata-24 ~]$ hadoop fs -ls dtap://isi-hdfs-kerb/
Found 8 items
-rw-rw-rw-   1 bluedata bluedata       6148 2015-10-14 23:20 dtap://isi-hdfs-kerb/.DS_Store
drwxrwxrwx   - bluedata bluedata          0 2015-10-28 21:27 dtap://isi-hdfs-kerb/HiBench
drwxrwxrwx   - bluedata bluedata          0 2015-10-29 11:55 dtap://isi-hdfs-kerb/anant
drwxrwxrwx   - bluedata bluedata          0 2015-10-26 13:48 dtap://isi-hdfs-kerb/home
-rw-rw-rw-   1 bluedata bluedata      45664 2015-10-28 13:46 dtap://isi-hdfs-kerb/ln
drwxrwxrwx   - bluedata bluedata          0 2015-10-28 08:29 dtap://isi-hdfs-kerb/products
drwxrwxrwx   - bluedata bluedata          0 2015-10-28 08:29 dtap://isi-hdfs-kerb/ratings
drwxrwxrwx   - bluedata bluedata          0 2015-10-26 08:23 dtap://isi-hdfs-kerb/tmp
[bluedata@bluedata-24 ~]$ 
```

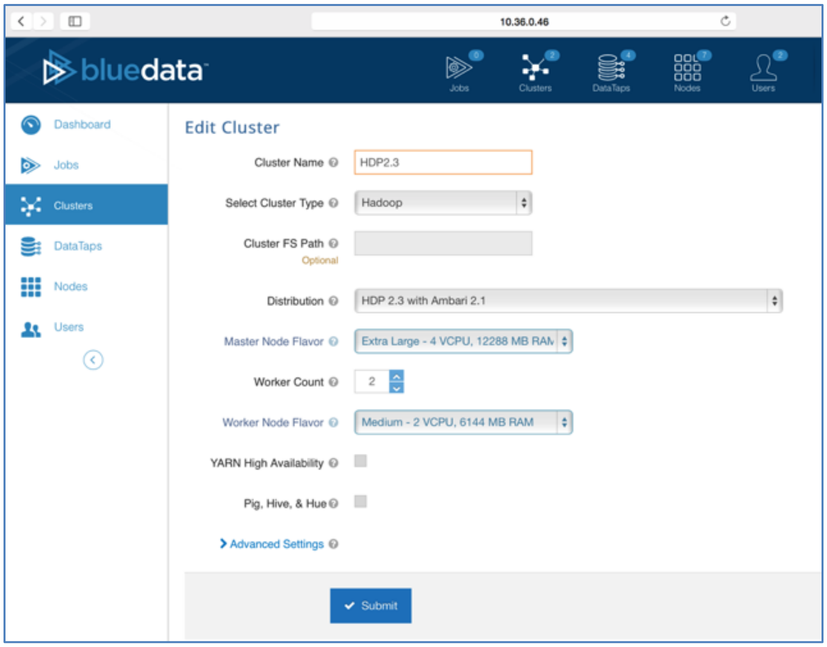
TEST # 4: Browse Kerberized Isilon HDFS DataTap from HDP (Hortonworks) Compute Cluster

f. Test Goal: Ability to connect to Kerberized Isilon HDFS DataTap from any Hadoop compute cluster without any specialized configuration (i.e. without any distribution specific integration with Isilon HDFS such as using Cloudera Manager or Ambari)

g. Pass Criteria: Issue hadoop fs commands against Isilon HDFS via BlueData DataTap URI (dtap://)

h. Test Preparation:

1. Create a Hortonworks (HDP) virtual cluster using BlueData



Edit Cluster

Cluster Name: HDP2.3

Select Cluster Type: Hadoop

Cluster FS Path (Optional):

Distribution: HDP 2.3 with Ambari 2.1

Master Node Flavor: Extra Large - 4 VCPU, 12288 MB RAM

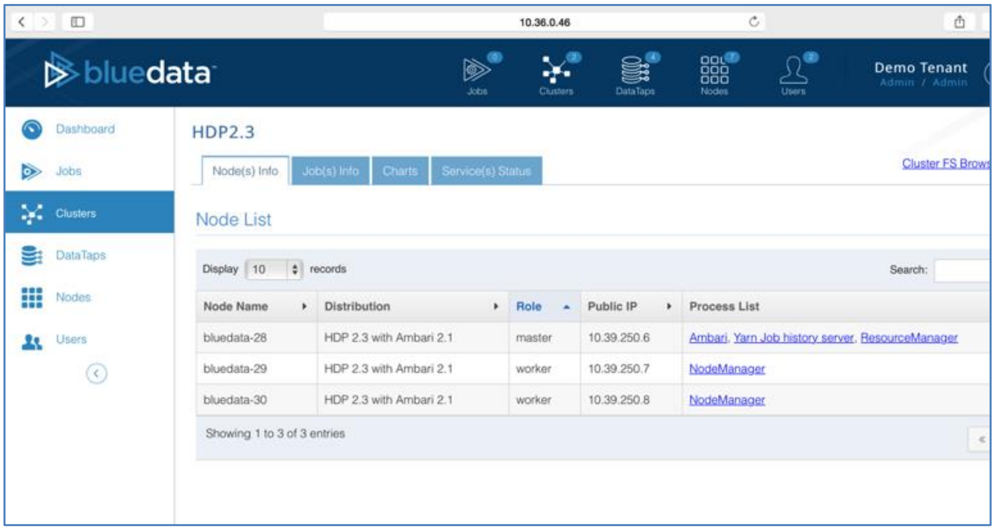
Worker Count: 2

Worker Node Flavor: Medium - 2 VCPU, 6144 MB RAM

YARN High Availability: □

Pig, Hive, & Hue: □

Note: The IP address of the virtual master node of the Hadoop cluster is 10.39.250.6



HDP2.3

Node List

Display 10 records Search:

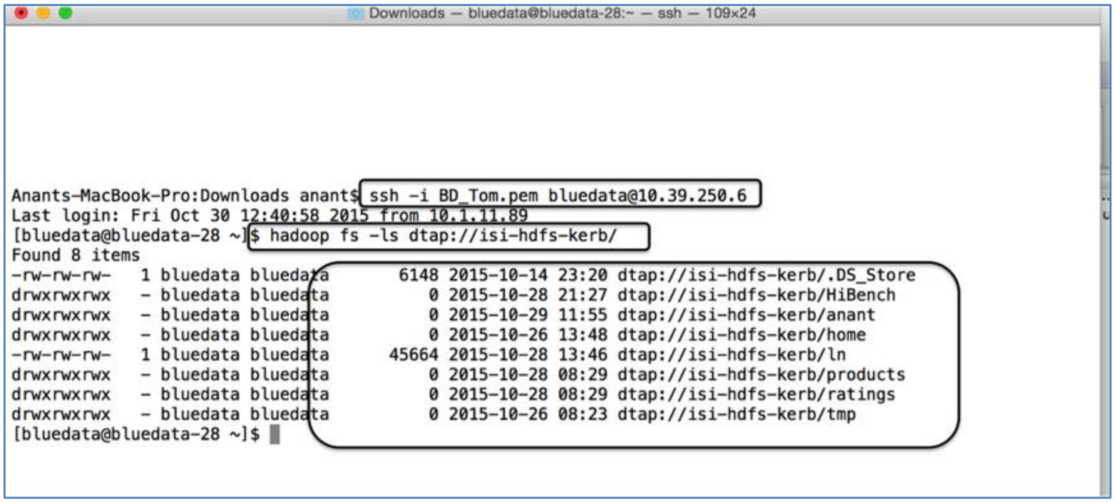
Node Name	Distribution	Role	Public IP	Process List
bluedata-28	HDP 2.3 with Ambari 2.1	master	10.39.250.6	Ambari, Yarn Job history server, ResourceManager
bluedata-29	HDP 2.3 with Ambari 2.1	worker	10.39.250.7	NodeManager
bluedata-30	HDP 2.3 with Ambari 2.1	worker	10.39.250.8	NodeManager

Showing 1 to 3 of 3 entries

i. Test Plan:

1. ssh into master node of virtual cluster
2. Execute command: hadoop fs – ls dtap://isi-hdfs-kerb
3. The directories/folders listed must match Isilon file system browser

j. Result: PASS (see screenshots below);



```
Anants-MacBook-Pro:Downloads anant$ ssh -i BD_Tom.pem bluedata@10.39.250.6
Last login: Fri Oct 30 12:40:58 2015 from 10.1.11.89
[bluedata@bluedata-28 ~]$ hadoop fs -ls dtap://isi-hdfs-kerb/
Found 8 items
-rw-rw-rw-   1 bluedata bluedata       6148 2015-10-14 23:20 dtap://isi-hdfs-kerb/.DS_Store
drwxrwxrwx   - bluedata bluedata          0 2015-10-28 21:27 dtap://isi-hdfs-kerb/HiBench
drwxrwxrwx   - bluedata bluedata          0 2015-10-29 11:55 dtap://isi-hdfs-kerb/anant
drwxrwxrwx   - bluedata bluedata          0 2015-10-26 13:48 dtap://isi-hdfs-kerb/home
-rw-rw-rw-   1 bluedata bluedata      45664 2015-10-28 13:46 dtap://isi-hdfs-kerb/ln
drwxrwxrwx   - bluedata bluedata          0 2015-10-28 08:29 dtap://isi-hdfs-kerb/products
drwxrwxrwx   - bluedata bluedata          0 2015-10-28 08:29 dtap://isi-hdfs-kerb/ratings
drwxrwxrwx   - bluedata bluedata          0 2015-10-26 08:23 dtap://isi-hdfs-kerb/tmp
[bluedata@bluedata-28 ~]$ 
```

TEST # 5: Run a Hive query against data stored in Isilon Kerberized HDFS via Hue console

k. Test Goal: Run Hadoop jobs directly against data in-place on Isilon HDFS

l. Pass Criteria: Create external table definitions against data stored in HDFS and obtain results from query

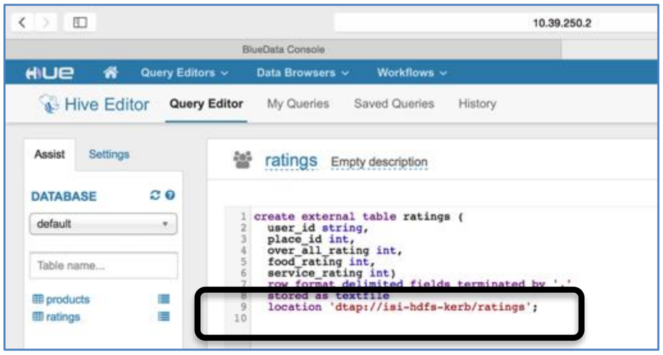
m. Test Preparation:

1. Create a CDH (Cloudera) virtual cluster using BlueData

2. Login to Hue

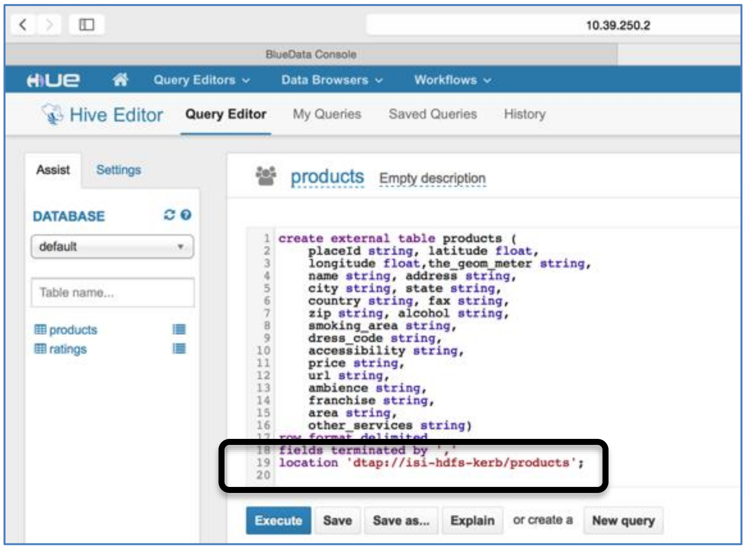
n. Test Plan:

1. Create external table ‘ratings’ against ratings.csv stored on Isilon HDFS under folder = ratings



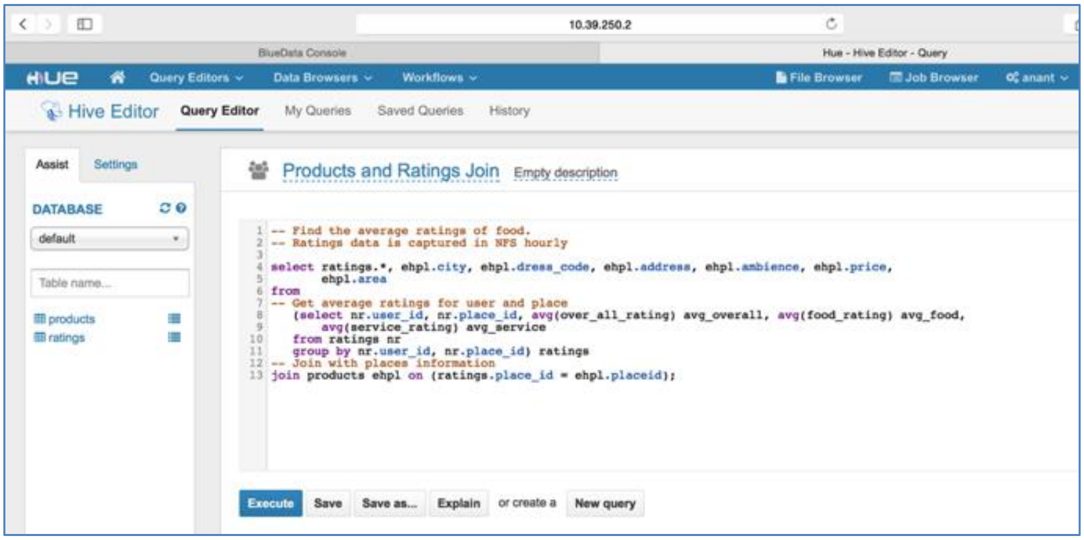
```
create external table ratings (
  user_id string,
  place_id int,
  over_all_rating int,
  food_rating int,
  service_rating int)
  row format delimited fields terminated by ','
  stored as textfile
  location 'dtap://isi-hdfs-kerb/ratings';
```

2. Create external table ‘products’ against geoplaces2.csv stored on Isilon HDFS under folder = products



```
create external table products (
     placeId string, latitude float,
     longitude float,the_geom_meter string,
     name string, address string,
     city string, state string,
     country string, fax string,
     zip string, alcohol string,
     smoking_area string,
     dress_code string,
     accessibility string,
     price string,
     url string,
     ambience string,
     franchise string,
     area string,
     other_services string)
row format delimited
fields terminated by ','
location 'dtap://isi-hdfs-kerb/products';
```

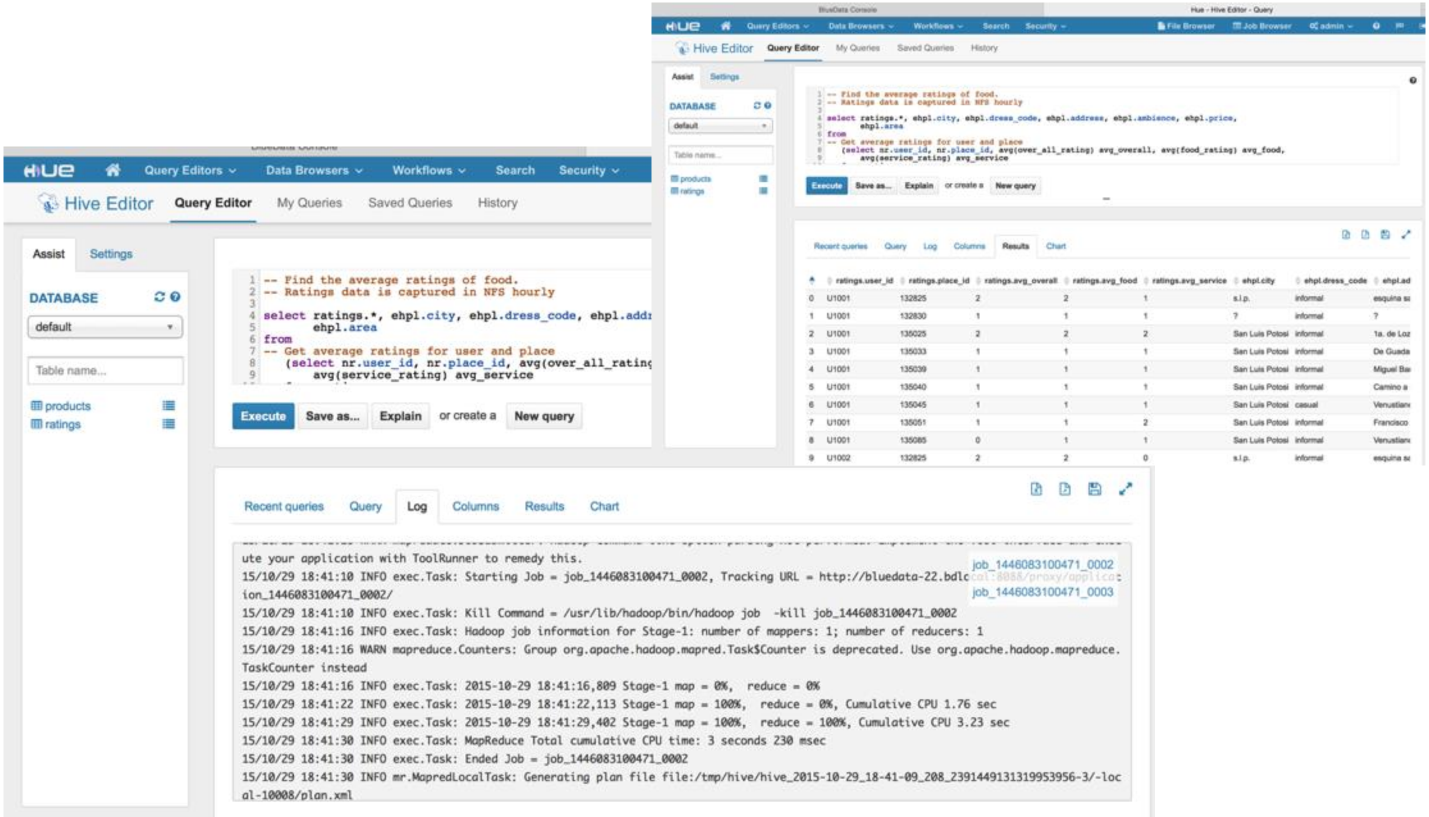
3. Create a join query between ratings and products tables



```
-- Find the average ratings of food.
-- Ratings data is captured in NFS hourly

select ratings.*, ehpl.city, ehpl.dress_code, ehpl.address, ehpl.ambience, ehpl.price,
       ehpl.area
from
-- Get average ratings for user and place
   (select nr.user_id, nr.place_id, avg(over_all_rating) avg_overall, avg(food_rating) avg_food,
      avg(service_rating) avg_service
   from ratings nr
   group by nr.user_id, nr.place_id) ratings
-- Join with places information
join products ehpl on (ratings.place_id = ehpl.placeid);
```

4. Execute the query



	ratings.user_id	ratings.place_id	ratings.avg_overall	ratings.avg_food	ratings.avg_service	ehpl.city	ehpl.dress_code	ehpl.ad
0	U1001	132825	2	2	1	s.l.p.	informal	esquina sa
1	U1001	132830	1	1	1	?	informal	?
2	U1001	135025	2	2	2	San Luis Potosi	informal	1a. de Loz
3	U1001	135033	1	1	1	San Luis Potosi	informal	De Guada
4	U1001	135039	1	1	1	San Luis Potosi	informal	Miguel Ba
5	U1001	135040	1	1	1	San Luis Potosi	informal	Camino a
6	U1001	135045	1	1	1	San Luis Potosi	casual	Venustiano
7	U1001	135051	1	1	2	San Luis Potosi	informal	Francisco
8	U1001	135085	0	1	1	San Luis Potosi	informal	Venustiano
9	U1002	132825	2	2	0	s.l.p.	informal	esquina sa

```
ute your application with ToolRunner to remedy this.
15/10/29 18:41:10 INFO exec.Task: Starting Job = job_1446083100471_0002, Tracking URL = http://bluedata-22.bdlocal:8088/proxy/applicat
ion_1446083100471_0002/
15/10/29 18:41:10 INFO exec.Task: Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1446083100471_0002
15/10/29 18:41:16 INFO exec.Task: Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
15/10/29 18:41:16 WARN mapreduce.Counters: Group org.apache.hadoop.mapred.Task$Counter is deprecated. Use org.apache.hadoop.mapreduce.
TaskCounter instead
15/10/29 18:41:16 INFO exec.Task: 2015-10-29 18:41:16,809 Stage-1 map = 0%,  reduce = 0%
15/10/29 18:41:22 INFO exec.Task: 2015-10-29 18:41:22,113 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.76 sec
15/10/29 18:41:29 INFO exec.Task: 2015-10-29 18:41:29,402 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 3.23 sec
15/10/29 18:41:30 INFO exec.Task: MapReduce Total cumulative CPU time: 3 seconds 230 msec
15/10/29 18:41:30 INFO exec.Task: Ended Job = job_1446083100471_0002
15/10/29 18:41:30 INFO mr.MapredLocalTask: Generating plan file file:/tmp/hive/hive_2015-10-29_18-41-09_208_2391449131319953956-3/-loc
al-10008/plan.xml
```

job_1446083100471_0002
job_1446083100471_0003

o. Result: PASS

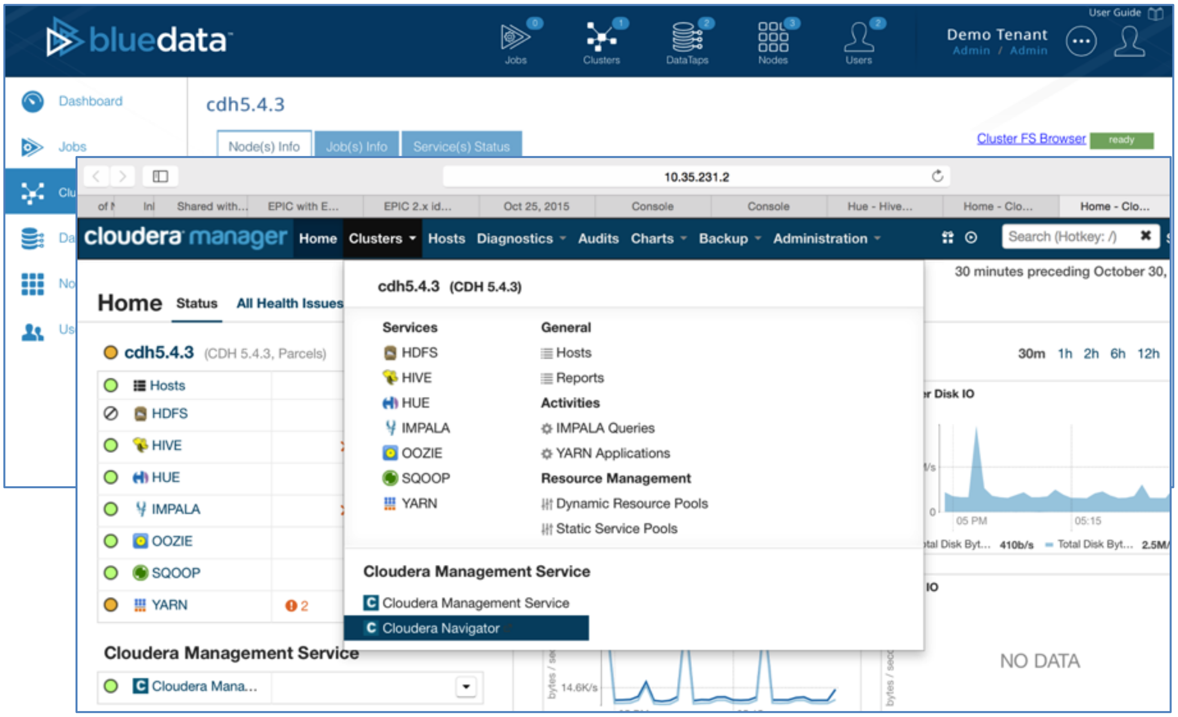
TEST # 5: Integrate with Cloudera Navigator for metadata search and auditing

a. Test Goal: Confirm that Cloudera Navigator can audit and track the metadata (incl. lineage) of compute services.

b. Pass Criteria: Cloudera Navigator search shows **all** the jobs from the compute cluster against Isilon

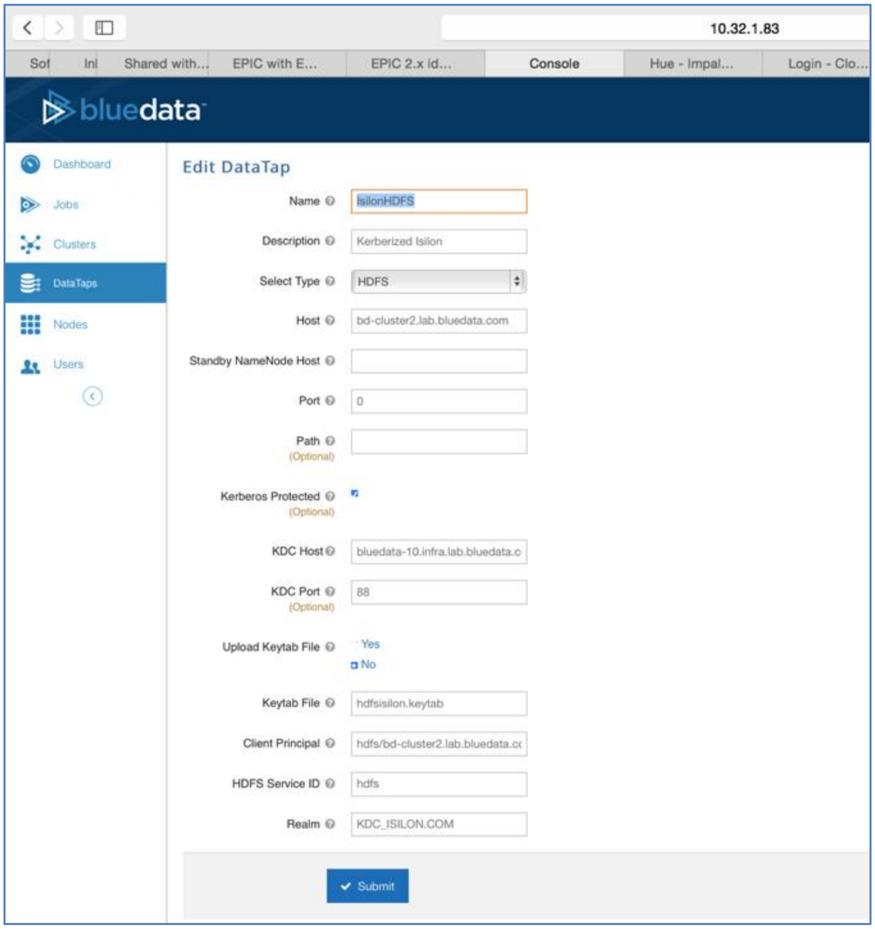
c. Test Preparation:

1. Create CDH 5.4.3 cluster



2. Use the Cloudera Manager (that is instantiated by BlueData) to configure Cloudera Navigator for that above cluster (note that Cloudera manager and hue console are running on 10.35.231.2). Cloudera Navigator services are configured to run on 10.35.231.3

3. Use the IsilonHDFS DataTap - note the configuration below:



Edit DataTap

Name: IsilonHDFS

Description: Kerberized Isilon

Select Type: HDFS

Host: bd-cluster2.lab.bluedata.com

Standby NameNode Host:

Port: 0

Path (Optional):

Kerberos Protected (Optional): ☑

KDC Host: bluedata-10.infra.lab.bluedata.c

KDC Port (Optional): 88

Upload Keytab File: ○ Yes ● No

Keytab File: hdfsisilon.keytab

Client Principal: hdfs/bd-cluster2.lab.bluedata.cc

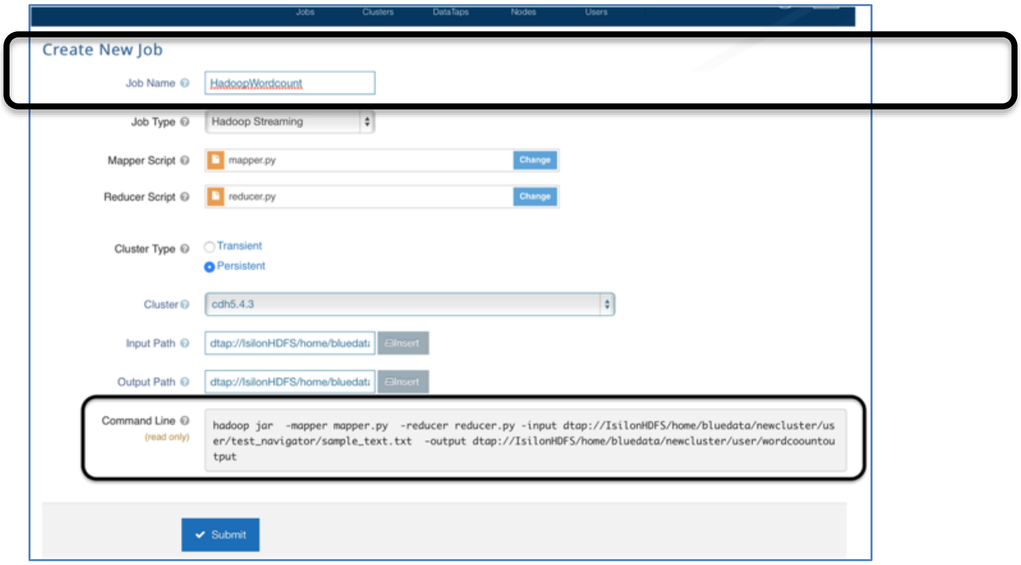
HDFS Service ID: hdfs

Realm: KDC_ISILON.COM

d. Test Plan:

1. Login to BlueData tenant

2. Use the BlueData console to a Hadoop M/R job utilizing the aforementioned cluster against data stored in Isilon (i.e. via dtap://)



Create New Job

Job Name: HadoopWordcount

Job Type: Hadoop Streaming

Mapper Script: mapper.py Change

Reducer Script: reducer.py Change

Cluster Type: ○ Transient ● Persistent

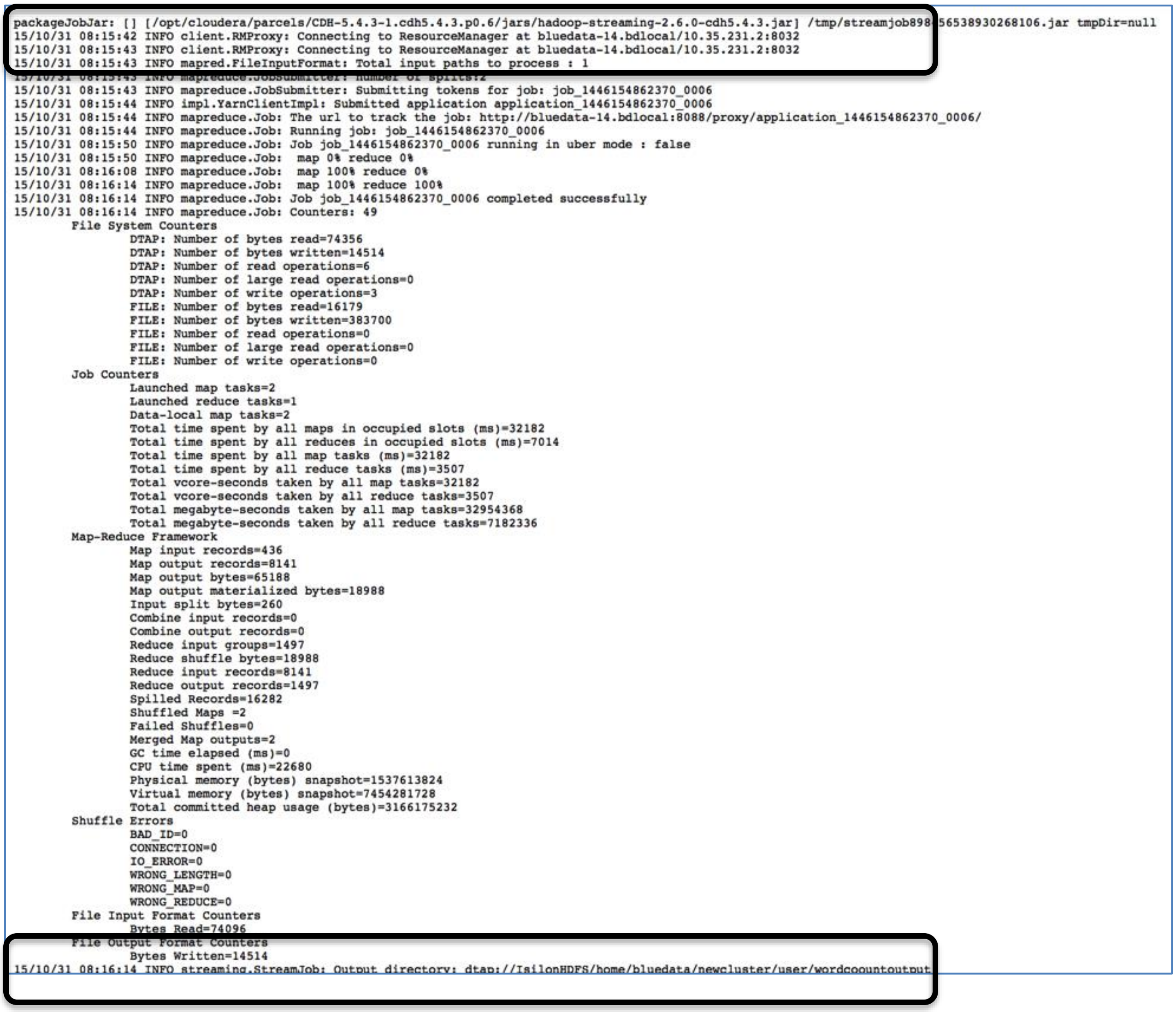
Cluster: cdh5.4.3

Input Path: dtap://IsilonHDFS/home/bluedat: Insert

Output Path: dtap://IsilonHDFS/home/bluedat: Insert

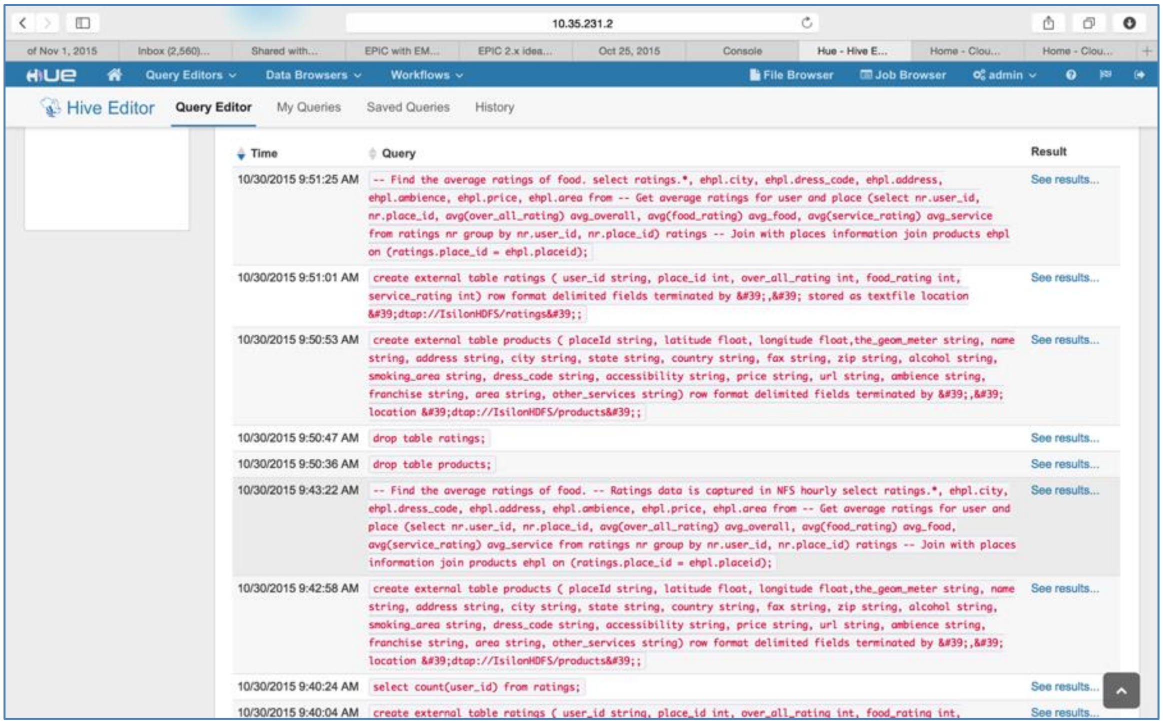
Command Line (read only):

```
hadoop jar  -mapper mapper.py  -reducer reducer.py -input dtap://IsilonHDFS/home/bluedata/newcluster/us
er/test_navigator/sample_text.txt  -output dtap://IsilonHDFS/home/bluedata/newcluster/user/wordcoountou
tput
```

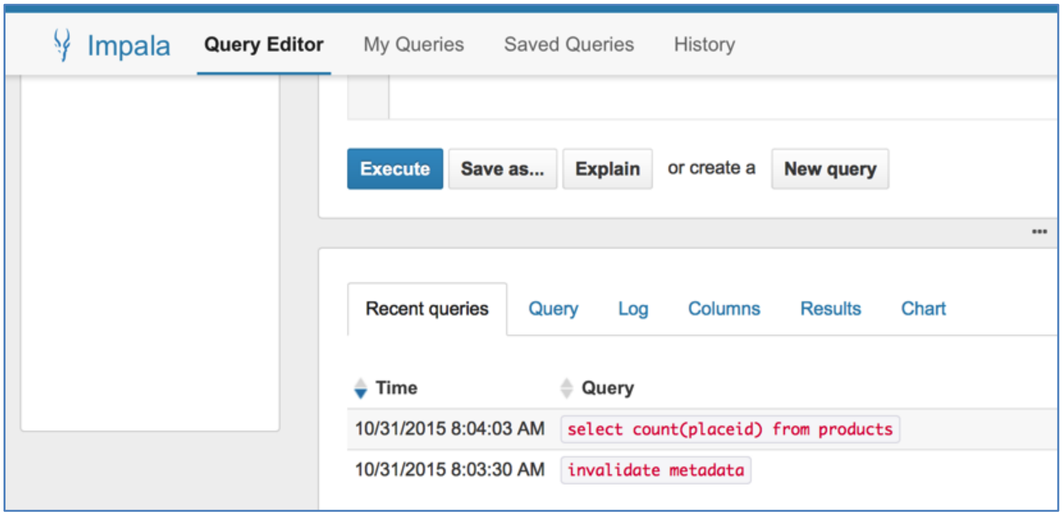


```
packageJobJar: [] [/opt/cloudera/parcels/CDH-5.4.3-1.cdh5.4.3.p0.6/jars/hadoop-streaming-2.6.0-cdh5.4.3.jar] /tmp/streamjob89856538930268106.jar tmpDir=null
15/10/31 08:15:42 INFO client.RMProxy: Connecting to ResourceManager at bluedata-14.bdlocal/10.35.231.2:8032
15/10/31 08:15:43 INFO client.RMProxy: Connecting to ResourceManager at bluedata-14.bdlocal/10.35.231.2:8032
15/10/31 08:15:43 INFO mapred.FileInputFormat: Total input paths to process : 1
15/10/31 08:15:43 INFO mapreduce.JobSubmitter: number of splits:2
15/10/31 08:15:43 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1446154862370_0006
15/10/31 08:15:44 INFO impl.YarnClientImpl: Submitted application application_1446154862370_0006
15/10/31 08:15:44 INFO mapreduce.Job: The url to track the job: http://bluedata-14.bdlocal:8088/proxy/application_1446154862370_0006/
15/10/31 08:15:44 INFO mapreduce.Job: Running job: job_1446154862370_0006
15/10/31 08:15:50 INFO mapreduce.Job: Job job_1446154862370_0006 running in uber mode : false
15/10/31 08:15:50 INFO mapreduce.Job:  map 0% reduce 0%
15/10/31 08:16:08 INFO mapreduce.Job:  map 100% reduce 0%
15/10/31 08:16:14 INFO mapreduce.Job:  map 100% reduce 100%
15/10/31 08:16:14 INFO mapreduce.Job: Job job_1446154862370_0006 completed successfully
15/10/31 08:16:14 INFO mapreduce.Job: Counters: 49
        File System Counters
                DTAP: Number of bytes read=74356
                DTAP: Number of bytes written=14514
                DTAP: Number of read operations=6
                DTAP: Number of large read operations=0
                DTAP: Number of write operations=3
                FILE: Number of bytes read=16179
                FILE: Number of bytes written=383700
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
        Job Counters
                Launched map tasks=2
                Launched reduce tasks=1
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=32182
                Total time spent by all reduces in occupied slots (ms)=7014
                Total time spent by all map tasks (ms)=32182
                Total time spent by all reduce tasks (ms)=3507
                Total vcore-seconds taken by all map tasks=32182
                Total vcore-seconds taken by all reduce tasks=3507
                Total megabyte-seconds taken by all map tasks=32954368
                Total megabyte-seconds taken by all reduce tasks=7182336
        Map-Reduce Framework
                Map input records=436
                Map output records=8141
                Map output bytes=65188
                Map output materialized bytes=18988
                Input split bytes=260
                Combine input records=0
                Combine output records=0
                Reduce input groups=1497
                Reduce shuffle bytes=18988
                Reduce input records=8141
                Reduce output records=1497
                Spilled Records=16282
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=0
                CPU time spent (ms)=22680
                Physical memory (bytes) snapshot=1537613824
                Virtual memory (bytes) snapshot=7454281728
                Total committed heap usage (bytes)=3166175232
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=74096
        File Output Format Counters
                Bytes Written=14514
15/10/31 08:16:14 INFO streaming.StreamJob: Output directory: dtap://IsilonHDFS/home/bluedata/newcluster/user/wordcoountoutput
```

3. Use the Hue Console to create external tables & execute queries against data stored in Isilon and run a join query. ***Note the execution time stamps***



Time	Query	Result
10/30/2015 9:51:25 AM	`-- Find the average ratings of food. select ratings.*, ehpl.city, ehpl.dress_code, ehpl.address, ehpl.ambience, ehpl.price, ehpl.area from -- Get average ratings for user and place (select nr.user_id, nr.place_id, avg(over_all_rating) avg_overall, avg(food_rating) avg_food, avg(service_rating) avg_service from ratings nr group by nr.user_id, nr.place_id) ratings -- Join with places information join products ehpl on (ratings.place_id = ehpl.placeid);`	See results...
10/30/2015 9:51:01 AM	`create external table ratings ( user_id string, place_id int, over_all_rating int, food_rating int, service_rating int) row format delimited fields terminated by &#39;,&#39; stored as textfile location &#39;dtap://IsilonHDFS/ratings&#39;;`	See results...
10/30/2015 9:50:53 AM	`create external table products (placeId string, latitude float, longitude float,the_geom_meter string, name string, address string, city string, state string, country string, fax string, zip string, alcohol string, smoking_area string, dress_code string, accessibility string, price string, url string, ambience string, franchise string, area string, other_services string) row format delimited fields terminated by &#39;,&#39; location &#39;dtap://IsilonHDFS/products&#39;;`	See results...
10/30/2015 9:50:47 AM	`drop table ratings;`	See results...
10/30/2015 9:50:36 AM	`drop table products;`	See results...
10/30/2015 9:43:22 AM	`-- Find the average ratings of food. -- Ratings data is captured in NFS hourly select ratings.*, ehpl.city, ehpl.dress_code, ehpl.address, ehpl.ambience, ehpl.price, ehpl.area from -- Get average ratings for user and place (select nr.user_id, nr.place_id, avg(over_all_rating) avg_overall, avg(food_rating) avg_food, avg(service_rating) avg_service from ratings nr group by nr.user_id, nr.place_id) ratings -- Join with places information join products ehpl on (ratings.place_id = ehpl.placeid);`	See results...
10/30/2015 9:42:58 AM	`create external table products (placeId string, latitude float, longitude float,the_geom_meter string, name string, address string, city string, state string, country string, fax string, zip string, alcohol string, smoking_area string, dress_code string, accessibility string, price string, url string, ambience string, franchise string, area string, other_services string) row format delimited fields terminated by &#39;,&#39; location &#39;dtap://IsilonHDFS/products&#39;;`	See results...
10/30/2015 9:40:24 AM	`select count(user_id) from ratings;`	See results...
10/30/2015 9:40:04 AM	`create external table ratings ( user_id string, place_id int, over_all_rating int, food_rating int,`	See results...



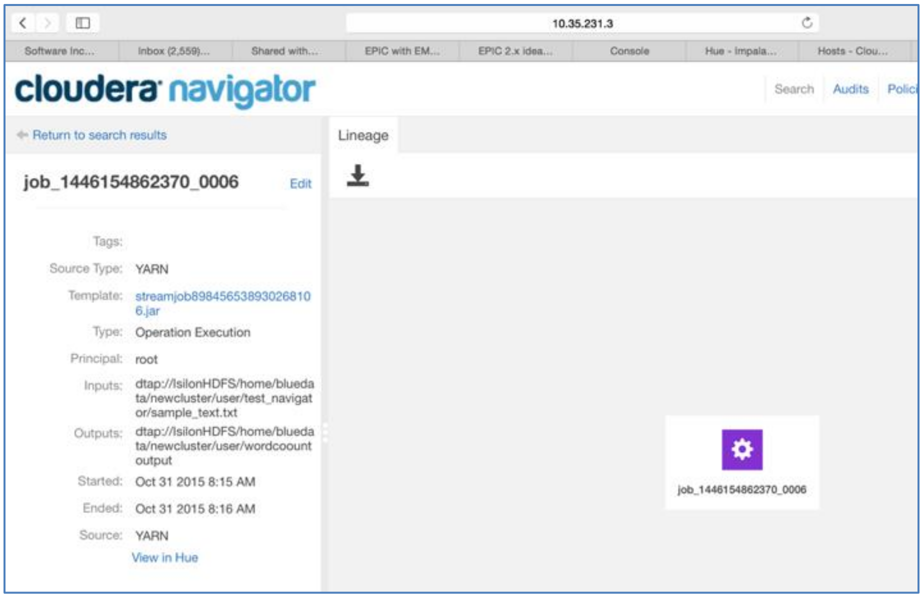
Time	Query
10/31/2015 8:04:03 AM	`select count(placeid) from products`
10/31/2015 8:03:30 AM	`invalidate metadata`

4. Login to Cloudera Navigator – confirm that the metadata search and audit logs show each of the actions as well as sub operations with the appropriate time stamps. Testing was specifically done with HIVE, IMPALA & M/R

e. Result: PASS (see screenshots below);

M/R Job

The time stamp of the job from the test preparation screen is: Oct 31, 2015 at 8:15:43

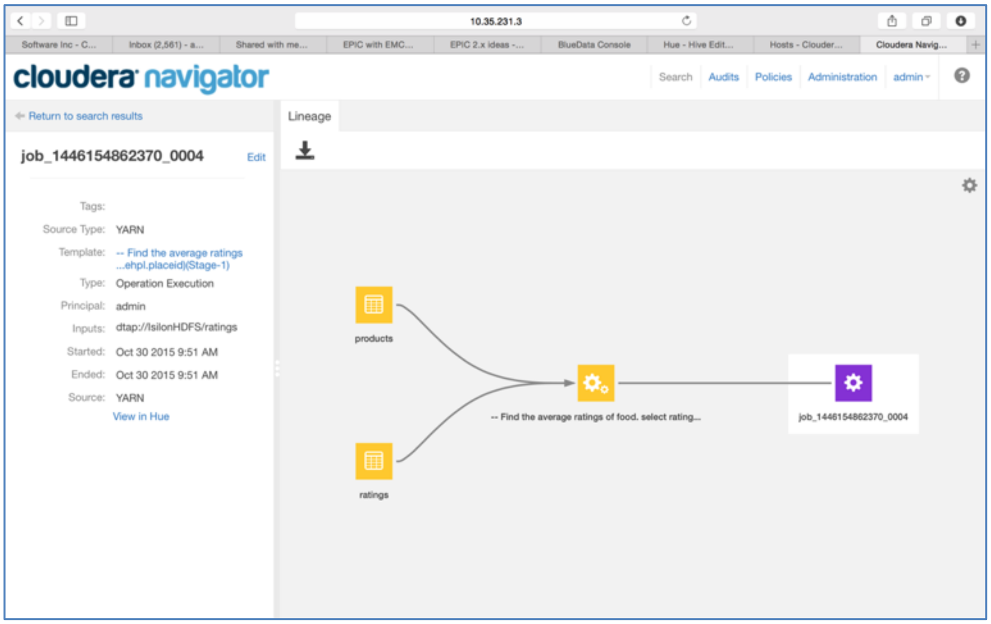


job_1446154862370_0006

Tags:
Source Type: YARN
Template: streamjob898456538930268106.jar
Type: Operation Execution
Principal: root
Inputs: dtap://IsilonHDFS/home/bluedata/newcluster/user/test_navigator/sample_text.txt
Outputs: dtap://IsilonHDFS/home/bluedata/newcluster/user/wordcoount output
Started: Oct 31 2015 8:15 AM
Ended: Oct 31 2015 8:16 AM
Source: YARN

Note that Navigator has the same time stamp and the job input and job output to IsilonHDFS datatap is also shown here.

HIVE QUERY: Note the time stamp from the previous Hue screens of 10/30/2015 from 9.51am

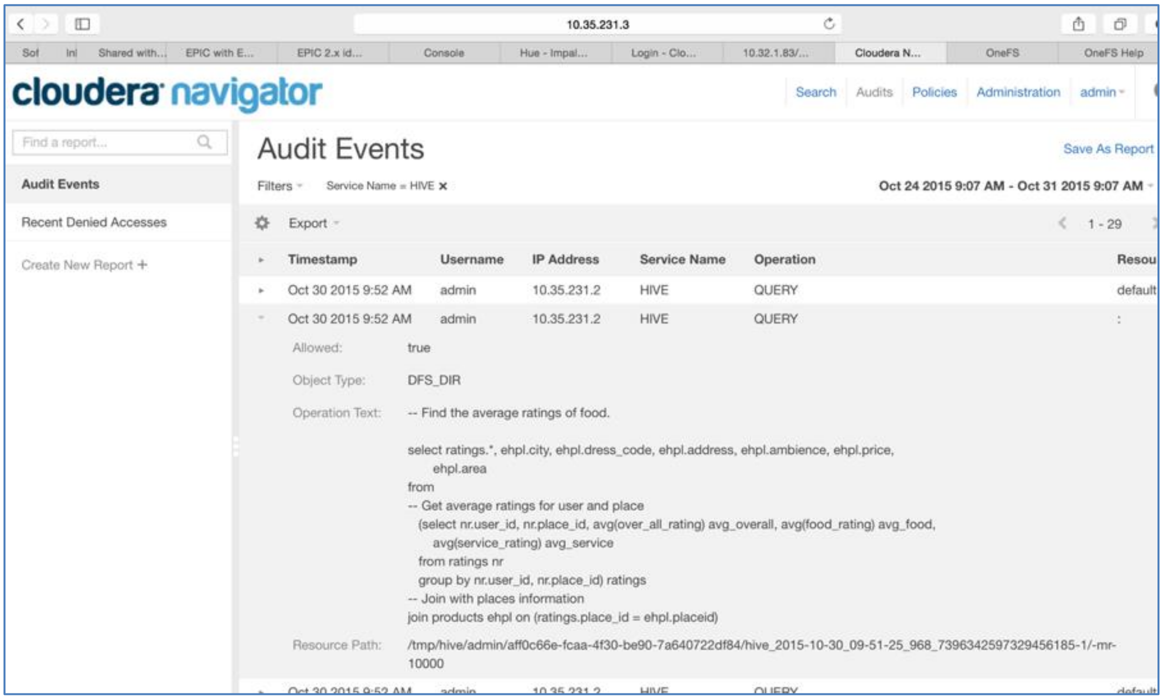


job_1446154862370_0004

Tags:
Source Type: YARN
Template: -- Find the average ratings ...ehpl.placeid)(Stage-1)
Type: Operation Execution
Principal: admin
Inputs: dtap://IsilonHDFS/ratings
Started: Oct 30 2015 9:51 AM
Ended: Oct 30 2015 9:51 AM
Source: YARN

Note that the external table is using the ratings folder in Isilon

Note the corresponding audit events in Cloudera Navigator for HIVE


10.35.231.3
Sof | Inl | Shared with... | EPIC with E... | EPIC 2.x id... | Console | Hue - Impal... | Login - Clo... | 10.32.1.83/... | Cloudera N... | OneFS | OneFS Help
cloudera navigator
Search | Audits | Policies | Administration | admin
Find a report...
Audit Events
Recent Denied Accesses
Create New Report +
Save As Report
Export
1 - 29

Audit Events

Filters · Service Name = HIVE ✕   Oct 24 2015 9:07 AM - Oct 31 2015 9:07 AM

Timestamp	Username	IP Address	Service Name	Operation	Resou
Oct 30 2015 9:52 AM	admin	10.35.231.2	HIVE	QUERY	default
Oct 30 2015 9:52 AM	admin	10.35.231.2	HIVE	QUERY	:

Allowed: true

Object Type: DFS_DIR

Operation Text:

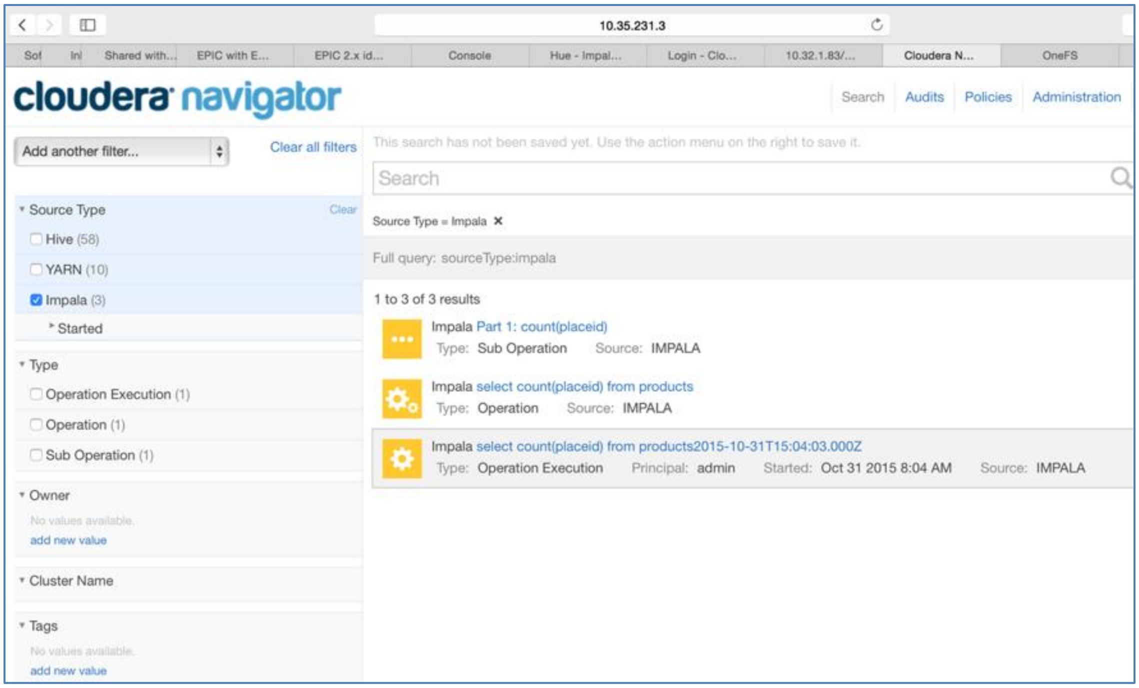
```
-- Find the average ratings of food.

select ratings.*, ehpl.city, ehpl.dress_code, ehpl.address, ehpl.ambience, ehpl.price,
     ehpl.area
from
-- Get average ratings for user and place
  (select nr.user_id, nr.place_id, avg(over_all_rating) avg_overall, avg(food_rating) avg_food,
     avg(service_rating) avg_service
  from ratings nr
  group by nr.user_id, nr.place_id) ratings
-- Join with places information
join products ehpl on (ratings.place_id = ehpl.placeid)
```

Resource Path: /tmp/hive/admin/aff0c66e-fcaa-4f30-be90-7a640722df84/hive_2015-10-30_09-51-25_968_7396342597329456185-1/-mr-10000

Oct 30 2015 9:52 AM	admin	10.35.231.2	HIVE	QUERY	default

IMPALA: Note the time stamp in Hue of 10/31/2015, 8.04am


10.35.231.3
Sof | Inl | Shared with... | EPIC with E... | EPIC 2.x id... | Console | Hue - Impal... | Login - Clo... | 10.32.1.83/... | Cloudera N... | OneFS
cloudera navigator
Search | Audits | Policies | Administration
Add another filter...
Clear all filters
Clear
add new value
add new value

This search has not been saved yet. Use the action menu on the right to save it.

Search

▾ Source Type
- □ Hive (58)
- □ YARN (10)
- ☑ Impala (3)
  - ▸ Started

▾ Type
- □ Operation Execution (1)
- □ Operation (1)
- □ Sub Operation (1)

▾ Owner

No values available.

▾ Cluster Name

▾ Tags

No values available.

Source Type = Impala ✕

Full query: sourceType:impala

1 to 3 of 3 results

Impala Part 1: count(placeid)
Type: Sub Operation   Source: IMPALA

Impala select count(placeid) from products
Type: Operation   Source: IMPALA

Impala select count(placeid) from products2015-10-31T15:04:03.000Z
Type: Operation Execution   Principal: admin   Started: Oct 31 2015 8:04 AM   Source: IMPALA

TEST # 6: Preliminary performance benchmarks using standard development/test hardware in BlueData Labs.

a. Test Goal: Run the Intel HiBench-Terasort test suite using BlueData compute clusters against various data sizes in Isilon

b. Pass Criteria: Successful completion of all the terasort for 100GB and 1TB and measurement of execution times in seconds.

   i. Note: Tuning performance was a non-goal since this environment used 1 Gig network vs. the recommended 10 gig network

   ii. This size of VMs used were limited to 51GB RAM due to the specs of the underlying hardware (dev/test machines)

c. Test Preparation:

1. Specs of Hardware for running BlueData compute cluster:

   Hosts:

   6 workers: 10G NIC, 64GB RAM, 4 cores, 8 cpu, 5 1TB drives
   1 controller: Same as the worker, but with 128GB ram
   CPU: Intel(R) Xeon(R) CPU E5-2609 v2 @ 2.50GHz

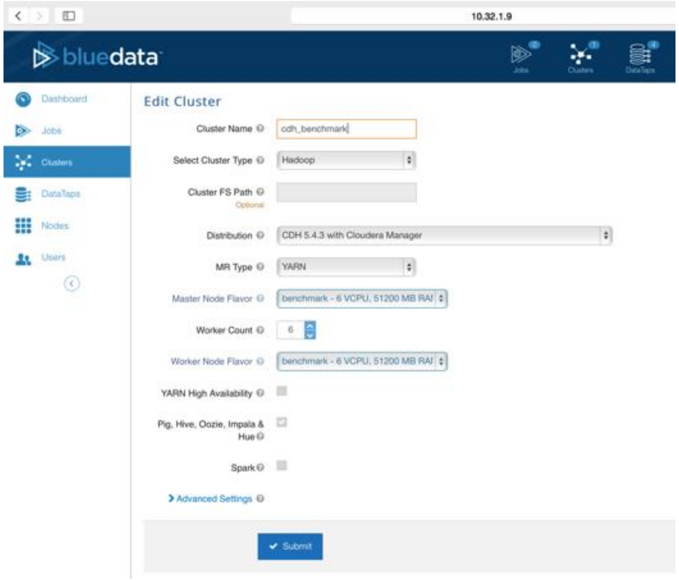
   All boxes have the same 5 drives:
   1TB SSDH hybrid drives (ssd + disk)

   Switch:
   10G switch, model:
   32x 10GBASE-T + 4x SFP+

2. Create special VM flavors for master/worker nodes

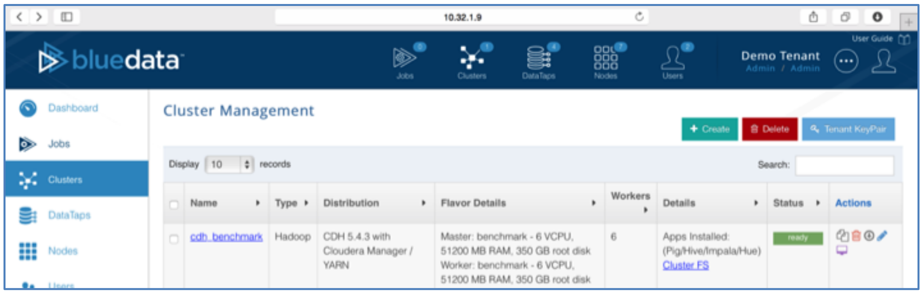
   Master: benchmark - 6 VCPU, 51200 MB RAM, 350 GB root disk
   Worker: benchmark - 6 VCPU, 51200 MB RAM, 350 GB root disk

3. Create CDH 5.4.3 cluster



Edit Cluster

Cluster Name: cdh_benchmark
Select Cluster Type: Hadoop
Cluster FS Path (Optional):
Distribution: CDH 5.4.3 with Cloudera Manager
MR Type: YARN
Master Node Flavor: benchmark - 6 VCPU, 51200 MB RAI
Worker Count: 6
Worker Node Flavor: benchmark - 6 VCPU, 51200 MB RAI
YARN High Availability: □
Pig, Hive, Oozie, Impala & Hue: ☑
Spark: □

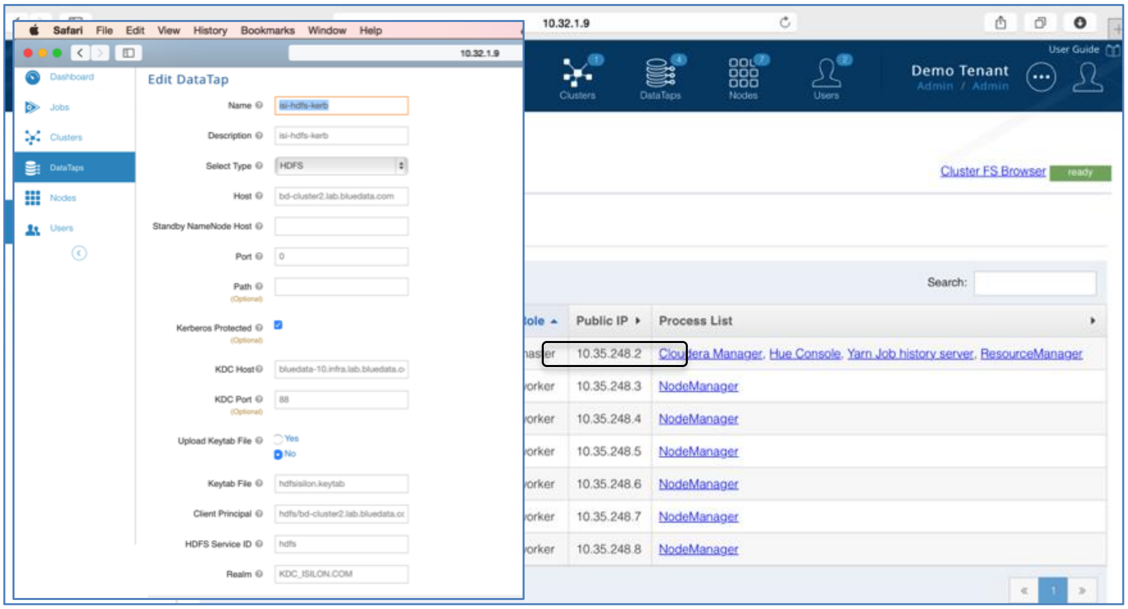


Cluster Management

Display 10 records Search:

Name	Type	Distribution	Flavor Details	Workers	Details	Status	Actions
cdh_benchmark	Hadoop	CDH 5.4.3 with Cloudera Manager / YARN	Master: benchmark - 6 VCPU, 51200 MB RAM, 350 GB root disk Worker: benchmark - 6 VCPU, 51200 MB RAM, 350 GB root disk	6	Apps Installed: (Pig/Hive/Impala/Hue) Cluster FS	ready	

4. Create Isilon DataTap (isi-hdfs)



Edit DataTap

Name: isi-hdfs-kerb

Description: isi-hdfs-kerb

Select Type: HDFS

Host: bd-cluster2.lab.bluedata.com

Standby NameNode Host:

Port: 0

Path (Optional):

Kerberos Protected (Optional): ☑

KDC Host: bluedata-10.infra.lab.bluedata.c

KDC Port (Optional): 88

Upload Keytab File: ○ Yes ● No

Keytab File: hdfsisilon.keytab

Client Principal: hdfs/bd-cluster2.lab.bluedata.c

HDFS Service ID: hdfs

Realm: KDC_ISILON.COM

Search:

ole	Public IP	Process List
aster	10.35.248.2	Cloudera Manager, Hue Console, Yarn Job history server, ResourceManager
orker	10.35.248.3	NodeManager
orker	10.35.248.4	NodeManager
orker	10.35.248.5	NodeManager
orker	10.35.248.6	NodeManager
orker	10.35.248.7	NodeManager
orker	10.35.248.8	NodeManager

d. Test Plan:

1. Login to BlueData tenant

2. Use the BlueData console to a Hadoop M/R job utilizing the aforementioned cluster against data stored in Isilon (i.e. via dtap://)

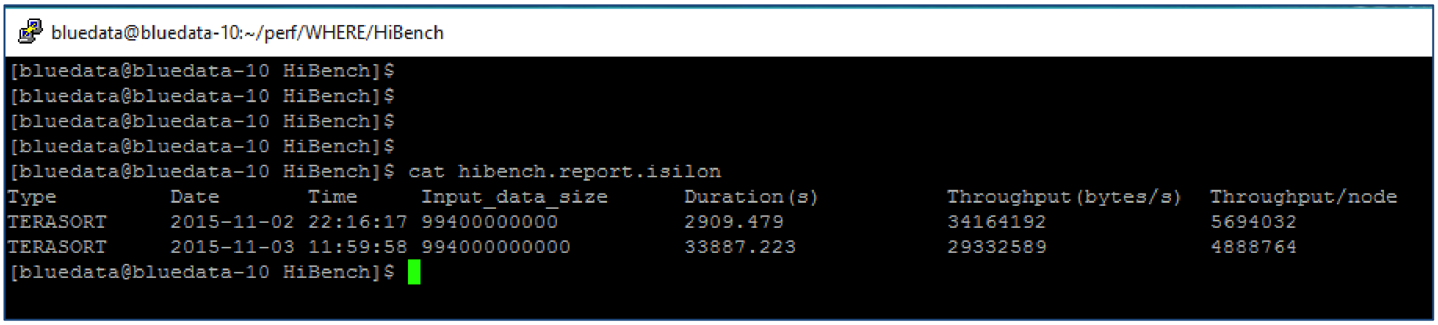
3. Use the Hue Console to create external tables against data stored in Isilon and run a join query

4. Login to Cloudera Navigator – confirm that the metadata search and audit logs show each of the actions as well as sub operations.

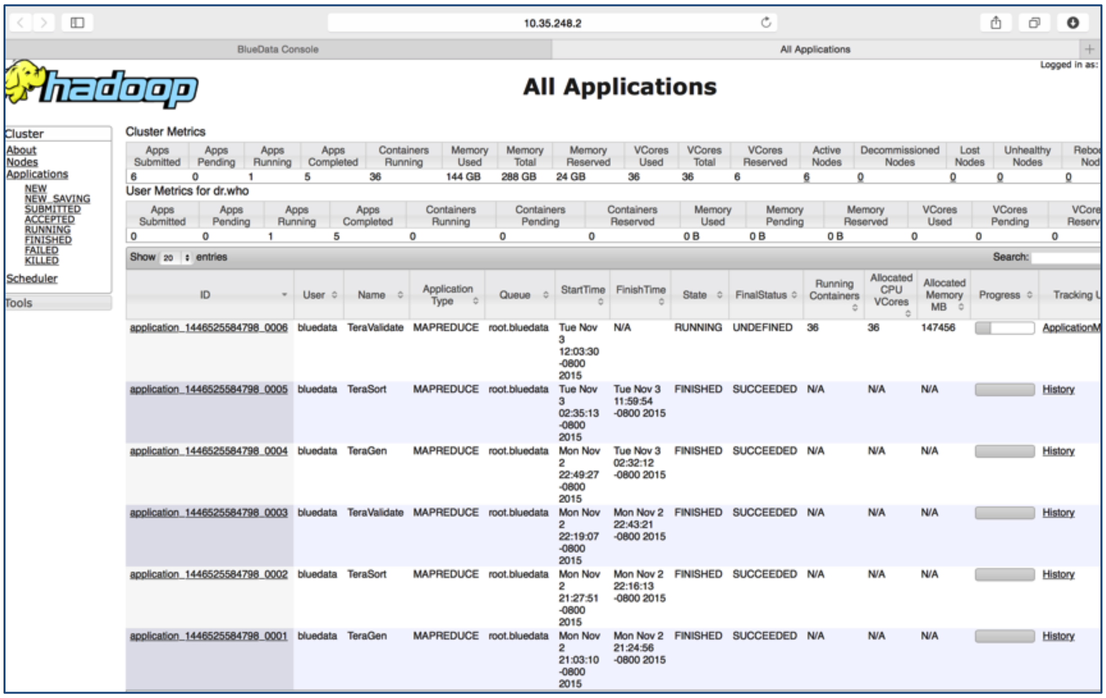
5. **SEE APPENDIX SECTION FOR THE ACTUAL COMMANDS EXECUTED**

e. Result: PASS

TERASORT	100GB	2909.479	seconds
TERASORT	1TB	33887.223	seconds



```
[bluedata@bluedata-10 HiBench]$
[bluedata@bluedata-10 HiBench]$
[bluedata@bluedata-10 HiBench]$
[bluedata@bluedata-10 HiBench]$
[bluedata@bluedata-10 HiBench]$ cat hibench.report.isilon
Type         Date       Time     Input_data_size      Duration(s)          Throughput(bytes/s)  Throughput/node
TERASORT     2015-11-02 22:16:17 99400000000          2909.479             34164192             5694032
TERASORT     2015-11-03 11:59:58 994000000000         33887.223            29332589             4888764
[bluedata@bluedata-10 HiBench]$
```



All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebo Nod
6	0	1	5	36	144 GB	288 GB	24 GB	36	36	6	6	0	0	0	0

User Metrics for dr.who

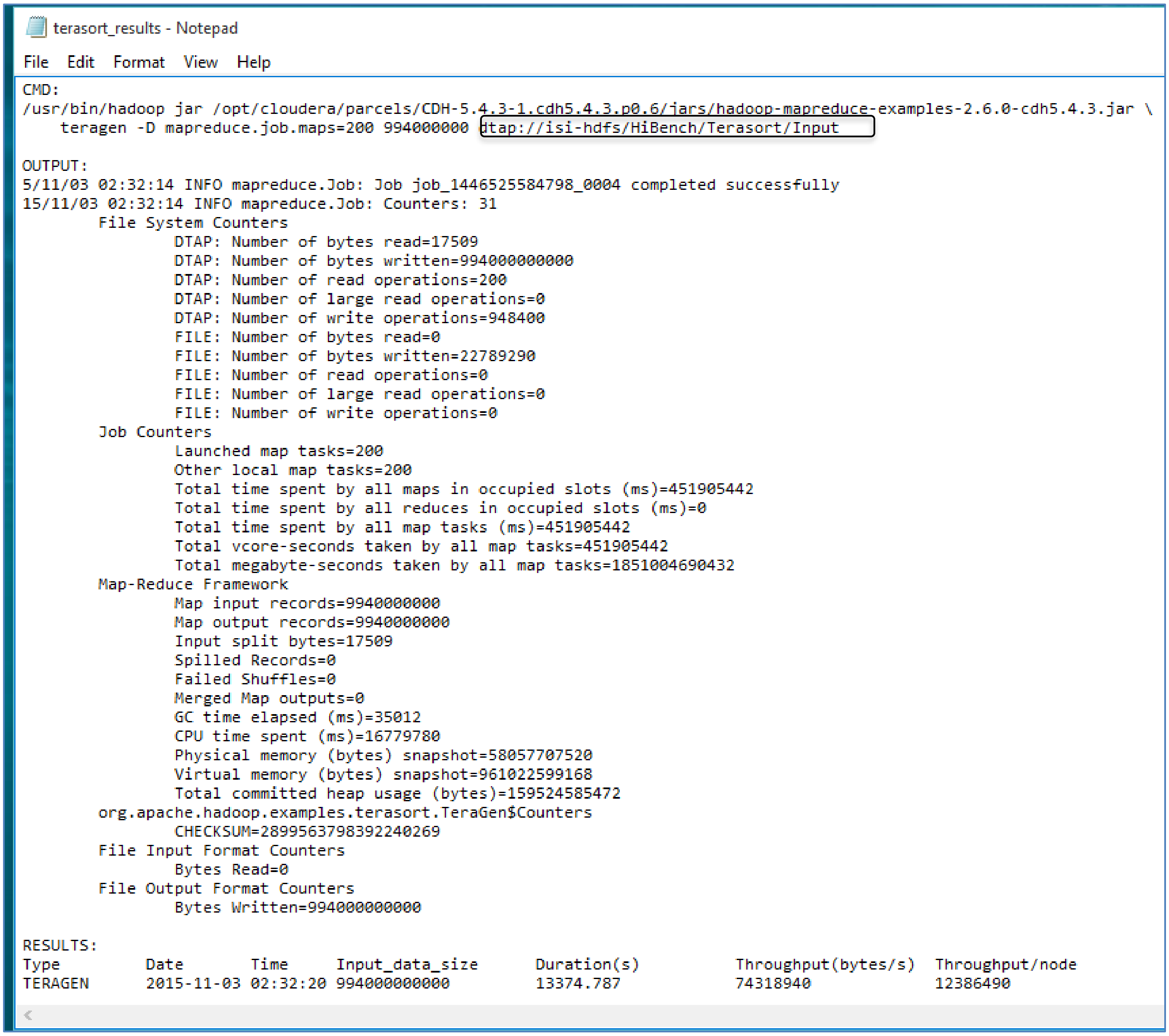
Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Containers Pending	Containers Reserved	Memory Used	Memory Pending	Memory Reserved	VCores Used	VCores Pending	VCore Reserv
0	0	1	5	0	0	0	0 B	0 B	0 B	0	0	0

Show 20 entries Search:

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	Progress	Tracking U
application_1446525584798_0006	bluedata	TeraValidate	MAPREDUCE	root.bluedata	Tue Nov 3 12:03:30 -0800 2015	N/A	RUNNING	UNDEFINED	36	36	147456		ApplicationM
application_1446525584798_0005	bluedata	TeraSort	MAPREDUCE	root.bluedata	Tue Nov 3 02:35:13 -0800 2015	Tue Nov 3 11:59:54 -0800 2015	FINISHED	SUCCEEDED	N/A	N/A	N/A		History
application_1446525584798_0004	bluedata	TeraGen	MAPREDUCE	root.bluedata	Mon Nov 2 22:49:27 -0800 2015	Tue Nov 3 02:32:12 -0800 2015	FINISHED	SUCCEEDED	N/A	N/A	N/A		History
application_1446525584798_0003	bluedata	TeraValidate	MAPREDUCE	root.bluedata	Mon Nov 2 22:19:07 -0800 2015	Mon Nov 2 22:43:21 -0800 2015	FINISHED	SUCCEEDED	N/A	N/A	N/A		History
application_1446525584798_0002	bluedata	TeraSort	MAPREDUCE	root.bluedata	Mon Nov 2 21:27:51 -0800 2015	Mon Nov 2 22:16:13 -0800 2015	FINISHED	SUCCEEDED	N/A	N/A	N/A		History
application_1446525584798_0001	bluedata	TeraGen	MAPREDUCE	root.bluedata	Mon Nov 2 21:03:10 -0800 2015	Mon Nov 2 21:24:56 -0800 2015	FINISHED	SUCCEEDED	N/A	N/A	N/A		History

APPENDIX

Teragen command

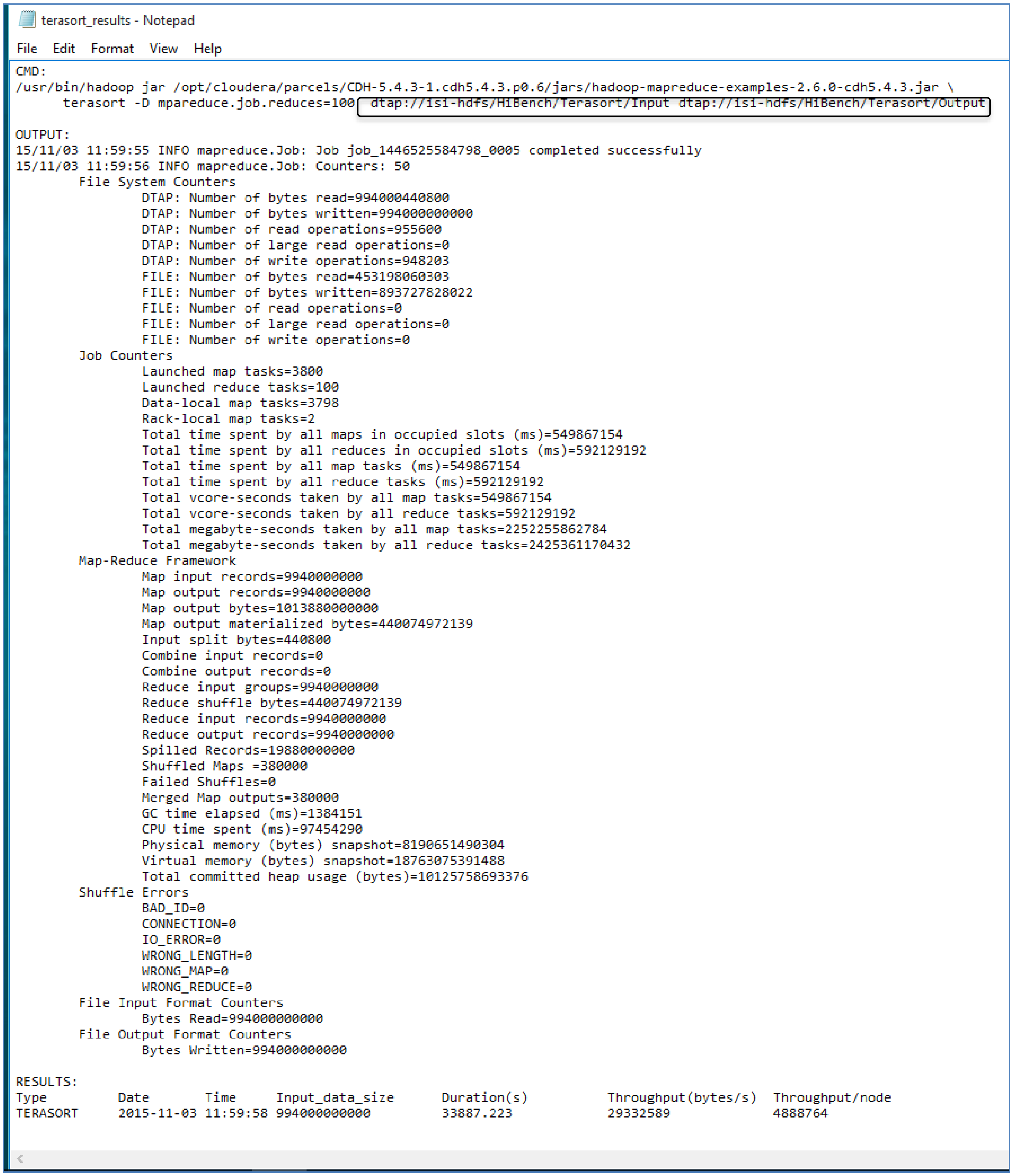


```
CMD:
/usr/bin/hadoop jar /opt/cloudera/parcels/CDH-5.4.3-1.cdh5.4.3.p0.6/jars/hadoop-mapreduce-examples-2.6.0-cdh5.4.3.jar \
    teragen -D mapreduce.job.maps=200 994000000 dtap://isi-hdfs/HiBench/Terasort/Input

OUTPUT:
5/11/03 02:32:14 INFO mapreduce.Job: Job job_1446525584798_0004 completed successfully
15/11/03 02:32:14 INFO mapreduce.Job: Counters: 31
        File System Counters
                DTAP: Number of bytes read=17509
                DTAP: Number of bytes written=994000000000
                DTAP: Number of read operations=200
                DTAP: Number of large read operations=0
                DTAP: Number of write operations=948400
                FILE: Number of bytes read=0
                FILE: Number of bytes written=22789290
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
        Job Counters
                Launched map tasks=200
                Other local map tasks=200
                Total time spent by all maps in occupied slots (ms)=451905442
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=451905442
                Total vcore-seconds taken by all map tasks=451905442
                Total megabyte-seconds taken by all map tasks=1851004690432
        Map-Reduce Framework
                Map input records=9940000000
                Map output records=9940000000
                Input split bytes=17509
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=35012
                CPU time spent (ms)=16779780
                Physical memory (bytes) snapshot=58057707520
                Virtual memory (bytes) snapshot=961022599168
                Total committed heap usage (bytes)=159524585472
        org.apache.hadoop.examples.terasort.TeraGen$Counters
                CHECKSUM=2899563798392240269
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=994000000000

RESULTS:
Type        Date       Time     Input_data_size     Duration(s)         Throughput(bytes/s)  Throughput/node
TERAGEN     2015-11-03 02:32:20 994000000000        13374.787           74318940             12386490
```

Terasort command

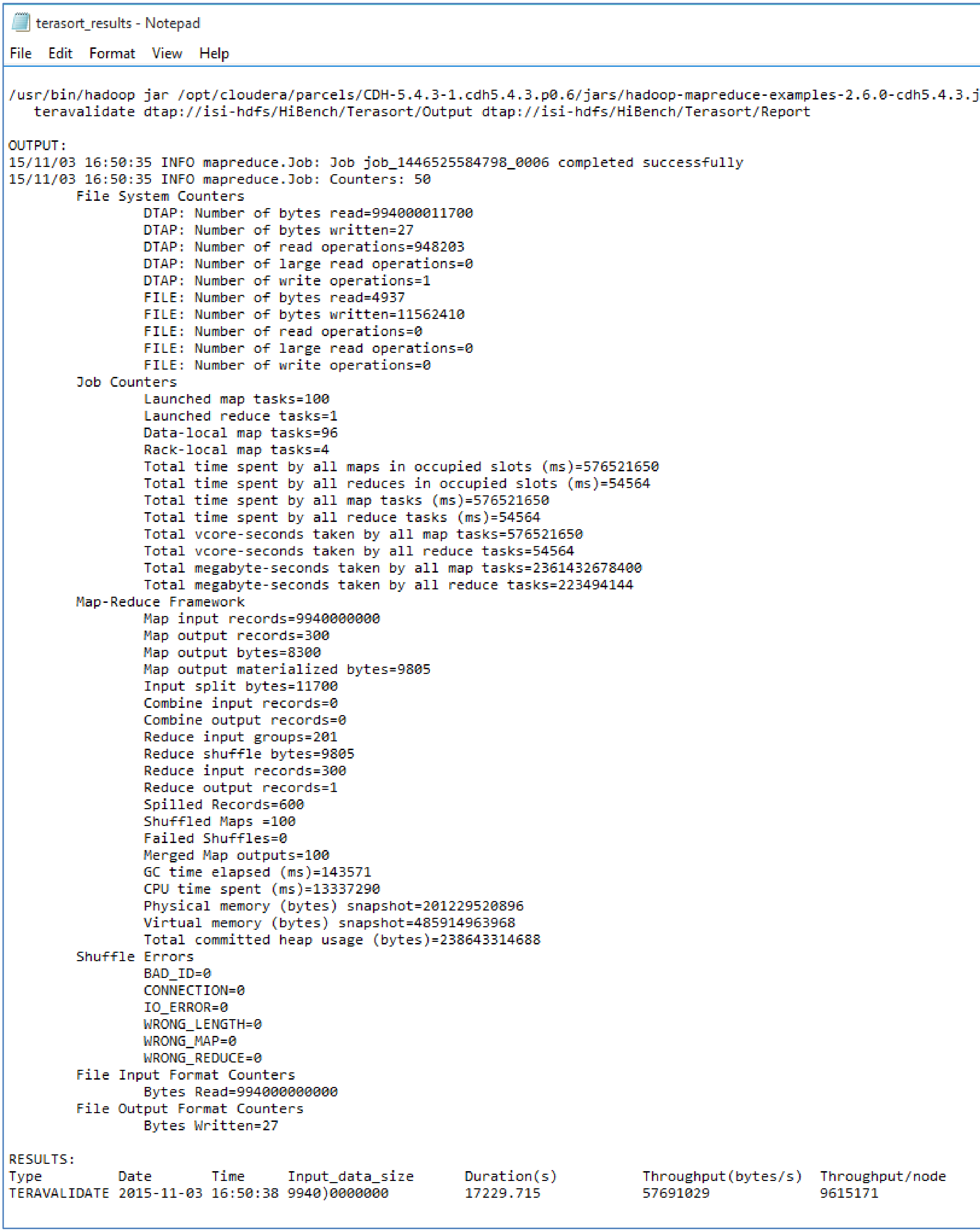


```
CMD:
/usr/bin/hadoop jar /opt/cloudera/parcels/CDH-5.4.3-1.cdh5.4.3.p0.6/jars/hadoop-mapreduce-examples-2.6.0-cdh5.4.3.jar \
      terasort -D mpareduce.job.reduces=100  dtap://isi-hdfs/HiBench/Terasort/Input dtap://isi-hdfs/HiBench/Terasort/Output

OUTPUT:
15/11/03 11:59:55 INFO mapreduce.Job: Job job_1446525584798_0005 completed successfully
15/11/03 11:59:56 INFO mapreduce.Job: Counters: 50
        File System Counters
                DTAP: Number of bytes read=994000440800
                DTAP: Number of bytes written=994000000000
                DTAP: Number of read operations=955600
                DTAP: Number of large read operations=0
                DTAP: Number of write operations=948203
                FILE: Number of bytes read=453198060303
                FILE: Number of bytes written=893727828022
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
        Job Counters
                Launched map tasks=3800
                Launched reduce tasks=100
                Data-local map tasks=3798
                Rack-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=549867154
                Total time spent by all reduces in occupied slots (ms)=592129192
                Total time spent by all map tasks (ms)=549867154
                Total time spent by all reduce tasks (ms)=592129192
                Total vcore-seconds taken by all map tasks=549867154
                Total vcore-seconds taken by all reduce tasks=592129192
                Total megabyte-seconds taken by all map tasks=2252255862784
                Total megabyte-seconds taken by all reduce tasks=2425361170432
        Map-Reduce Framework
                Map input records=9940000000
                Map output records=9940000000
                Map output bytes=1013880000000
                Map output materialized bytes=440074972139
                Input split bytes=440800
                Combine input records=0
                Combine output records=0
                Reduce input groups=9940000000
                Reduce shuffle bytes=440074972139
                Reduce input records=9940000000
                Reduce output records=9940000000
                Spilled Records=19880000000
                Shuffled Maps =380000
                Failed Shuffles=0
                Merged Map outputs=380000
                GC time elapsed (ms)=1384151
                CPU time spent (ms)=97454290
                Physical memory (bytes) snapshot=8190651490304
                Virtual memory (bytes) snapshot=18763075391488
                Total committed heap usage (bytes)=10125758693376
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=994000000000
        File Output Format Counters
                Bytes Written=994000000000

RESULTS:
Type         Date       Time     Input_data_size      Duration(s)          Throughput(bytes/s)  Throughput/node
TERASORT     2015-11-03 11:59:58 994000000000         33887.223            29332589             4888764
```

Teravalidate command



```
/usr/bin/hadoop jar /opt/cloudera/parcels/CDH-5.4.3-1.cdh5.4.3.p0.6/jars/hadoop-mapreduce-examples-2.6.0-cdh5.4.3.j
   teravalidate dtap://isi-hdfs/HiBench/Terasort/Output dtap://isi-hdfs/HiBench/Terasort/Report

OUTPUT:
15/11/03 16:50:35 INFO mapreduce.Job: Job job_1446525584798_0006 completed successfully
15/11/03 16:50:35 INFO mapreduce.Job: Counters: 50
        File System Counters
                DTAP: Number of bytes read=994000011700
                DTAP: Number of bytes written=27
                DTAP: Number of read operations=948203
                DTAP: Number of large read operations=0
                DTAP: Number of write operations=1
                FILE: Number of bytes read=4937
                FILE: Number of bytes written=11562410
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
        Job Counters
                Launched map tasks=100
                Launched reduce tasks=1
                Data-local map tasks=96
                Rack-local map tasks=4
                Total time spent by all maps in occupied slots (ms)=576521650
                Total time spent by all reduces in occupied slots (ms)=54564
                Total time spent by all map tasks (ms)=576521650
                Total time spent by all reduce tasks (ms)=54564
                Total vcore-seconds taken by all map tasks=576521650
                Total vcore-seconds taken by all reduce tasks=54564
                Total megabyte-seconds taken by all map tasks=2361432678400
                Total megabyte-seconds taken by all reduce tasks=223494144
        Map-Reduce Framework
                Map input records=9940000000
                Map output records=300
                Map output bytes=8300
                Map output materialized bytes=9805
                Input split bytes=11700
                Combine input records=0
                Combine output records=0
                Reduce input groups=201
                Reduce shuffle bytes=9805
                Reduce input records=300
                Reduce output records=1
                Spilled Records=600
                Shuffled Maps =100
                Failed Shuffles=0
                Merged Map outputs=100
                GC time elapsed (ms)=143571
                CPU time spent (ms)=13337290
                Physical memory (bytes) snapshot=201229520896
                Virtual memory (bytes) snapshot=485914963968
                Total committed heap usage (bytes)=238643314688
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=994000000000
        File Output Format Counters
                Bytes Written=27

RESULTS:
Type         Date       Time     Input_data_size     Duration(s)         Throughput(bytes/s)  Throughput/node
TERAVALIDATE 2015-11-03 16:50:38 9940)0000000        17229.715           57691029             9615171
```