CLOUDERA
Educational Services

Cloudera Administrator Training for CDP PVC Base

200712b

Introduction

Chapter 1

Course Chapters

Trademark Information

- **The names and logos of Apache products mentioned in Cloudera training courses, including those listed below, are trademarks of the Apache Software Foundation**

Apache Accumulo	Apache Hive	Apache Pig
Apache Avro	Apache Impala	Apache Ranger
Apache Ambari	Apache Kafka	Apache Sentry
Apache Atlas	Apache Knox	Apache Solr
Apache Bigtop	Apache Kudu	Apache Spark
Apache Crunch	Apache Lucene	Apache Sqoop
Apache Druid	Apache Mahout	Apache Storm
Apache Flink	Apache NiFi	Apache Tez
Apache Flume	Apache Oozie	Apache Tika
Apache Hadoop	Apache ORC	Apache Zeppelin
Apache HBase	Apache Parquet	Apache ZooKeeper
Apache HCatalog	Apache Phoenix	

- **All other product names, logos, and brands cited herein are the property of their respective owners**

Chapter Topics

Introduction

- **About This Course**
- Introductions
- About Cloudera
- About Cloudera Educational Services
- Course Logistics

Course Objectives

During this course, you will learn

- **About the topology of a typical Cloudera cluster and the role the major components play in the cluster**
- **How to install Cloudera Manager and CDP**
- **How to use Cloudera Manager to create, configure, deploy, and monitor a cluster**
- **What tools Cloudera provides to ingest data from outside sources into a cluster**
- **How to configure cluster components for optimal performance**
- **What routine tasks are necessary to maintain a cluster, including updating to a new version of CDP**
- **About detecting, troubleshooting, and repairing problems**
- **Key Cloudera security features**

Chapter Topics

Introduction

- About This Course
- **Introductions**
- About Cloudera
- About Cloudera Educational Services
- Course Logistics

Introductions

- **About your instructor**
- **About you**
  - Currently, what do you do at your workplace?
  - What is your experience with database technologies, programming, and query languages?
  - How much experience do you have with UNIX or Linux?
  - What is your experience with big data?
  - What do you expect to gain from this course? What would you like to be able to do at the end that you cannot do now?

Chapter Topics

Introduction

- About This Course
- Introductions
- **About Cloudera**
- About Cloudera Educational Services
- Course Logistics

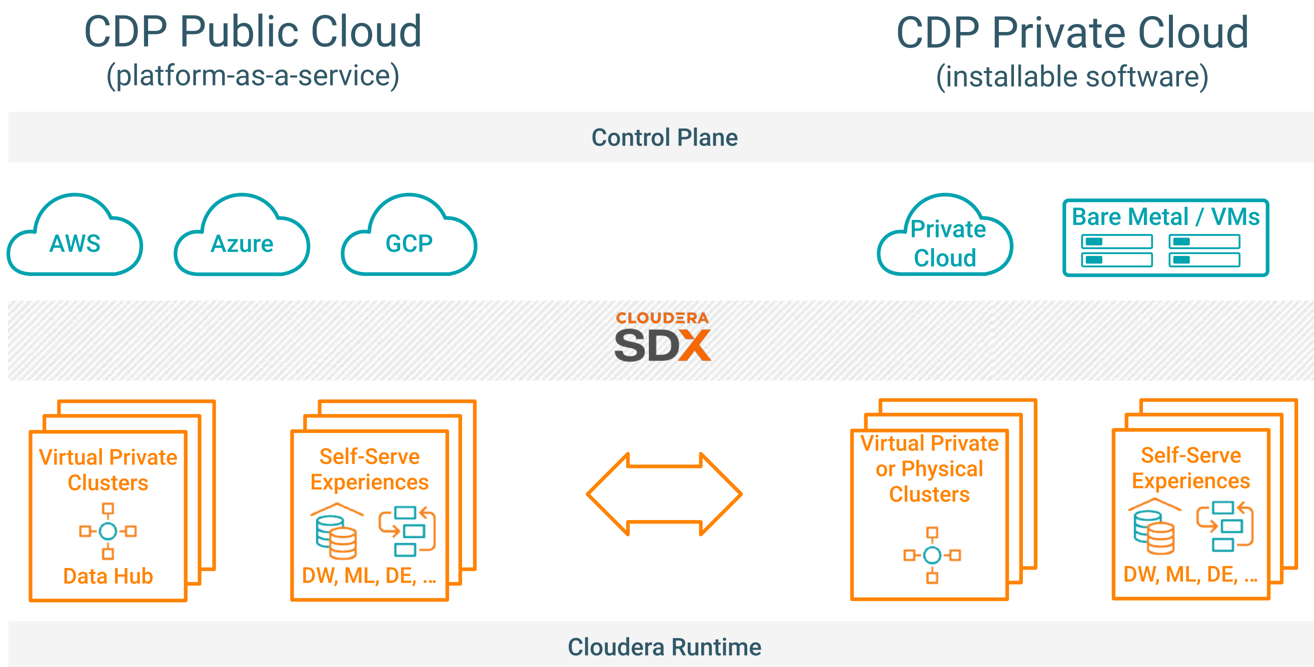
About Cloudera

CLOUDERA

THE ENTERPRISE DATA CLOUD COMPANY

- **Cloudera (founded 2008) and Hortonworks (founded 2011) merged in 2019**
- **The new Cloudera improves on the best of both companies**
  - Introduced the world's first Enterprise Data Cloud
  - Delivers an comprehensive platform for any data from the Edge to AI
  - Leads in training, certification, support, and consulting for data professionals
  - Remains committed to open source and open standards

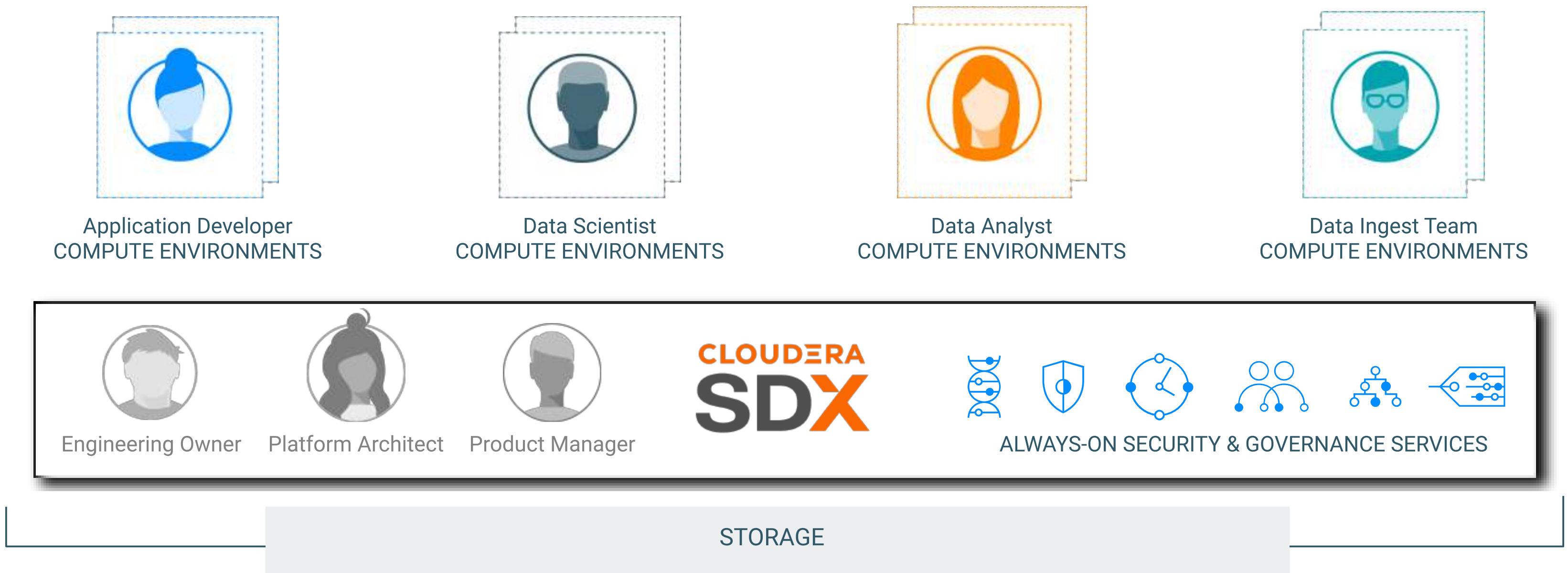
Cloudera Data Platform



A suite of products to collect, curate, report, serve, and predict

- **Cloud native or bare metal deployment**
- **Powered by open source**
- **Analytics from the Edge to AI**
- **Unified data control plane**
- **Shared Data Experience (SDX)**

Cloudera Shared Data Experience (SDX)



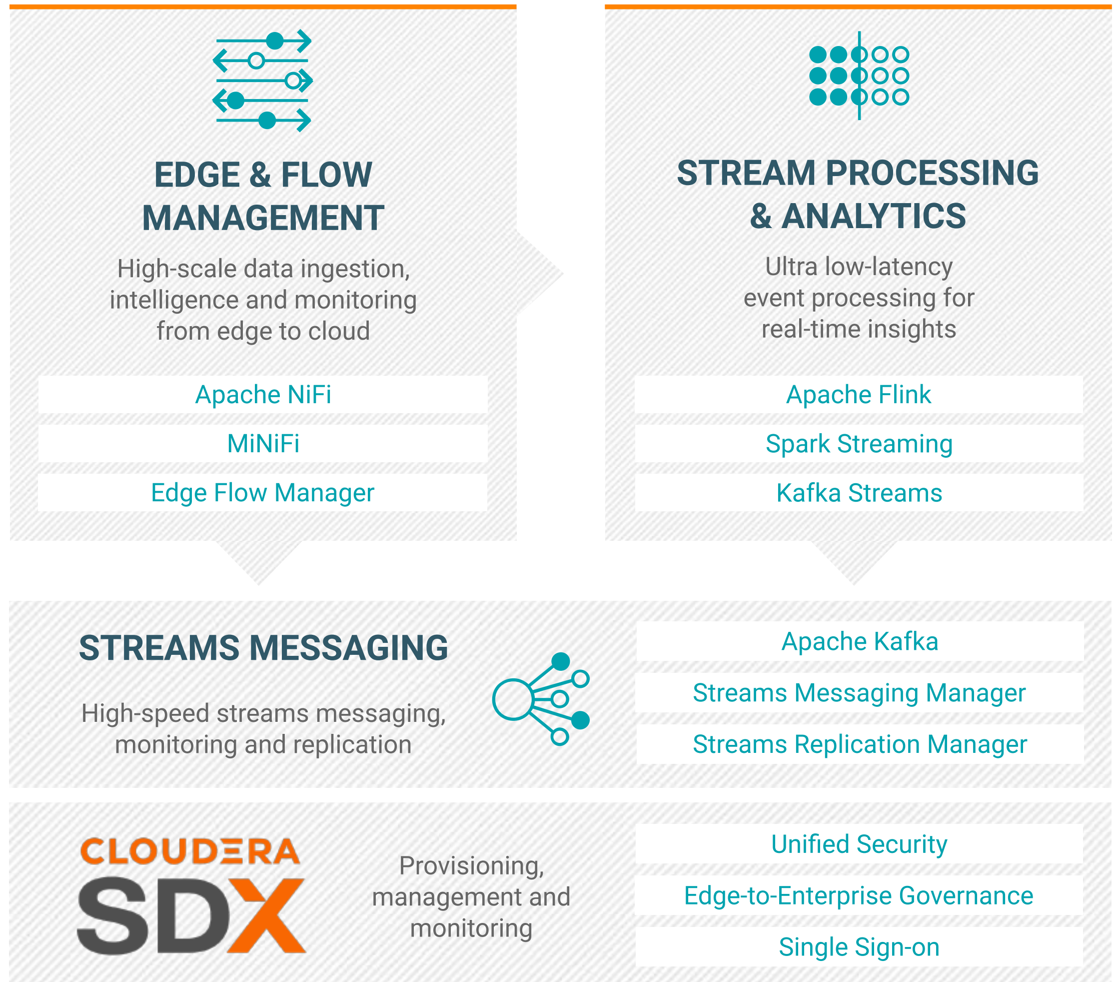
- **Full data lifecycle:** Manages your data from ingestion to actionable insights
- **Unified security:** Protects sensitive data with consistent controls
- **Consistent governance:** Enables safe self-service access

Self-Serve Experiences for Cloud Form Factors

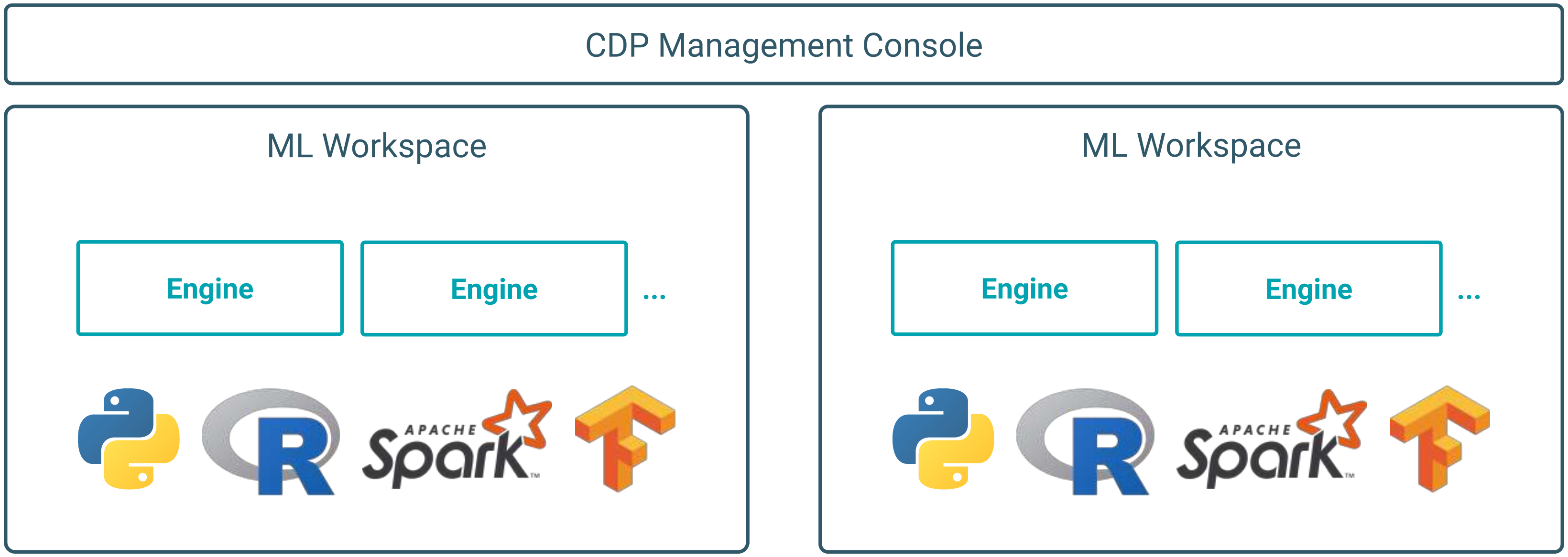
- **Services customized for specific steps in the data lifecycle**
  - Emphasize productivity and ease of use
  - Auto-scale compute resources to match changing demands
  - Isolate compute resources to maintain workload performance

Cloudera DataFlow

- Data-in-motion platform
- Reduces data integration development time
- Manages and secures your data from edge to enterprise

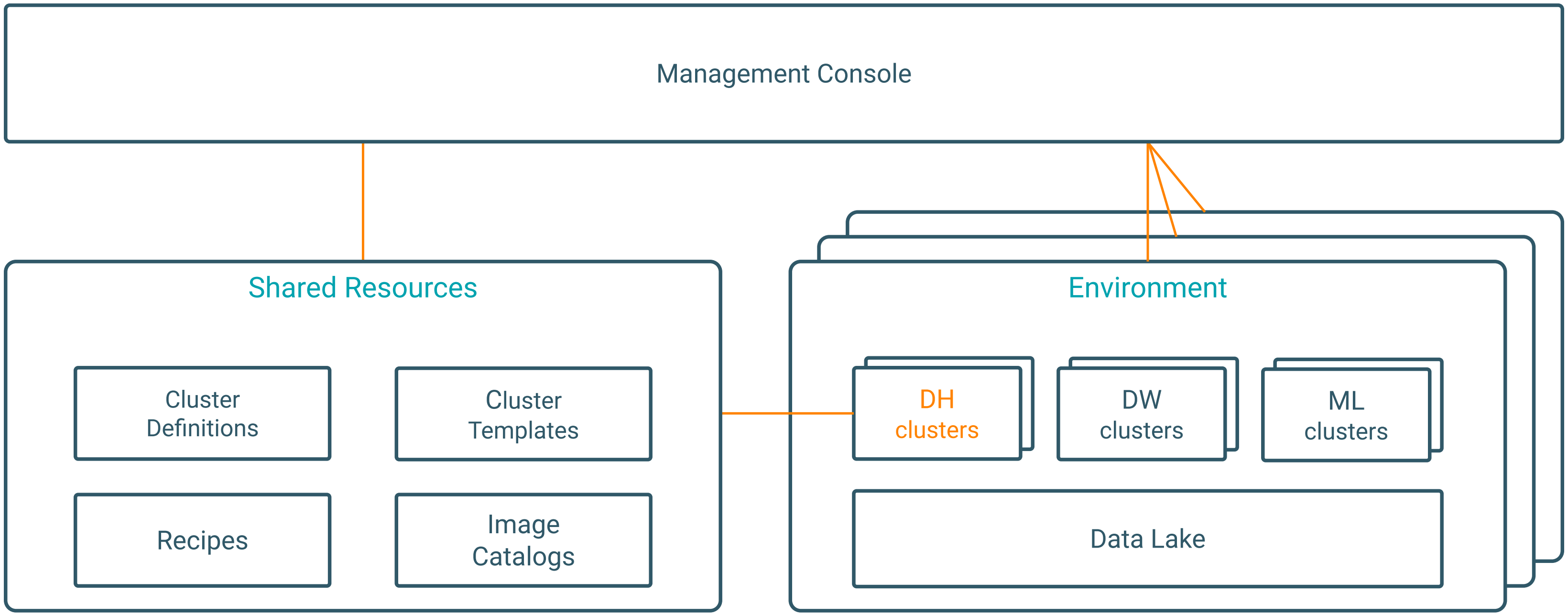


Cloudera Machine Learning



- **Cloud-native enterprise machine learning**
  - Fast, easy, and secure self-service data science in enterprise environments
  - Direct access to a secure cluster running Spark and other tools
  - Isolated environments for running Python, R, and Scala code
  - Teams, version control, collaboration, and project sharing

Cloudera Data Hub



Customize your own experience in cloud form factors

- **Integrated suite of analytic engines**
- **Cloudera SDX applies consistent security and governance**
- **Fueled by open source innovation**

Chapter Topics

Introduction

- About This Course
- Introductions
- About Cloudera
- **About Cloudera Educational Services**
- Course Logistics

Cloudera Educational Services

- **We offer a variety of ways to take our courses**
  - Instructor-led, both in physical and virtual classrooms
    - Private and customized courses also available
  - Self-paced, through Cloudera OnDemand
- **Courses for all kinds of data professionals**
  - Executives and managers
  - Data scientists and machine learning specialists
  - Data analysts
  - Developers and data engineers
  - System administrators
  - Security professionals

Cloudera Education Catalog

- **A broad portfolio across multiple platforms**
  - Not all courses shown here
  - See our website for the complete catalog

<table>
<tr><td>ADMINISTRATOR</td><td>Administrator
CDH|HDP|CDP</td><td>Security
CDH|HDP</td><td>NiFi
CDF</td><td>AWS Fundamentals for CDP</td><td></td><td></td></tr>
<tr><td>DATA ANALYST</td><td>Data Analyst
CDH | CDP</td><td>Hive 3
HDP</td><td>Kudu
CDH</td><td>Cloudera Data Warehouse
CDP</td><td></td><td></td></tr>
<tr><td>DEVELOPER & DATA ENGINEER</td><td>Spark
CDH | HDP</td><td>Spark Performance Tuning
CDH</td><td>Stream Developer
CDF</td><td>Kafka
CDP</td><td>Flink & SQL Stream Builder
CDP</td><td>Architecture Workshop
CDH</td></tr>
<tr><td>DATA SCIENTIST</td><td>Data Scientist
CDH|HDP|CDP</td><td>Cloudera DS Workbench
CDH | HDP</td><td></td><td>CML
CDP</td><td></td><td></td></tr>
</table>

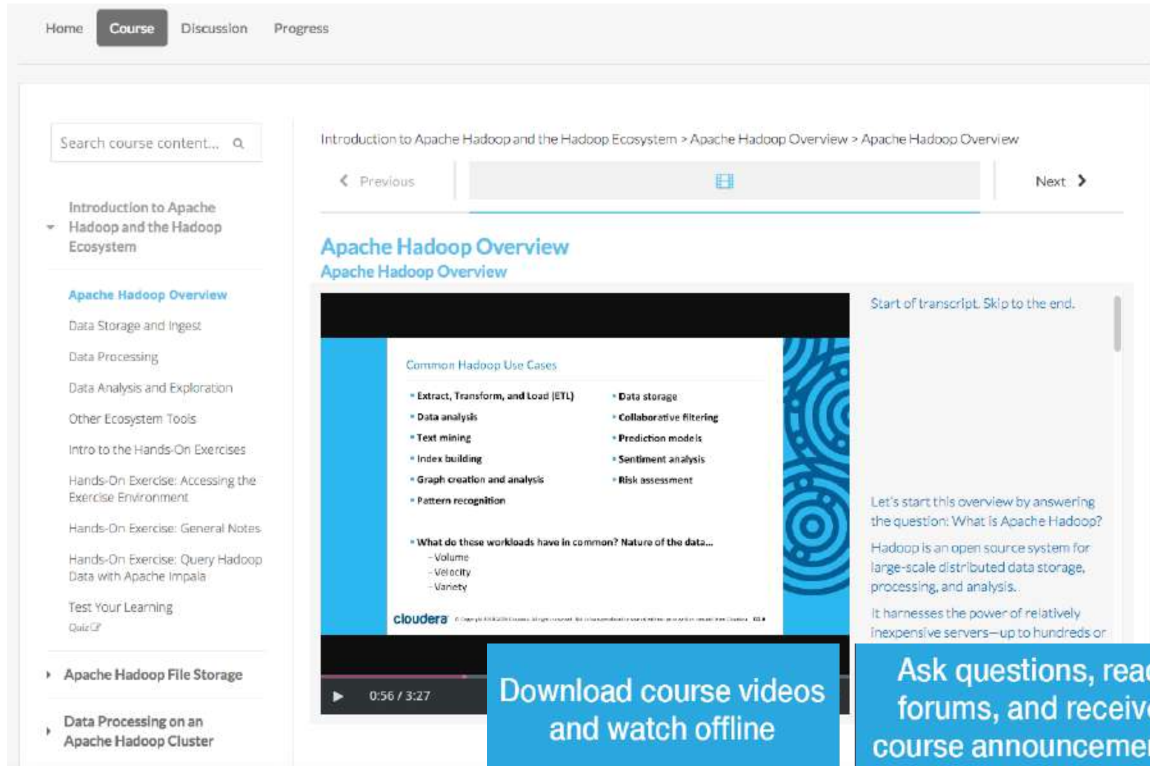
Private Class
Public Class
OnDemand

Cloudera OnDemand

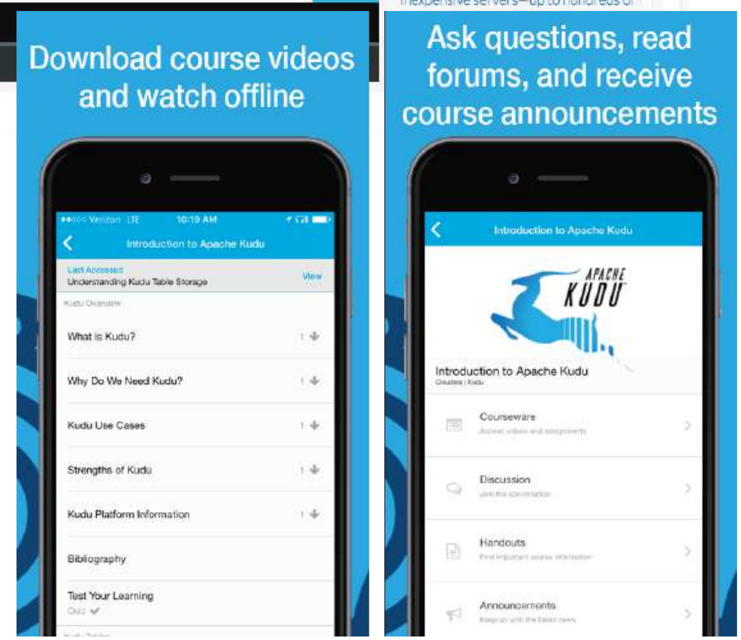
- **Our OnDemand catalog includes**
  - Courses for developers, data analysts, administrators, and data scientists, updated regularly
  - Exclusive OnDemand-only courses, such as those covering security and Cloudera Data Science Workbench
  - Free courses such as *Essentials* and *Cloudera Director* available to all with or without an OnDemand account
- **Features include**
  - Video lectures and demonstrations with searchable transcripts
  - Hands-on exercises through a browser-based virtual environment
  - Discussion forums monitored by Cloudera course instructors
  - Searchable content within and across courses
- **Purchase access to a library of courses or individual courses**
- **See the Cloudera OnDemand information page for more details or to make a purchase, or go directly to the OnDemand Course Catalog**

Accessing Cloudera OnDemand

- **Cloudera OnDemand subscribers can access their courses online through a web browser**



- **Cloudera OnDemand is also available through an iOS app**
  - Search for "Cloudera OnDemand" in the iOS App Store



Cloudera Certification

- **The leader in Apache Hadoop-based certification**
- **Cloudera certification exams favor hands-on, performance-based problems that require execution of a set of real-world tasks against a live, working cluster**
- **We offer two levels of certifications**
  - **Cloudera Certified Associate (CCA)**
    - CCA Spark and Hadoop Developer
    - CCA Data Analyst
    - CCA CDH Administrator and CCA HDP Administrator
  - **Cloudera Certified Professional (CCP)**
    - CCP Data Engineer

Chapter Topics

Introduction

- About This Course
- Introductions
- About Cloudera
- About Cloudera Educational Services
- **Course Logistics**

Logistics

- Class start and finish time
- Lunch
- Breaks
- Restrooms
- Wi-Fi access
- Virtual machines

Downloading the Course Materials

1. **Log in using `https://university.cloudera.com/user`**
   - If necessary, use the **Register Now** link on the right to create an account
   - If you have forgotten your password, use the **Reset Password** link

2. **Scroll down to find this course**
   - If necessary, click **My Learning** under the photo
   - You may also want to use the **Current** filter
3. **Select the course title**
4. **Click the Resources tab**
5. **Click a file to download it**

Cloudera Data Platform

Chapter 2

Course Chapters

Cloudera Data Platform

By the end of this chapter, you will be able to

- **Explain industry trends for big data**
- **Describe steps in the data-driven journey**
- **Describe key features of the Enterprise Data Cloud**
- **Describe the components of the Cloudera Data Platform (CDP)**
- **Explain CDP Form Factors**

Chapter Topics

Cloudera Data Platform

- **Industry Trends for Big Data**
- The Challenge to Become Data-Driven
- The Enterprise Data Cloud
- CDP Overview
- CDP Form Factors
- Essential Points
- Hands-On Exercise: Configure the Exercise Network

The Big Data Imperative

- **Data is extremely valuable**
- **Proper use of data is crucial for success**
- **Effective use of data provides a competitive advantage**
- **More sophisticated analysis and use of data is needed**
- **The capability to manage and analyze data from the edge to AI is important**

EDGE TO AI

Trends in Architecture and Technology

- **Cloud experience that is simple, flexible, on-demand**
- **Separation of compute and storage for optimized deployments and cost savings**
- **Kubernetes with containers for flexibility and efficiency**
- **Analytics with real-time streaming, machine learning (ML), and artificial intelligence (AI)**

Chapter Topics

Cloudera Data Platform

- Industry Trends for Big Data
- **The Challenge to Become Data-Driven**
- The Enterprise Data Cloud
- CDP Overview
- CDP Form Factors
- Essential Points
- Hands-On Exercise: Configure the Exercise Network

The Challenge for Effective Data Use

IMPOSSIBLE?
POSSIBLE.

- **Many organizations do not effectively use data**
- **They have difficulty collecting, analyzing, and securing data**
  - Analytic workloads are often siloed
  - Data resides in multiple, separate locations
  - There is no comprehensive security across multiple data locations
  - Proprietary algorithms and storage exist in cloud environments
- **This hinders their ability to respond quickly to customers and business needs**
- **Moving to effective data use can make the impossible possible**

The Data-Driven Journey

- **Data-driven organizations have a competitive advantage**
- **Need to move beyond siloed, disjointed data collection and analysis**
- **Stages in the journey**
  - Comprehensive data visibility
  - Organizational productivity
  - Enterprise transformation

- Extensive Customer Insights
- Advanced Threat Detection
- Risk Modelling and Analysis
- Applied Machine Learning

- Marketing Systems Integration
- Preventive and Proactive Maintenance
- Exploratory Data Science

- Data Warehouse

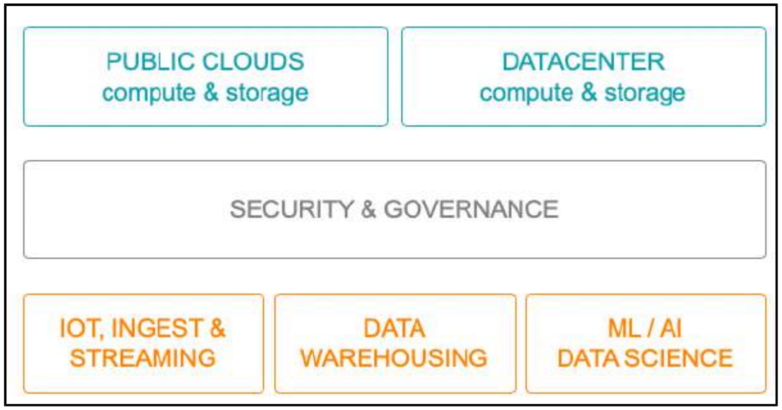
Chapter Topics

Cloudera Data Platform

- Industry Trends for Big Data
- The Challenge to Become Data-Driven
- **The Enterprise Data Cloud**
- CDP Overview
- CDP Form Factors
- Essential Points
- Hands-On Exercise: Configure the Exercise Network

The Enterprise Data Cloud Enables Data-Driven Organizations

- **Key architectural elements**
- **Essential features**
  - Hybrid and multi-cloud
  - Secure and governed
  - Multi-function analytics
  - Open platform
- **The Cloudera Data Platform (CDP) is an *implementation* of the Enterprise Data Cloud**



The World's First Enterprise Data Cloud

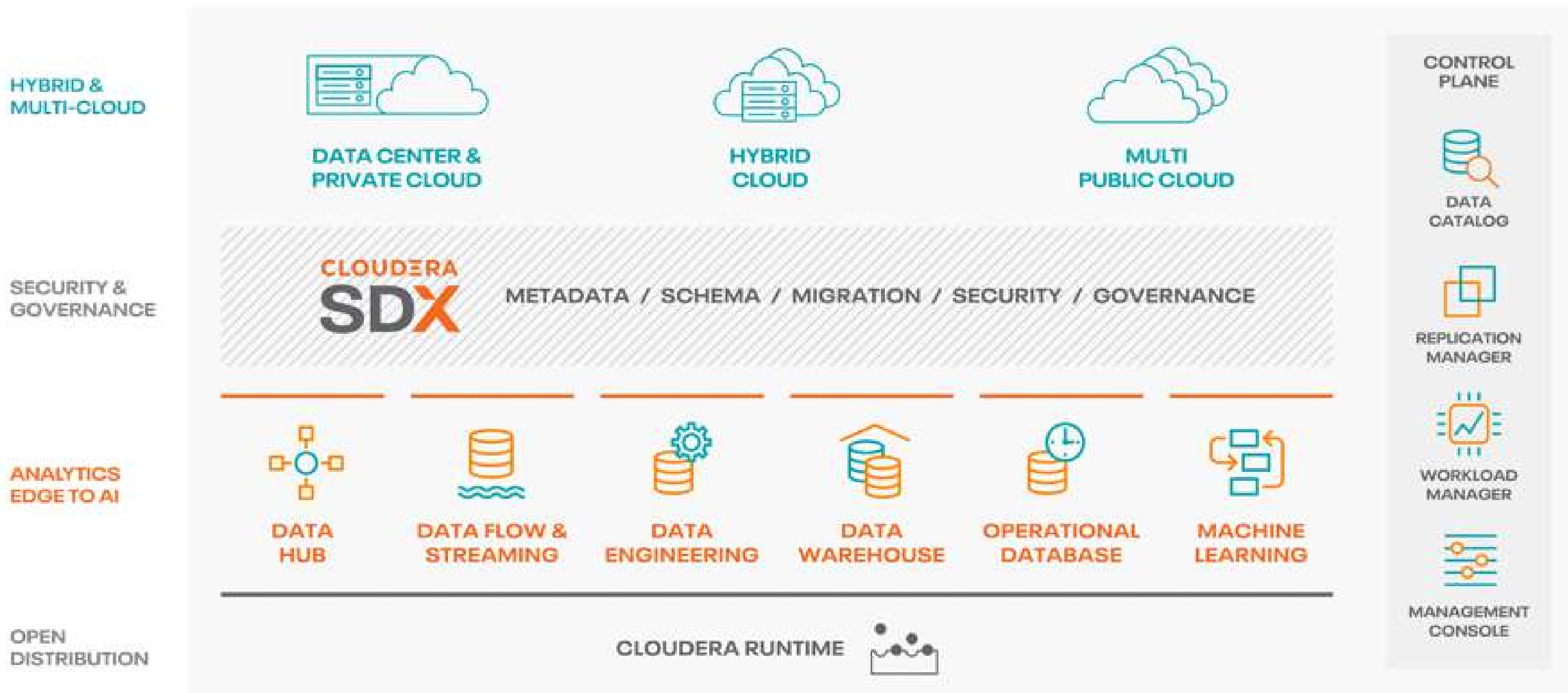
- **Finally, a platform for both IT and the business, Cloudera Data Platform is:**
  - On-premises and public cloud
  - Multi-cloud and multi-function
  - Simple to use and secure by design
  - Manual and automated
  - Open and extensible
  - For data engineers and data scientists

Chapter Topics

Cloudera Data Platform

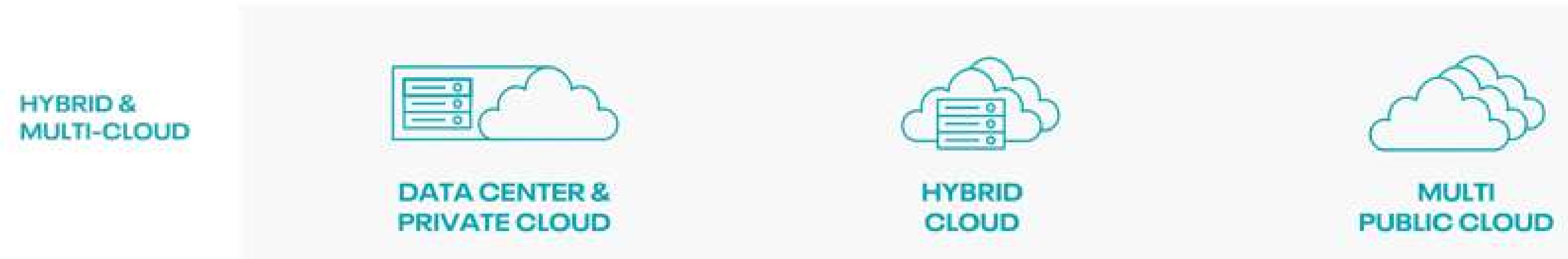
- Industry Trends for Big Data
- The Challenge to Become Data-Driven
- The Enterprise Data Cloud
- **CDP Overview**
- CDP Form Factors
- Essential Points
- Hands-On Exercise: Configure the Exercise Network

The Cloudera Data Platform (CDP)



- **The world's first Enterprise Data Cloud implementation**
- **Fulfills the essential requirements of the Enterprise Data Cloud architecture**

CDP - Multiple Deployment Editions



- **CDP provides data analysis, management and control capabilities in any deployment environment**
  - Private Cloud - built for hybrid cloud
    - Leveraging on-premise container management infrastructure
    - Seamlessly connects on premises to public clouds
  - Public Cloud - multiple cloud vendors
    - Services managed by Cloudera
    - Supports multiple cloud vendors
    - Data secured in your own Virtual Private Cloud (VPC)

CDP Security and Governance - SDX

- **The Shared Data Experience (SDX) provides security and governance**
- **Spans all Cloudera products**
- **Policies are set and enforced on all data and workloads**
- **Centralized schema management leveraged across the platform**

CDP Multi-Function Analytics

- Data Hub **- clusters deployed for specific use cases and workloads**
- Data Flow & Streaming **- edge and flow management, stream processing, streams management, and streaming analytics**
- Data Engineering **- use ETL jobs to create data pipelines**
- Data Warehouse **- self-service creation of data warehouses and data marts from multiple sources**
- Operational Database **- provides instant read/write access to large datasets**
- Machine Learning **- self-service cloud-native ML capability with clusters running on Kubernetes**

CDP Open Source Distribution - Cloudera Runtime

- **The distribution of core components and tools in CDP**
- **Made up of multiple open source projects**
- **Maintained, supported, versioned, and packaged as a single entity by Cloudera**

CDP Control Plane (1)

- **Integrated set of services to manage infrastructure, data, and analytic workloads:**
- **Data Catalog**
  - Understand, manage, secure, search, and govern data assets
- **Replication Manager**
  - Copy and migrate data, metadata, and policies between deployment environments

CDP Control Plane (2)

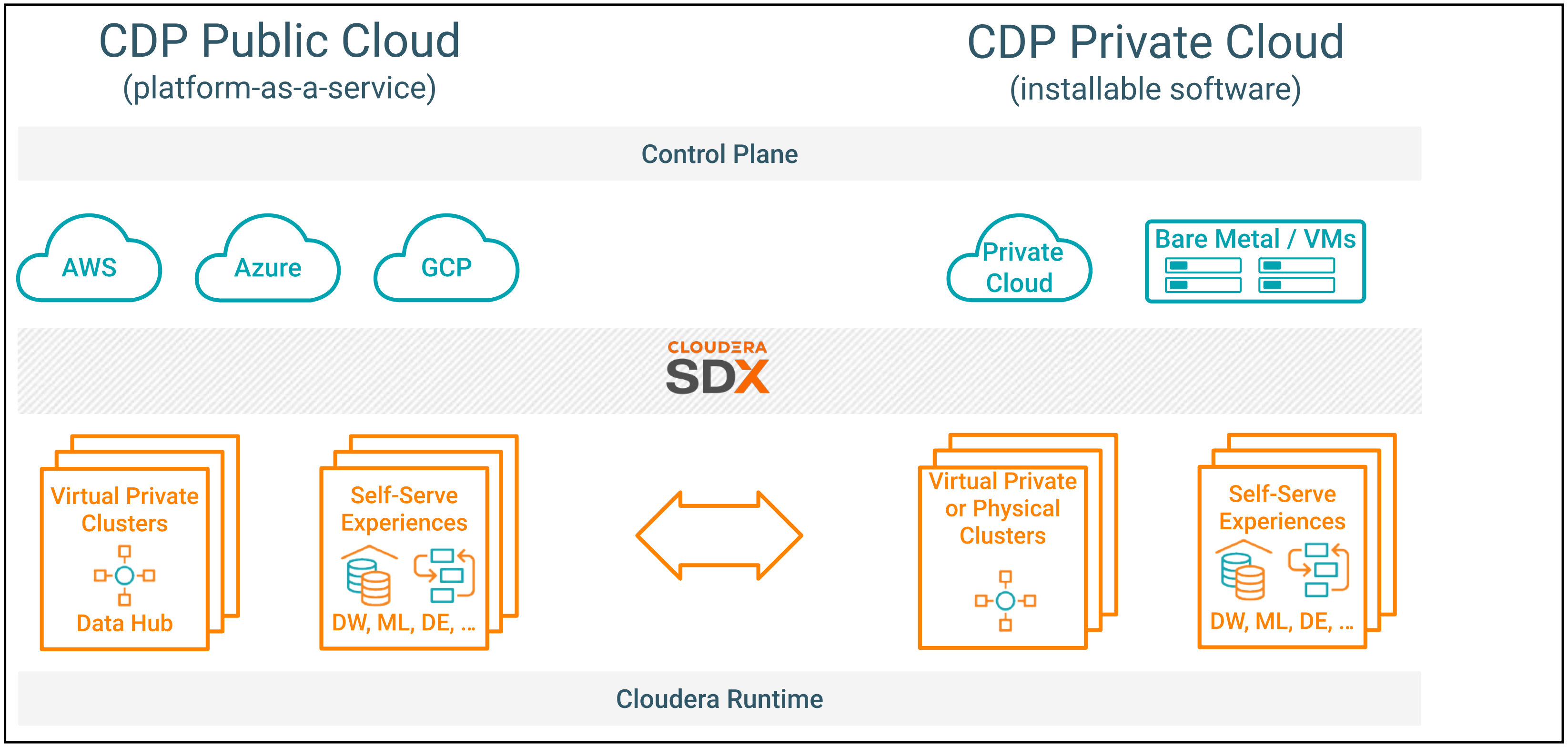
- **Workload Manager**
  - Analyze, optimize, and troubleshoot workloads
- **Management Console**
  - Manage environments, users, and services

Chapter Topics

Cloudera Data Platform

- Industry Trends for Big Data
- The Challenge to Become Data-Driven
- The Enterprise Data Cloud
- CDP Overview
- **CDP Form Factors**
- Essential Points
- Hands-On Exercise: Configure the Exercise Network

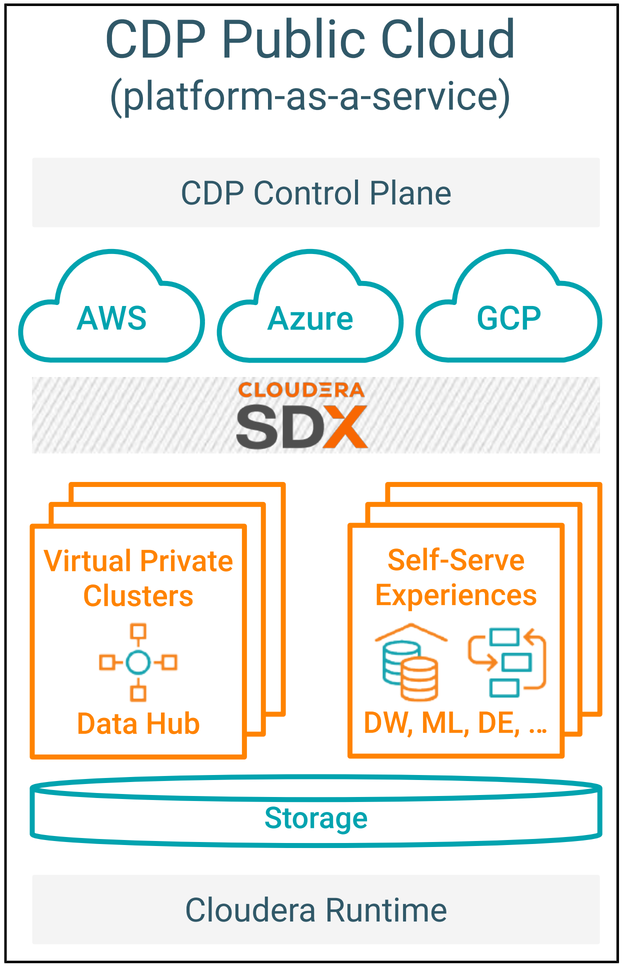
CDP Editions



- **CDP Public Cloud: Platform-as-a-Service (PaaS)**
- **CDP Private Cloud and CDP Private Cloud Base: Installable software**

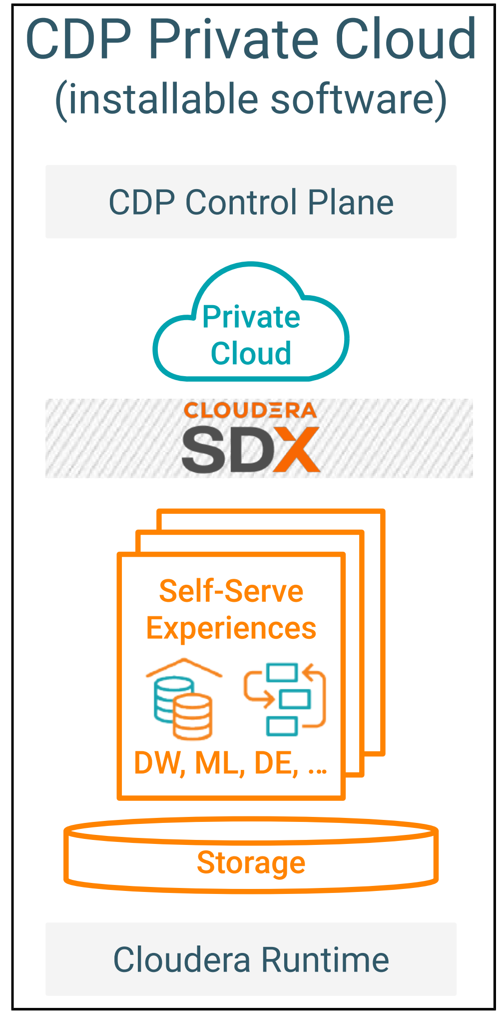
CDP Public Cloud

- **Common CDP components**
  - Control Plane
  - SDX Security and Governance
  - Cloudera Runtime
- **PaaS by Cloudera - no installation of common CDP components**
- **Cloud-native architecture with containers for self-service experiences**
  - Cloud provider storage used (S3, ADLS, GCS)
  - Separate compute and storage



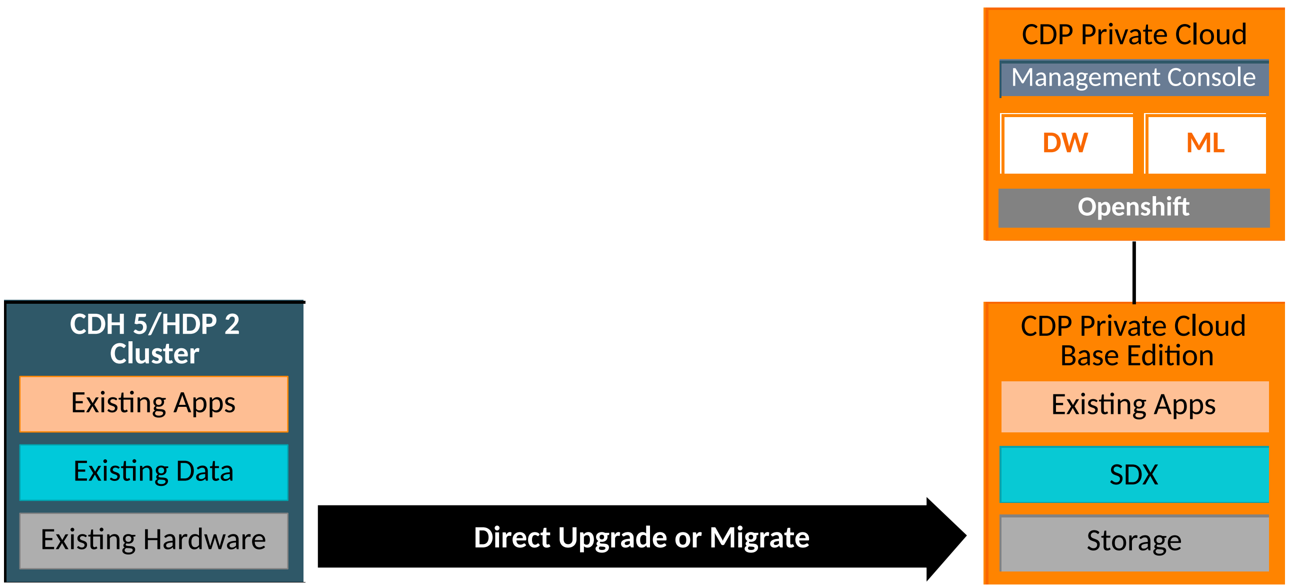
CDP Private Cloud with Private Cloud Base

- **Common CDP components**
  - Control Plane - Cloudera Manager
  - SDX Security and Governance
  - Cloudera Runtime
- **Software installed and managed by your own team**
- **Clusters are on bare metal hosts or VMs**
- **Object storage capability in tech preview**
- **Can burst on-prem workloads to the cloud as needed**



CDP PvC Base to Private Cloud

- **CDP PvC Base provides the storage and SDX for CDP Private Cloud**
  - Storage
  - Table Schema
  - Authentication and Authorization
  - Governance
- **Install PvC Base now, expand to new experiences soon**



Chapter Topics

Cloudera Data Platform

- Industry Trends for Big Data
- The Challenge to Become Data-Driven
- The Enterprise Data Cloud
- CDP Overview
- CDP Form Factors
- **Essential Points**
- Hands-On Exercise: Configure the Exercise Network

Essential Points (1)

- **Data is a critical business resource that must be properly managed and analyzed for success**
- **Architectural and technical trends include:**
  - Use of the cloud
  - Separation of compute and storage
  - Leveraging Kubernetes and containers
  - Use of real-time streaming and the move to ML and AI
  - The Cloudera Data Platform (CDP) is an implementation of the Enterprise Data Cloud
  - Hybrid and Multi-Cloud environments
  - SDX - common security and governance
  - Cloudera Runtime - open source distribution
  - Multi-Function Analytics
  - Control Plane services

Essential Points (2)

Architectural and technical trends also include:

- **Three Form Factors**
  - CDP Public Cloud (PC)
  - CDP Private Cloud (PvC)
  - CDP Private Cloud Base (PvC Base) - the focus of this course
- **Most organizations are not yet effectively data-driven**
  - The Enterprise Data Cloud is an architecture with essential characteristics meeting data-driven needs
  - The Cloudera Data Platform (CDP) is an *implementation* of the Enterprise Data Cloud

Bibliography

The following offer more information on topics discussed in this chapter

- **Big Data Management**
  - Solving the Pain Points of Big Data Management
  - How the Rise of Data and AI Have Redefined the Data Driven Enterprise
  - Cloudera Data Platform Editions
- **SDX**
- **Cloudera Data Hub**
- **Cloudera Data Warehouse**
- **Cloudera Machine Learning**
- **Enterprise Data Cloud**

Chapter Topics

Cloudera Data Platform

- Industry Trends for Big Data
- The Challenge to Become Data-Driven
- The Enterprise Data Cloud
- CDP Overview
- CDP Form Factors
- Essential Points
- **Hands-On Exercise: Configure the Exercise Network**

Hands-On Exercise: Configure the Exercise Network

- **In this exercise, you will ensure the network is prepared for your cluster installation**
- **Please refer to the Hands-On Exercise Manual for instructions**

CDP Private Cloud Base Installation

Chapter 3

Course Chapters

- Introduction
- Cloudera Data Platform
- **CDP Private Cloud Base Installation**
- Cluster Configuration
- Data Storage
- Data Ingest
- Data Flow
- Data Access and Discovery
- Data Compute
- Managing Resources
- Planning Your Cluster
- Advanced Cluster Configuration
- Cluster Maintenance
- Cluster Monitoring
- Cluster Troubleshooting
- Security
- Private Cloud / Public Cloud
- Conclusion
- Appendix: Cloudera Manager API
- Appendix: Ozone Overview

CDP Private Cloud Base Installation

After completing this chapter, you will be able to

- **Describe the requirements for CDP Private Cloud Base installation**
- **Describe the steps for a CDP installation**
- **Describe the features of CDP**
- **View Cloudera Manager and Runtime installation**
- **Review Cloudera Manager**

Chapter Topics

CDP Private Cloud Base Installation

- **Installation Overview**
- Cloudera Manager Installation
- Hands-On Exercise: Installing Cloudera Manager Server
- CDP Runtime Overview
- Cloudera Manager Introduction
- Essential Points
- Instructor-Led Demonstration: Cloudera Manager
- Hands-On Exercise: Cluster Installation

CDP Private Cloud Base Requirements

- Hardware Requirements
- Operating System Requirements
- Database Requirements
- Java Requirements
- Networking and Security Requirements

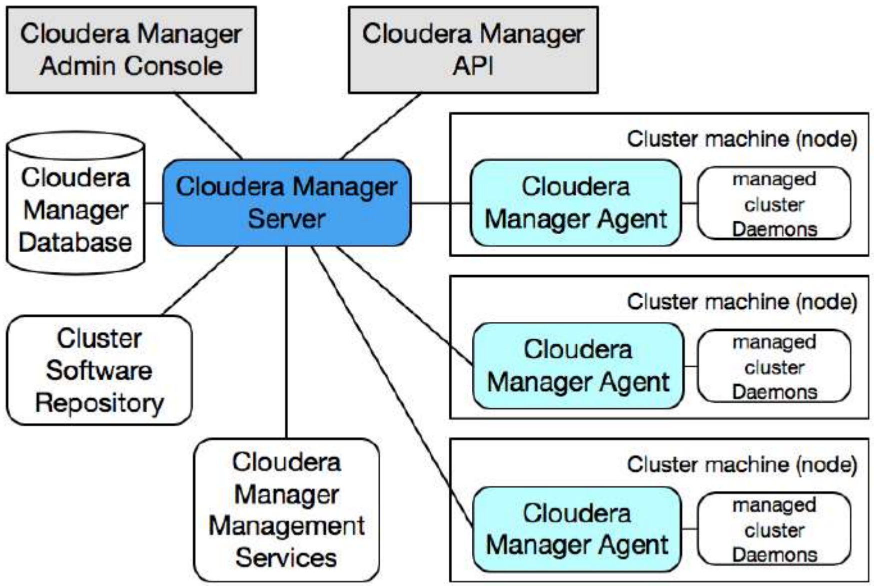
Hardware Requirements

- **To assess the hardware and resource allocations for your cluster consider the following:**
  - Types of workloads to be supported
  - Runtime components needed
  - Size of the data to be stored and processed
  - Frequency of the workloads
  - Number of concurent jobs that need to run
  - Speed required for your applications
- **Goal: Allocate Cloudera Manager and Runtime roles among the hosts to maximize your use of available resources.**

Managing Multiple Hosts in a Cluster

- **Use automated OS deployment tools for commissioning many hosts**
  - Example tools: Red Hat's Kickstart, Dell Crowbar
  - These tools are optional
- **Use configuration management tools to manage the underlying OS**
  - Example tools: Puppet, Chef, Ansible
  - These tools are also optional
- **Use Cloudera Manager to install and manage cluster software**

Basic Cloudera Manager Topology



- **Cloudera Manager Server**
  - Stores cluster configuration information in a database
  - Sends configurations and commands to agents over HTTP(S)
  - Receives heartbeats every 15 seconds from all agents
  - Accessible from the Admin Console and by way of the API

- **Cloudera Manager Agents**
  - Installed on every managed host
  - Receive updated configurations from server
  - Start and stop cluster daemons, collect statistics
  - Send status heartbeats to the server

Operating System Requirements

- **CDP Private Cloud Base Supported Operating System**
  - Red Hat Enterprise Linux/CentOS 7.6, 7.7, 7.8
  - Oracle Enterprise Linux 7.6, 7.7, 7.8
- **In order to be covered by Cloudera Support:**
  - All Runtime Hosts must run the same major OS release
  - Mixed OS configuration only supported during upgrade project
  - Cloudera Manager must run on the same OS release as one of the clusters it manages
  - Cloudera does not support Runtime cluster deployments in Docker containers

JDK and Database Requirements

- **All hosts must run the same major JDK version**
- **CDP Private Cloud Base JDK 8 Versions**
  - CDP 7.0: OpenJDK 1.8 and Oracle JDK 1.8
  - CDP 7.1: OpenJDK 1.8, Open JDK 11 and Oracle JDK 1.8
  - Only 64-bit JDKs are supported
- **Supported databases**
  - MySQL 5.7
  - MariaDB 10.2
  - Oracle 12 (new installations only)
  - PostgreSQL 10
- **See** `Database Requirements` **for version-specific details**

Additional CDP Requirements

- **See release notes for detailed requirements for factors such as:**
  - Filesystems: ext3, ext4, XFS, S3
  - Network configuration - TLS
  - OS configuration
  - CDP requires IPv4
  - IPv6 is not supported and must be disabled
  - All hosts must have a working network name resolution system
  - Third-party installations
    - Python 2.7 or higher; Python 3 is not supported
    - Perl is required by Cloudera Manager
    - iproute package is required for CDP Private Cloud Base
  - See installation guide for more details

Chapter Topics

CDP Private Cloud Base Installation

- Installation Overview
- **Cloudera Manager Installation**
- Hands-On Exercise: Installing Cloudera Manager Server
- CDP Runtime Overview
- Cloudera Manager Introduction
- Essential Points
- Instructor-Led Demonstration: Cloudera Manager
- Hands-On Exercise: Cluster Installation

Cloudera Manager: Before You Install

- **Pre-install considerations**
  - Storage space planning
  - Configure network names (DNS or `/etc/hosts`)
  - Disable firewall for each host in the cluster
  - Temporarily disable SELinux
  - Enable an NTP service
- **Configure parcel repositories for both:**
  - Cloudera Manager
  - CDP Private Cloud Base

CDP Installation Steps

- **Step 1: Configure a repository for Cloudera Manager**
- **Step 2: Install JDK if needed**
- **Step 3: Install Cloudera Manager Server**
- **Step 4: Install and configure databases**
- **Step 5: Setup Cloudera Manager Database**
- **Step 6: Install Runtime and other software**
- **Step 7: Set up a cluster using the wizard**

Configure a Repository

- **Cloudera maintains Internet-accessible repositories for Runtime and Cloudera Manager installation files**
- **Create an internal repository for hosts that do not have Internet access**
- **To use the Cloudera repository:**
  - Download the repository file on the Cloudera Manager server host

```
https://[username]:[password]@archive.cloudera.com/p/
cm7/7.1.3/redhat7/yum/cloudera-manager.repo
```

  - Move the cloudera-manager.repo file to the /etc/yum.repos.d/ directory
  - Edit the repository file and add your username and password (`changeme`)
  - Import the repository signing GPG key: (RHEL 7)

```
sudo rpm --import https://[username]:
[password]@archive.cloudera.com/p/\cm7/7.1.3/redhat7/yum/
RPM-GPG-KEY-cloudera
```

Install JDK

- **There are several options for installing a JDK on your CDP Private Cloud Base hosts:**
  - Install OpenJDK 8 on the Cloudera Manager server host and then allow Cloudera Manager to install OpenJDK 8 on its managed hosts
  - Manually install a supported JDK on all cluster hosts
- **Supported JDKs for CDP Private Cloud Base 7.1**
  - OpenJDK 1.8
  - OpenJDK 11
  - Oracle JDK 1.8
- **The JDK must be 64-bit**
- **The same version of the JDK must be installed on each cluster host**
- **The JDK must be installed at `/usr/java/jdk-version`**

Install Cloudera Manager Server

- **Install Cloudera Manager Packages**
  - Type the command to install
  - RHEL install command:
    `sudo yum install cloudera-manager-daemons cloudera-manager-server`
- **Enable Auto-TLS (recommended)**
  - Auto-TLS automates the creation of an internal certificate authority (CA)
  - When you start CM with TLS enabled, all hosts and services will automatically have TLS configured and enabled
  - You can enable auto-TLS on existing clusters or during install
- **Alternative: Use an existing Certificate Authority (CA)**

Install and Configure Databases

- **Required databases are based on the services installed, such as:**
  - Cloudera Manager Server
  - Oozie Server
  - Reports Manager
  - Hive Metastore Server
  - Hue Server
  - Ranger
  - Schema Registry
  - Streams Messaging Manager
- **Command to setup Cloudera Manager databases:**

```
$ sudo /opt/cloudera/cm/schema/scm_prepare_database.sh
```

Install Runtime

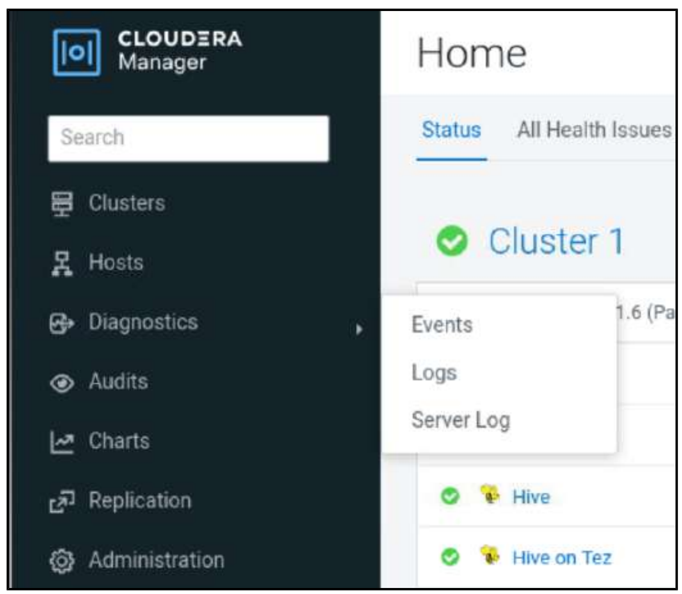
- **Start Cloudera Manager Server**
  - RHEL 7: `sudo service cloudera-scm-server start`
- **In a web browser, go to `http://server_host:7180`**
- **If you enabled auto-TLS, you are redirected to `https://server_host:7183`**
- **Log into Cloudera Manager Admin Console:**
  - Username: `admin`
  - Password: `admin`
- **Installation wizard will start**

Set Up a Cluster Using the Wizard

- Select services
- Assign roles
- Setup databases
- Enter required parameters
- Review changes
- Command Details page lists details of the First Run command
- Summary Page reports success or failure of set up

Cloudera Manager Log Locations

- **Check log files for install details**
- **On the Cloudera Manager Server host**
  - `/var/log/cloudera-scm-server/cloudera-scm-server.log`
- **Agent logs on other hosts in the cluster**
  - `/var/log/cloudera-scm-agent/cloudera-scm-agent.log`
- **The Cloudera Manager web UI also provides log access**



Chapter Topics

CDP Private Cloud Base Installation

- Installation Overview
- Cloudera Manager Installation
- **Hands-On Exercise: Installing Cloudera Manager Server**
- CDP Runtime Overview
- Cloudera Manager Introduction
- Essential Points
- Instructor-Led Demonstration: Cloudera Manager
- Hands-On Exercise: Cluster Installation

Hands-On Exercise: Installing Cloudera Manager Server

- **In this exercise, you will install Cloudera Manager**
- **Please refer to the Hands-On Exercise Manual for instructions**

Chapter Topics

CDP Private Cloud Base Installation

- Installation Overview
- Cloudera Manager Installation
- Hands-On Exercise: Installing Cloudera Manager Server
- **CDP Runtime Overview**
- Cloudera Manager Introduction
- Essential Points
- Instructor-Led Demonstration: Cloudera Manager
- Hands-On Exercise: Cluster Installation

Runtime Overview

- **CDP Runtime includes components to manage and analyze data**
- **Cloudera Runtime is:**
  - The core open source software distribution within CDP
  - Includes data management tools within CDP
  - Maintained, supported, versioned, and packaged as a single entity
  - Includes 50 open source projects
- **Runtime does NOT include:**
  - Cloud Services: Data Hub, DWX, and MLX
  - Management Console, Workload Manager, Replication Manager
  - Data Catalog
  - Add-on products such as CDSW, CDF, and Metron

Cloudera Data Platform Features (1)

- **CDP Private Cloud Base includes the following components "out of the box":**
  - Apache Hadoop: Distributed batch processing of large data sets
  - Apache HBase: Database for structured data storage of large tables
  - Apache Hive: Data warehouse summarization and ad hoc querying
  - Hive Metastore (HMS): Metadata store for Hive tables
  - Apache Oozie: Workflow scheduler to manage Hadoop jobs
  - Apache Parquet: Columnar storage format for Hadoop ecosystem
  - Apache Spark: Fast compute engine for ETL, ML, stream processing
  - Apache Sqoop: Bulk data between Hadoop and structured datastores
  - YARN: Job scheduling and cluster resource management
  - Apache ZooKeeper: Coordination service for distributed applications
  - Apache Atlas: Metadata management, governance and data catalog
  - Apache Hadoop Ozone: Distributed object store for Hadoop (in Tech Preview)

Cloudera Data Platform (2)

- **CDP Data Private Cloud Base includes the following components "out of the box":**
  - Apache Phoenix: OLTP and real-time SQL access of large dataset
  - Apache Ranger: Manage data security across the Hadoop ecosystem
  - Apache ORC: Smallest, fastest columnar storage for Hadoop
  - Apache Tez: Data-flow framework for batch, interactive use-cases
  - Apache Avro: Data serialization system
  - Cloudera Manager: Manage and control Hadoop ecosystem functions
  - Hue: SQL workbench for data warehouses
  - Apache Impala: Distributed MPP SQL query engine for Hadoop
  - Apache Kudu: Column-oriented data store for fast data analytics
  - Apache Solr: Enterprise search platform
  - Apache Kafka: Real-time streaming data pipelines and apps
  - Ranger KMS: Cryptographic key management

Chapter Topics

CDP Private Cloud Base Installation

- Installation Overview
- Cloudera Manager Installation
- Hands-On Exercise: Installing Cloudera Manager Server
- CDP Runtime Overview
- **Cloudera Manager Introduction**
- Essential Points
- Instructor-Led Demonstration: Cloudera Manager
- Hands-On Exercise: Cluster Installation

What is Cloudera Manager?

- **Web application used by administrators to:**
  - Add a cluster(s)
  - Start and stop the cluster
  - Start and stop services
  - Manage hosts
  - Manage cluster resources
  - Monitoring cluster(s)
  - Configure cluster(s)
  - Manage users and security
  - Upgrade the cluster

Cluster State Management

- **The Cloudera Manager Server maintains the state of the cluster**
- **This state can be divided into two categories: "model" and "runtime", both of which are stored in the Cloudera Manager Server database**
  - Model state:
    - Captures what is supposed to run, where, and with what configuration
    - For example: It knows there are 15 hosts and each run the datanode daemon
    - When you update a configuration, you update the model state
  - Runtime state:
    - Captures the processes, where and what commands are currently running
    - Includes the exact configuration files needed to run a process

Service Roles in CDP

- **Specific roles need to be assigned to provide services within a cluster**
- **Daemons run on the hosts to play the roles assigned**
- **Roles needed are based on the service, for example:**
  - HDFS: namenode, datanode
  - YARN: resource manager, node manager
  - Impala: impala daemons, catalog server, state store

Types of Hosts: Master

- **Master hosts**
  - For roles that coordinate or manage a specific service on the cluster
- **Some typical service roles**
  - HDFS NameNode
  - YARN ResourceManager
  - Job history servers
  - ZooKeeper

Types of Hosts: Worker

- **Worker hosts**
  - For roles that do the distributed work for a specific service on the cluster
- **Some typical service roles**
  - HDFS DataNode
  - YARN NodeManager
  - Impala daemons

Types of Cluster Hosts: Gateway

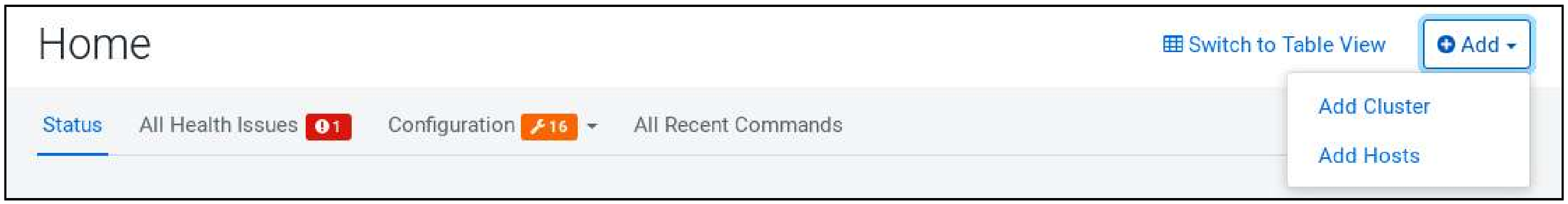
- **Gateway hosts**
  - Act as gateway between the rest of the network and the cluster
  - Provide runtime environment for applications that require cluster services
    - Such as Spark and MapReduce applications
  - Do not run service daemons
  - Also known as edge hosts
- **Some typical roles**
  - HDFS Gateway
  - YARN Gateway
  - Hive Gateway
  - Spark 2 Gateway
  - Hue Server
  - HiveServer2

Distribute the Daemons

- **The Installation wizard suggests Hadoop daemon host assignments**
  - Based on available resources
  - Easy to override suggestions
- **Detailed recommendations for every CDP service role**
  - Recommended Cluster Host and Role Distribution documentation

Multiple Cluster Management

- **One instance of Cloudera Manager Private Cloud can manage multiple clusters**
- **Cloudera Manager allows for adding and deleting clusters**
- **From the Cloudera Manager Home page**
  - Select the Add button on the top right
  - Choose Add Cluster

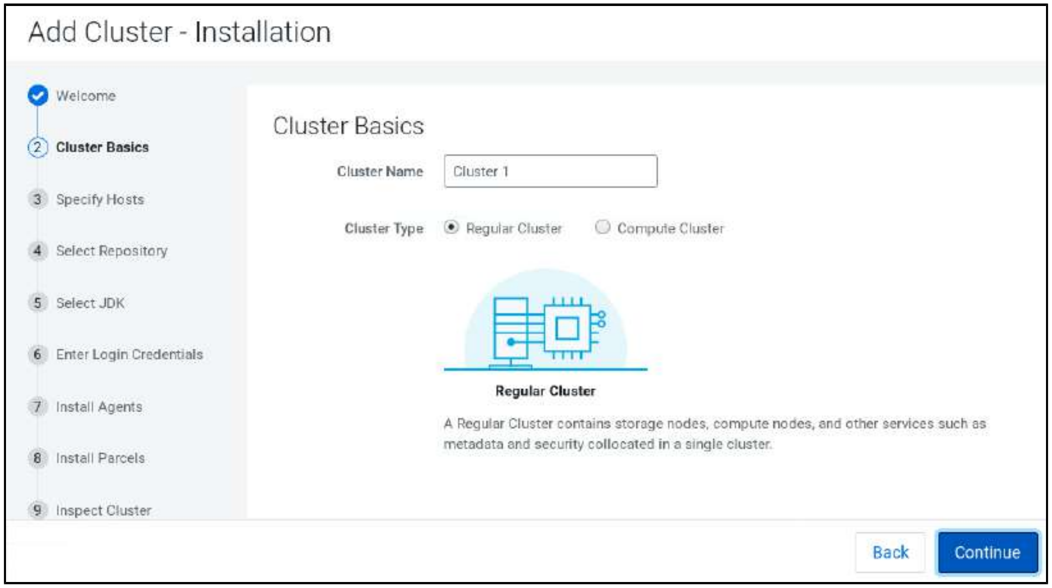


Add a Cluster (1)

- **Adding a cluster consists of two steps:**
  - Add a set of hosts and install Cloudera Runtime and the CM Agent
  - Select and configure the services to run on the new cluster

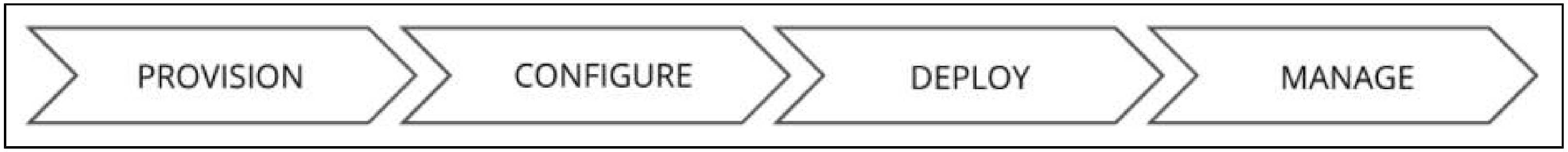
Add a Cluster (2)

- **Enter Cluster Basics:**
  - Cluster Name
  - Cluster Types
    - Regular Cluster: contains storage nodes, compute nodes and other services
    - Compute Cluster: consists of only compute nodes. You must connect to existing storage, metadata or security services



Usage of Provisioning Tools

- **Managing the provisioning of servers can be very costly**
- **Many use a product like Ansible, Puppet, or Chef to automate provisioning, configuration management, and application deployment**
- **Ansible is an open-source tool**



Ansible and CDP

- **RedHat's enterprise configuration and orchestration engine**
  - Installs required OS packages and configs
  - Installs and/or prepares supporting infra (databases, kerberos, TLS)
  - Deploys CDP Private Cloud Base clusters
  - Enables security, encryption and high availability
  - Schedules CDP to spin up or down Data Hubs and Experiences
  - Adds local configurations, datasets, and applications

Chapter Topics

CDP Private Cloud Base Installation

- Installation Overview
- Cloudera Manager Installation
- Hands-On Exercise: Installing Cloudera Manager Server
- CDP Runtime Overview
- Cloudera Manager Introduction
- **Essential Points**
- Instructor-Led Demonstration: Cloudera Manager
- Hands-On Exercise: Cluster Installation

Essential Points

- **Prepare cluster hosts for Cloudera installation**
  - Install database and third-party packages, configure OS and networking
  - See release notes for details
- **Install Cloudera Manager first**
  - Then use Cloudera Manager to create cluster
- **Create a cluster**
  - Add hosts
  - Select services and assign roles
  - Cloudera Manager automatically deploys the software using the CM agents

Chapter Topics

CDP Private Cloud Base Installation

- Installation Overview
- Cloudera Manager Installation
- Hands-On Exercise: Installing Cloudera Manager Server
- CDP Runtime Overview
- Cloudera Manager Introduction
- Essential Points
- **Instructor-Led Demonstration: Cloudera Manager**
- Hands-On Exercise: Cluster Installation

Instructor-Led Demonstration of the Cloudera Manager UI

- **The instructor will demonstrate how to navigate the Cloudera Manager UI**
- **This demonstration will introduce key pages in Cloudera Manager**

Chapter Topics

CDP Private Cloud Base Installation

- Installation Overview
- Cloudera Manager Installation
- Hands-On Exercise: Installing Cloudera Manager Server
- CDP Runtime Overview
- Cloudera Manager Introduction
- Essential Points
- Instructor-Led Demonstration: Cloudera Manager
- **Hands-On Exercise: Cluster Installation**

Hands-On Exercise: Cluster Installation

- **In this exercise, you will install a cluster**
- **Please refer to the Hands-On Exercise Manual for instructions**
- **Note that you installed Cloudera Manager version 7.1.3, and will be installing a cluster with 7.1.2**
- **You will be upgrading the cluster to 7.1.3 later in the course**

Cluster Configuration

Chapter 4

Course Chapters

Cluster Configuration

After completing this chapter, you will be able to

- **Set and modify configuration settings**
- **Add code snippets to perform advanced configuration**
- **Resolve stale configurations**
- **Describe how configuration properties inherit values**
- **Install and deploy add-on services**
- **Create a host template and use it to apply service roles to a host in the cluster**
- **Add new hosts to a cluster**

Chapter Topics

Cluster Configuration

- **Overview**
- Configuration Settings
- Modifying Service Configurations
- Configuration Files
- Managing Role Instances
- Adding New Services
- Adding and Removing Hosts
- Essential Points
- Hands-On Exercise: Configuring a Hadoop Cluster

Configuring Clusters

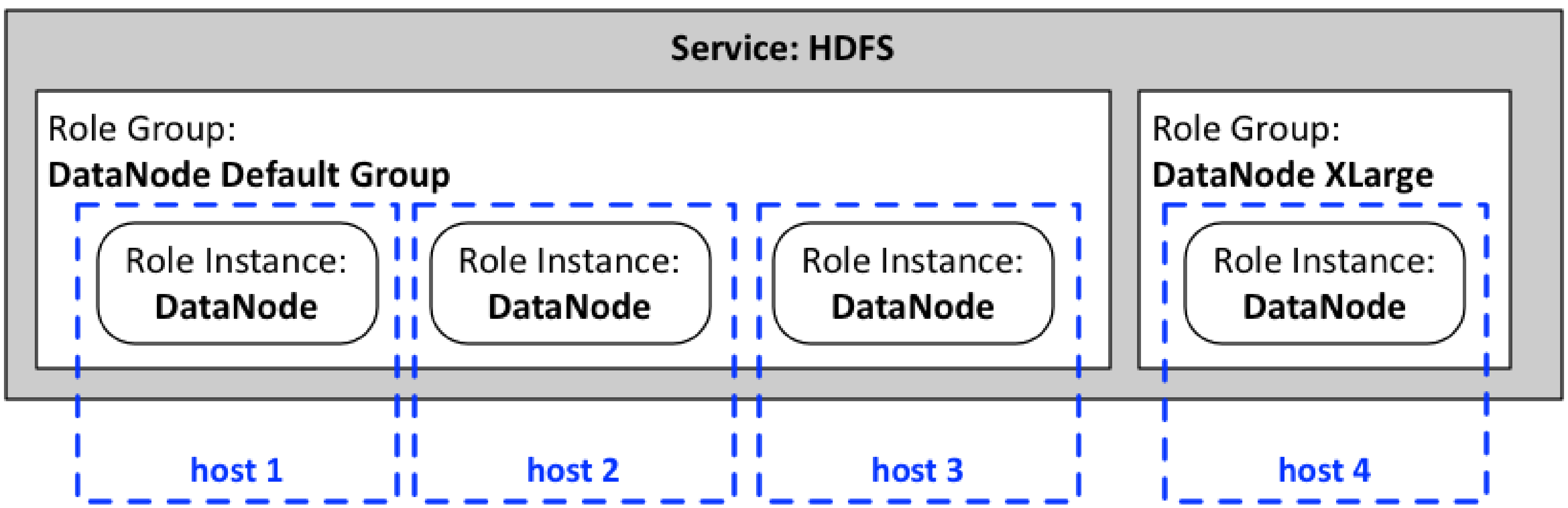
- **Over time, you will occasionally need to change the cluster configuration**
- **Common examples**
  - Adjust settings of an existing service
  - Add more services to a cluster
  - Add new hosts to a cluster
  - Remove hosts from a cluster
- **Cloudera Manager streamlines configuration changes**

Cloudera Manager—Configuration Terminology (1)

- **Service**: A set of related components providing a type of Hadoop functionality on a cluster
  - Examples: HDFS, YARN, Spark, Hive
- **Role**: An individual component of a service
  - Examples: HDFS NameNode and DataNode, YARN NodeManager and ResourceManager

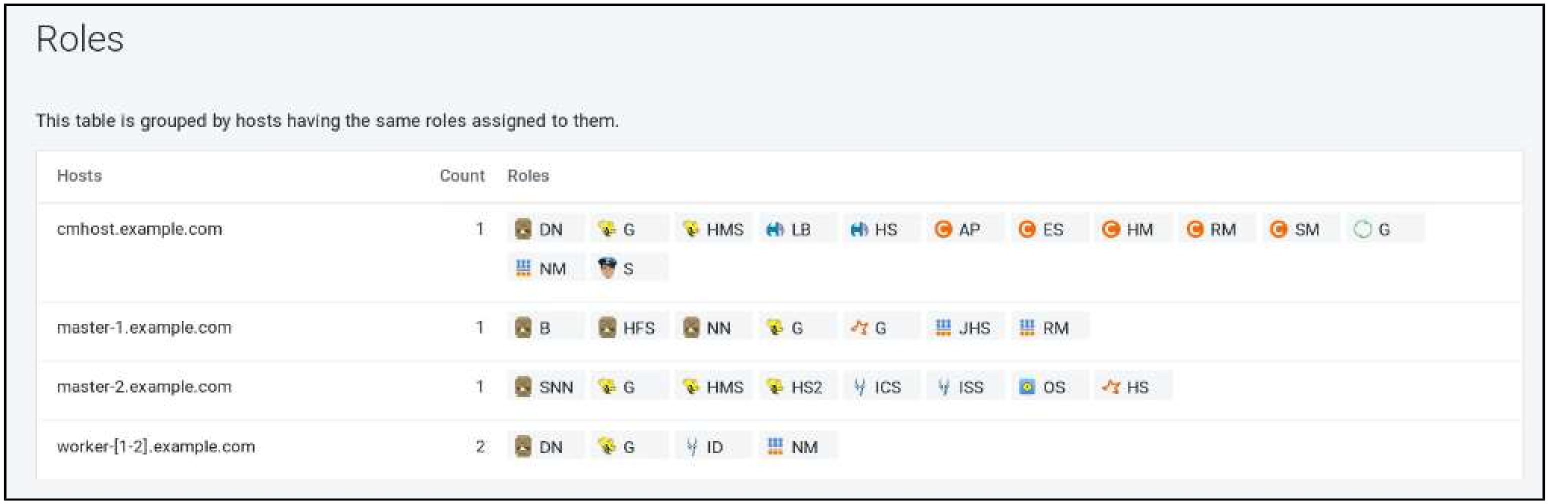
Cloudera Manager—Configuration Terminology (2)

- **Role Instance**: Usually a daemon running on a specific host
  - Example: A DataNode daemon running on a specific host
  - For gateway roles, usually a set of service client files and/or configuration sets instead
- **Role Group**: A group of roles distributed across multiple hosts with the same configuration
  - Example: A set of DataNode roles with default settings



Host Role Assignments

- **Select** Hosts > Roles **to see roles assigned to each host**



Roles

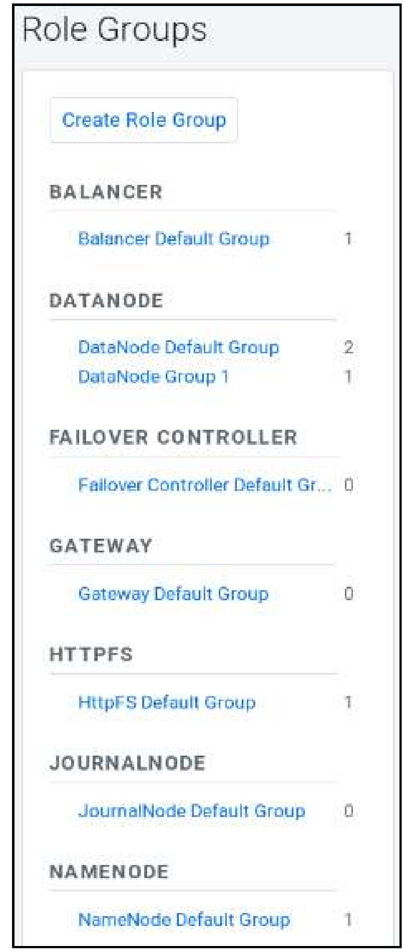
This table is grouped by hosts having the same roles assigned to them.

Hosts	Count	Roles
cmhost.example.com	1	DN G HMS LB HS AP ES HM RM SM G NM S
master-1.example.com	1	B HFS NN G G JHS RM
master-2.example.com	1	SNN G HMS HS2 ICS ISS OS HS
worker-[1-2].example.com	2	DN G ID NM

Role Groups

- **Group role instances that will be configured the same way**
  - This eases configuration management
- **Create new role groups as needed**
  - Cloudera Manager creates some automatically when services are added
  - You can add additional groups manually
  - A role instance can be moved to a different group
- **Example: Configure role instances on hosts with similar hardware**

HDFS Role Groups



Chapter Topics

Cluster Configuration

- Overview
- **Configuration Settings**
- Modifying Service Configurations
- Configuration Files
- Managing Role Instances
- Adding New Services
- Adding and Removing Hosts
- Essential Points
- Hands-On Exercise: Configuring a Hadoop Cluster

Initial Settings

- **All configuration properties have default settings**
  - Some are generic Apache values
  - Some are recommendations based on Cloudera's experience
- **Cloudera Manager auto-configuration sets *initial values* for role groups when cluster is created**
  - Usually uses defaults
  - May override defaults based on resources available in the cluster
    - For example, memory and CPU cores

Configuration Levels

- **You can override initial property settings**
- **Properties can be set at different levels**
  - Service
  - Role Group
  - Role Instance
- **Settings are inherited from higher levels**

Inheritance and Precedence

- **Configuration inheritance order of priority (lowest to highest)**
  - Service > Role Group > Role Instance
- **Role group settings override service level settings**
  - Example: Some hosts have more storage capacity than others
    - Configure an `XLarge` role group for larger hosts
    - Set a higher storage capacity than the service default
    - Add the larger hosts' DataNode roles to the role group
- **Role instance settings override role group and service settings**
  - Example: Enable verbose logging for a role instance while troubleshooting that instance
- **To indicate that the property value cannot be overridden, select the *final* checkbox**

Chapter Topics

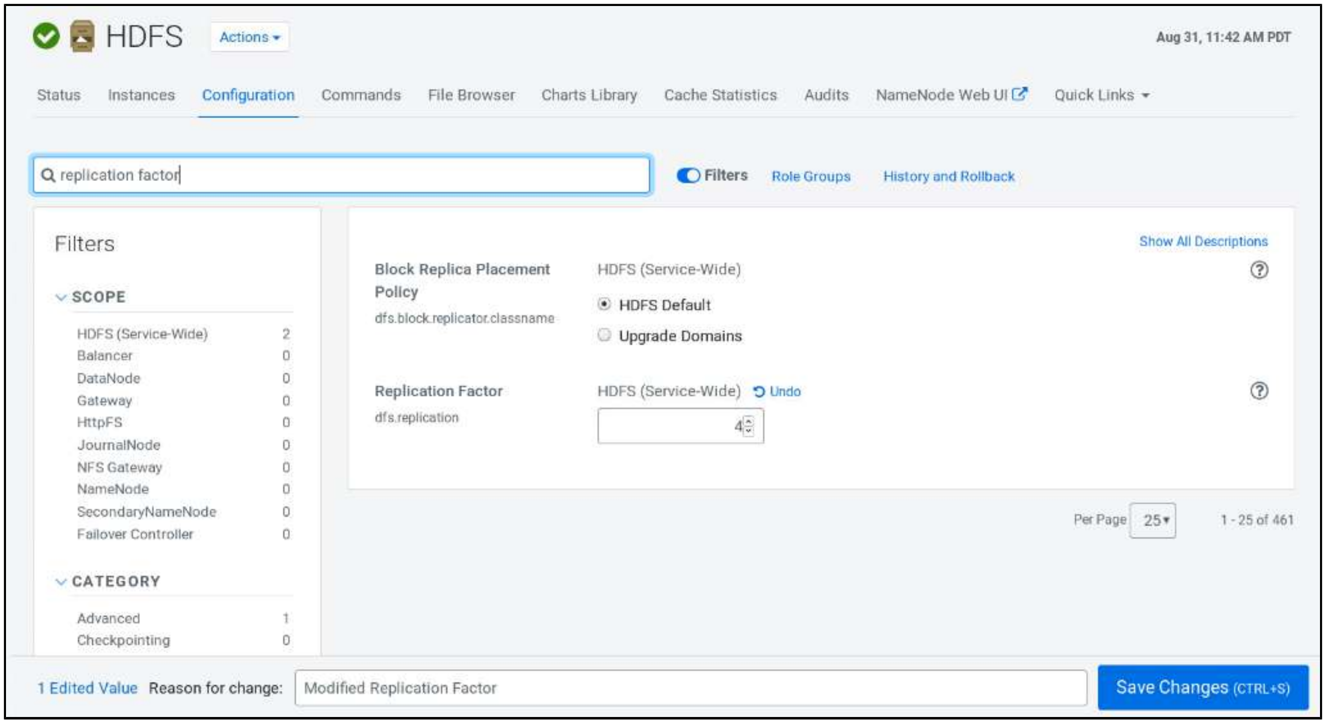
Cluster Configuration

Cluster Configuration

- **Over 1800 cluster and service properties can be configured**
- **Most common properties are exposed in Cloudera Manager**
  - Set in the configuration page for a cluster, service, role instance, or host
- **Cloudera Manager hides some properties**
  - Only the most commonly reconfigured properties are exposed
  - Discourages potentially dangerous settings changes
- **Hidden properties use the Cloudera default value**
  - Set within the service parcel or package JAR file

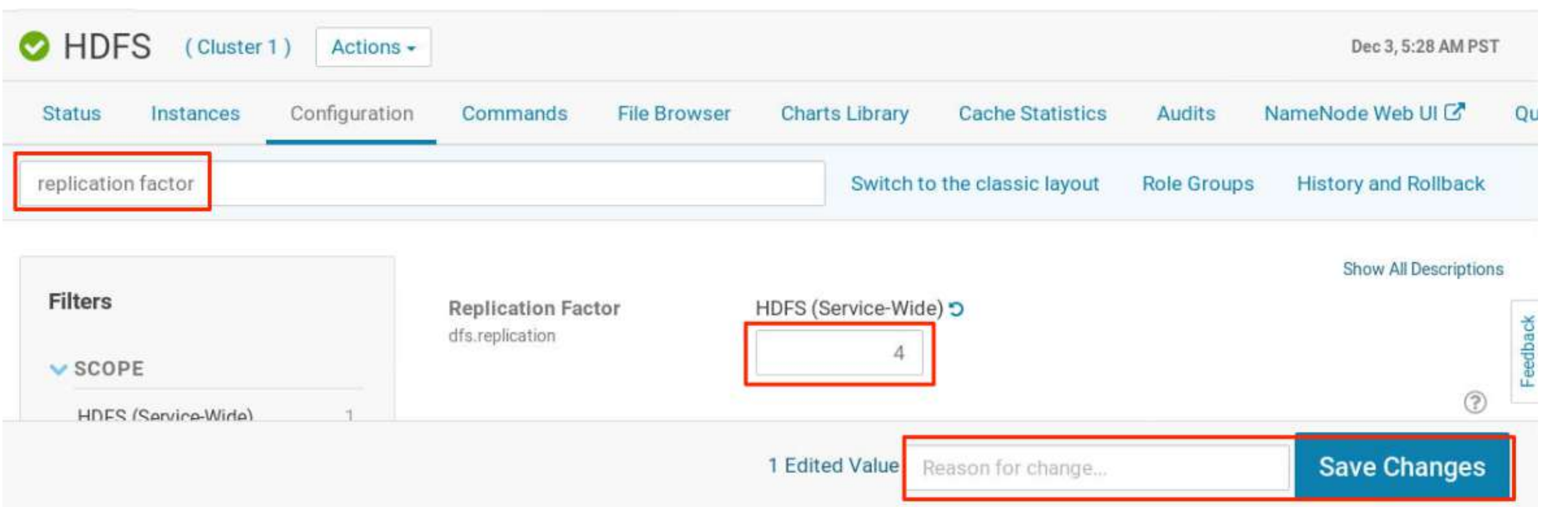
Locating a Configuration Property

- **A few ways to find a property in Cloudera Manager**
  - Global search box
  - Search box on a specific service's Configuration page
  - Scope and Category filters for services



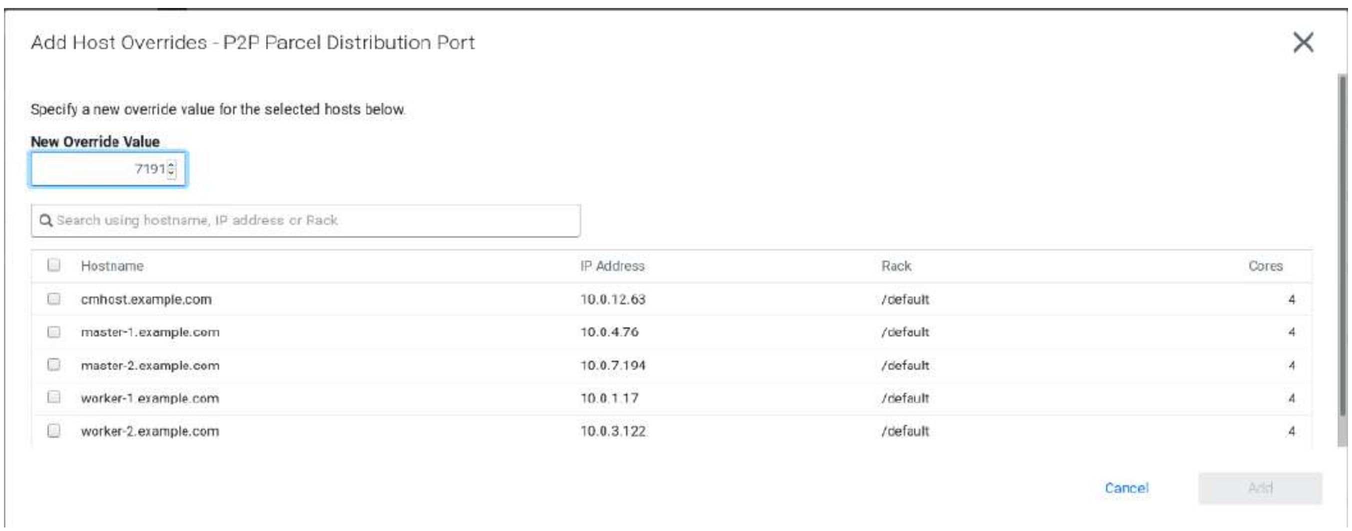
Making a Configuration Change

- **Locate the configuration you want to change**
- **Modify value, optionally add a reason for the change, and click Save Changes**
- **The blue arrow icon indicates the current value is not the default**
  - Click it to reset to the default value



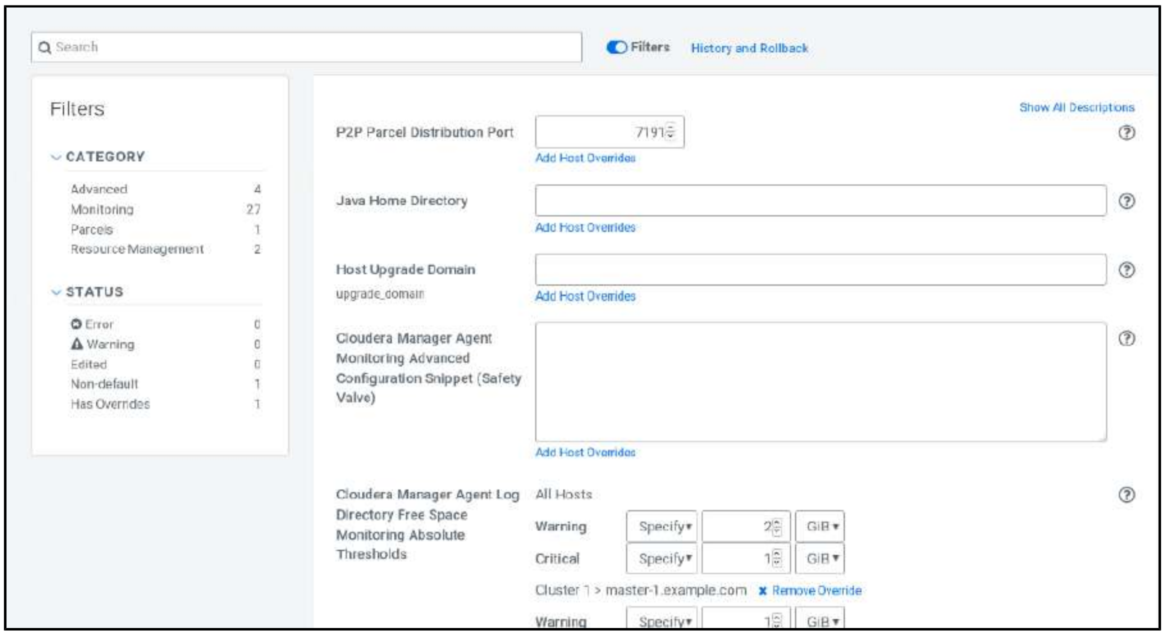
Host Overrides

- **You can override the properties of individual hosts in your cluster**
  - Click **Hosts > Hosts Configuration**
  - Use the Filters or Search box to locate the property that you want to override
  - Click Add **Host Overrides** link
  - Edit the overrides and **Save** the changes



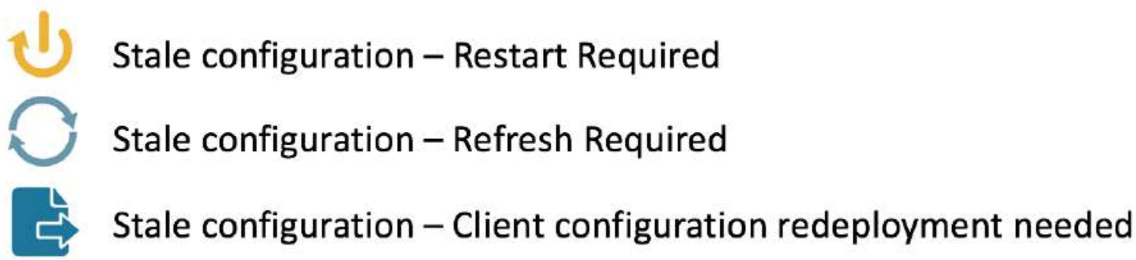
Viewing Host Overrides

- **List of all role instances with an override value**
  - **Configuration** tab on the service or host page
  - Select from **Filters > Status > Has overrides**
  - List of properties that have been overridden displays
  - Click on the **X** icon to remove overridden value



Stale Configurations

- **A modified configuration may require one or both of the following**
  - Client redeployment
  - Role instance restart or refresh
- **Cloudera Manager prompts you to resolve outdated configurations**
- **The affected service(s) will display one or more of the icons below**
- **Click the icon to display the Stale Configurations page**

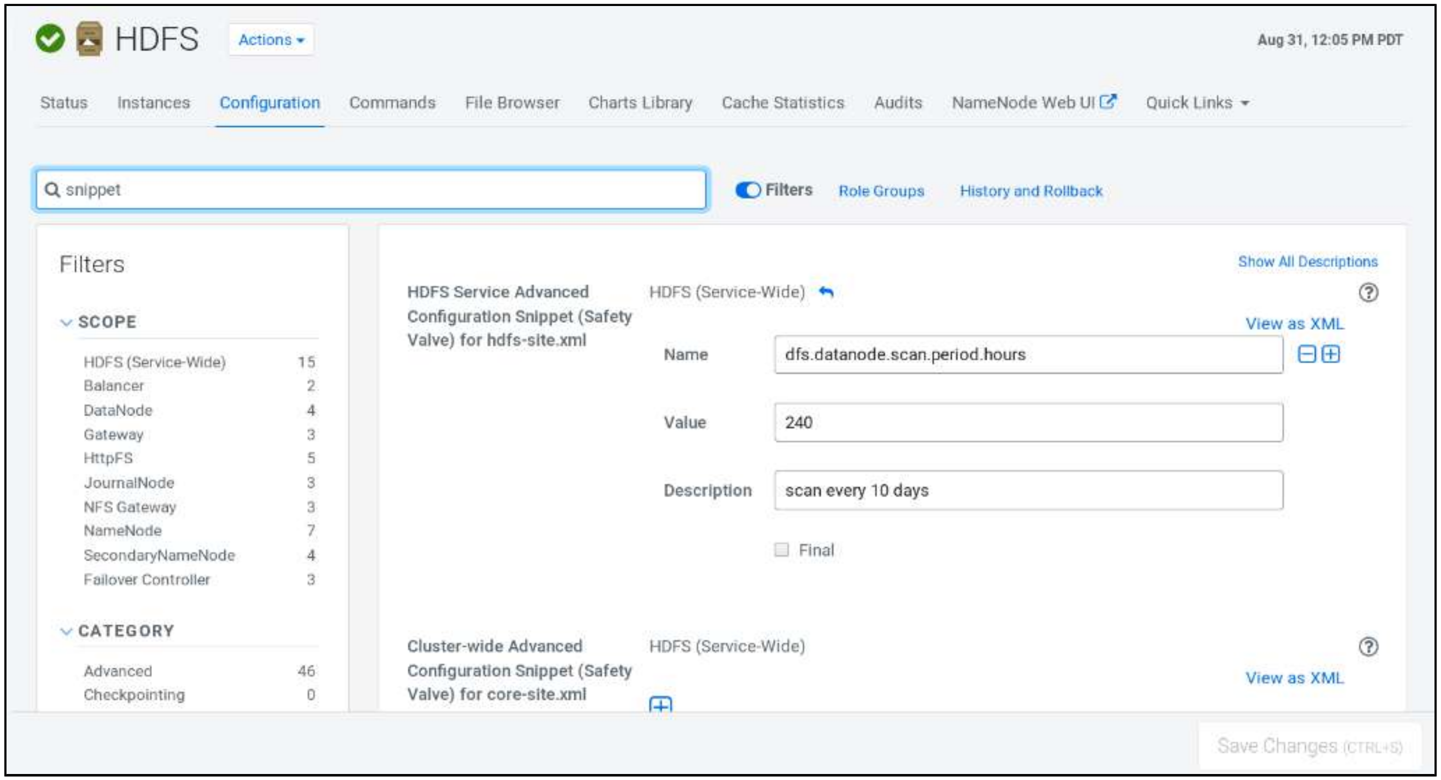


Applying a Configuration Change

- **Review changes on Cloudera Manager Stale Configurations page**
  - Shows which changes will be applied to which role instance(s)
  - Prompts you to redeploy, refresh, or restart as necessary

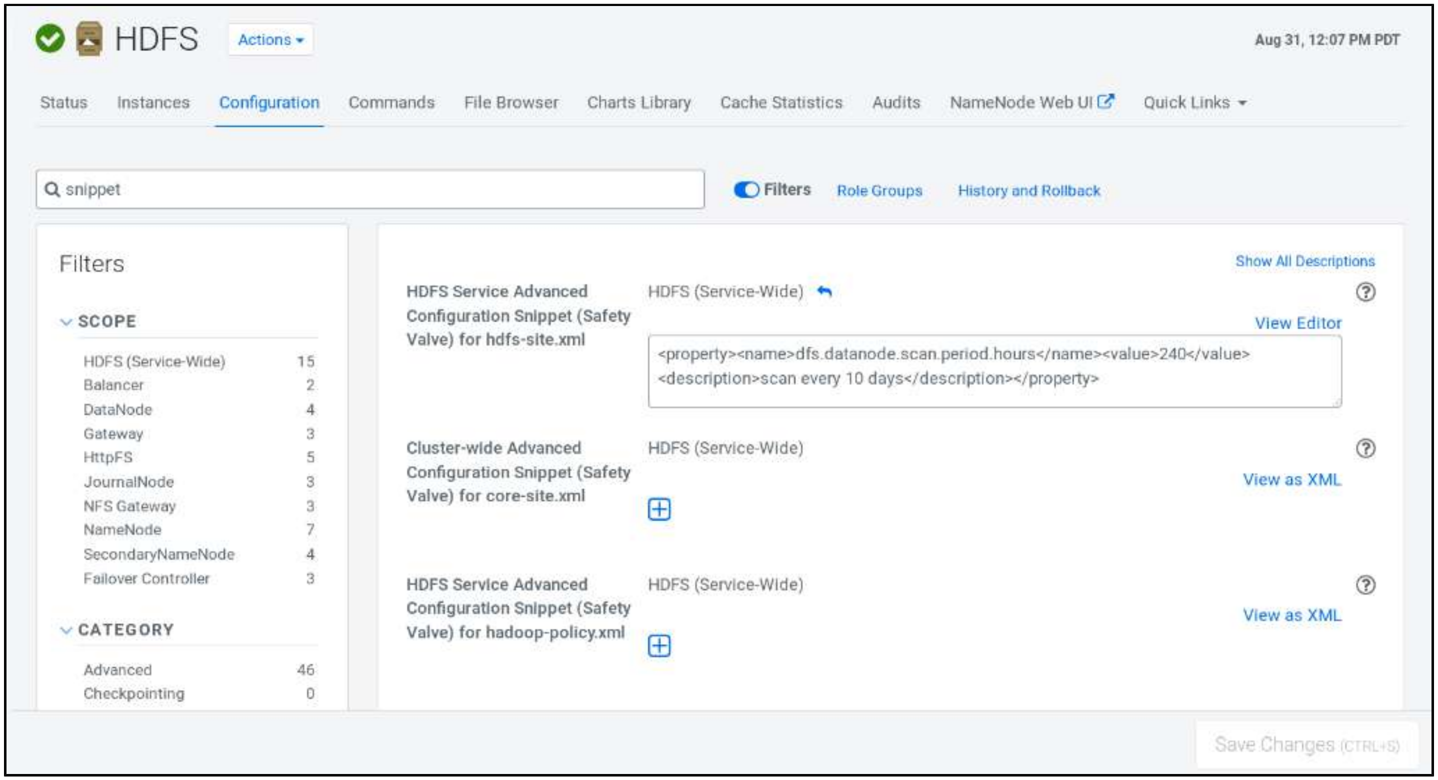
Setting or Modifying Unlisted Properties

- **Use Advanced Configuration Snippets (safety valves) to**
  - Override settings for properties not exposed in Cloudera Manager
  - Define additional configurations



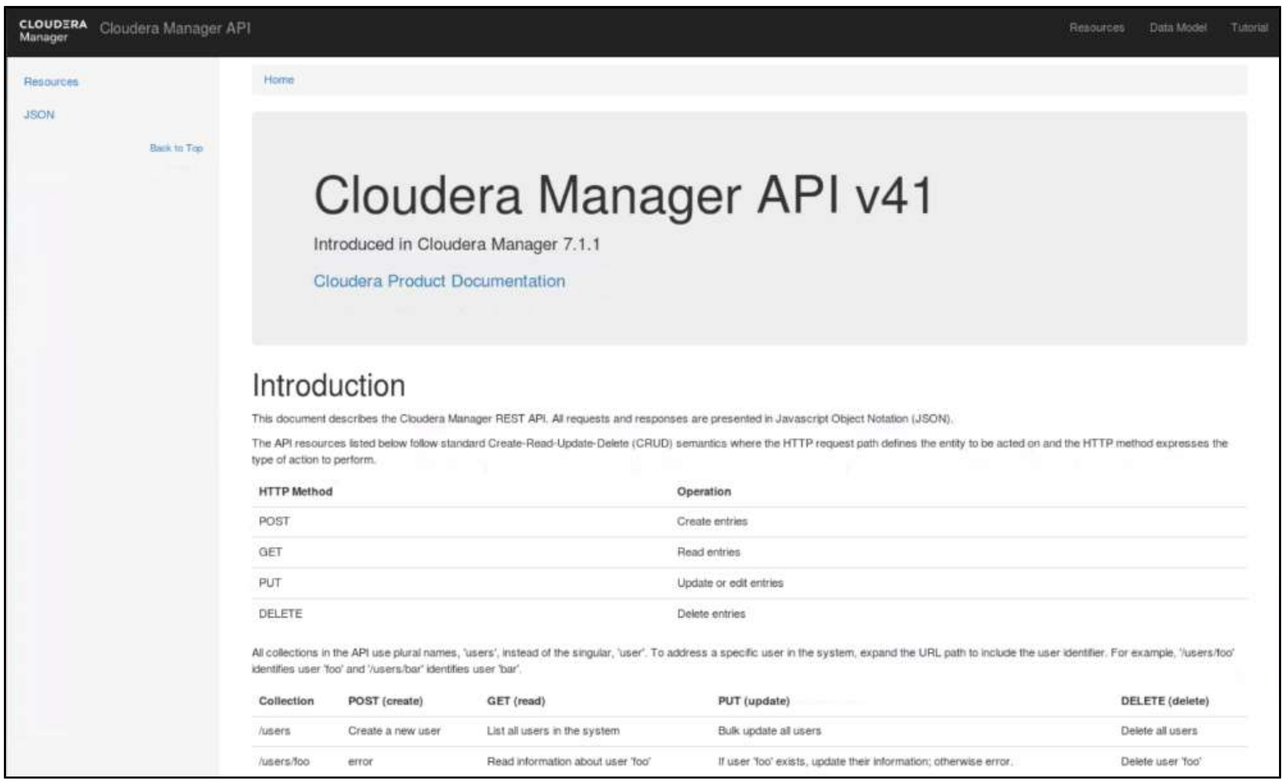
View as XML

- Click the View as XML link to view or edit the XML directly



The Cloudera Manager API

- **Use Cloudera Manager API to manage Hadoop clusters programmatically**
  - Deploy a cluster, configure, monitor, start and stop services, configure high availability, and more
  - Import and export entire cluster configuration



Cloudera Manager API

- **Access from CM at Support > API Documentation**
  - Access using `curl` or included client libraries
  - Python or Java clients recommended
  - You may also enjoy the Swagger UI provided for this API
- **The API accepts HTTP `POST`, `GET`, `PUT`, and `DELETE` methods**
  - Accepts and returns JSON formatted data
- **Cloudera Manager Tutorial at: https://cloudera.github.io/cm_api/**
- **See the Appendix to this course titled: *Cloudera Manager API* for more information.**

Chapter Topics

Cluster Configuration

- Overview
- Configuration Settings
- Modifying Service Configurations
- **Configuration Files**
- Managing Role Instances
- Adding New Services
- Adding and Removing Hosts
- Essential Points
- Hands-On Exercise: Configuring a Hadoop Cluster

Configuration Files on Cluster Hosts

- **Cloudera Manager deploys configuration settings to Cloudera Manager agents**
- **Cloudera Manager agents save settings to local files**
- **Service settings apply to service daemon**
  - Examples: YARN ResourceManager, HDFS NameNode daemons
- **Client settings are used by client application and command line tools to access cluster services**
  - Clients typically run on gateway nodes
  - Examples:
    - Spark and MapReduce applications access HDFS services
    - The Hive command line tool accesses the Hive services
- **Service and client configuration settings are stored separately**

Service Configuration Files

- **Cloudera Manager starts each service daemon with its own execution and configuration environment**
- **The Cloudera Manager agent pulls daemon configuration settings from the Cloudera Manager server and stores on disk**
- **Each daemon has a separate configuration file**
- **Files contain information daemon requires such as**
  - Arguments to `exec()`
  - Directories to create
  - cgroup settings
- **Default location for service daemon configuration files**
  - `/var/run/cloudera-scm-agent/process/`

Client Configuration Files (1)

- **Cloudera Manager creates client files with settings needed to access cluster services**
- **Example: a MapReduce application client configuration archive contains copies of**
  - `core-site.xml`
  - `hadoop-env.sh`
  - `hdfs-site.xml`
  - `log4j.properties`
  - `mapred-site.xml`
- **Default client configurations location:** `/etc/hadoop/conf`

Client Configuration Files (2)

- **Client configuration files are generated by Cloudera Manager when**
  - A cluster is created
  - A service or a gateway role is added on a host
- **A gateway role includes client configurations, libraries, and binaries, but no daemons**
- **Cloudera Manager prompts when client configuration redeployment is needed**
  - There is also an option to download and deploy manually

Summary: Server and Client Settings are Maintained Separately

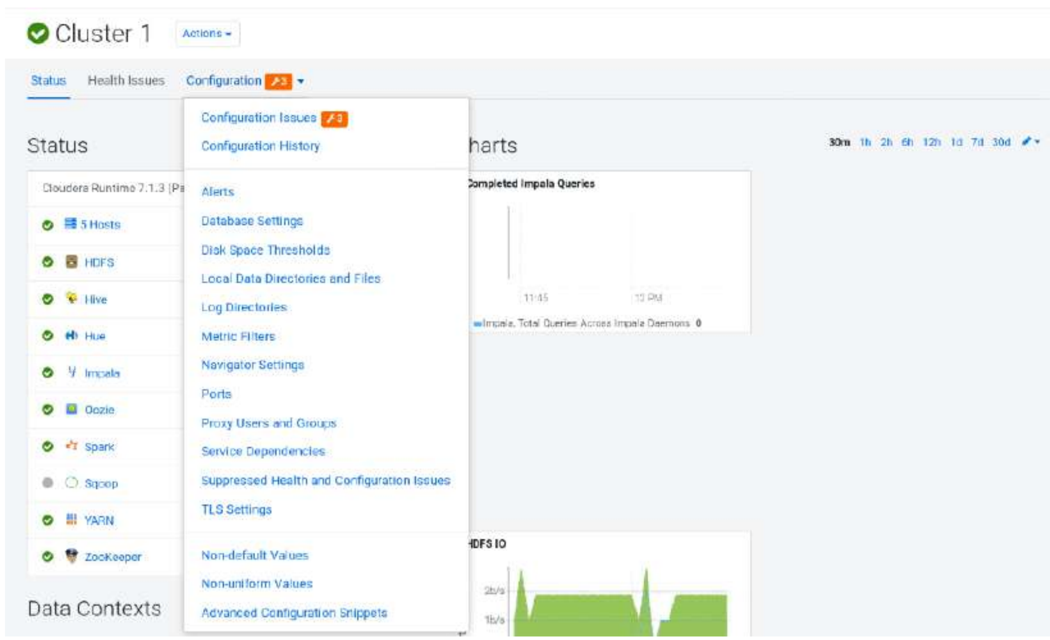
- **Cloudera Manager decouples server and client configurations**
  - Server settings (NameNode, DataNode) default location is `/var/run/cloudera-scm-agent/process` subdirectories on namenode
  - Client settings default location is `/etc/hadoop` subdirectories

```
training@tiger:/etc/hadoop$ tree conf
conf
├── __cloudera_generation__
├── container-executor.cfg
├── core-site.xml
├── hadoop-env.sh
├── hdfs-site.xml
├── log4j.properties
├── mapred-site.xml
├── ssl-client.xml
├── topology.map
├── topology.py
└── yarn-site.xml
```

```
training@tiger:/var/run/cloudera-scm-agent/process$ sudo tree
.
├── 10-hdfs-DATANODE
│   ├── cloudera-monitor.properties
│   ├── cloudera-stack-monitor.properties
│   ├── core-site.xml
│   ├── event-filter-rules.json
│   ├── hadoop-metrics2.properties
│   ├── hadoop-policy.xml
│   ├── hdfs.keytab
│   ├── hdfs-site.xml
│   ├── http-auth-signature-secret
│   ├── log4j.properties
│   ├── logs
│   │   ├── stderr.log
│   │   └── stdout.log
│   ├── ssl-client.xml
│   └── ssl-server.xml
├── 11-hdfs-SECONDARYNAMENODE
│   ├── cloudera-monitor.properties
│   ├── cloudera-stack-monitor.properties
│   ├── core-site.xml
│   ├── event-filter-rules.json
│   ├── hadoop-metrics2.properties
│   ├── hadoop-policy.xml
│   ├── hdfs.keytab
│   ├── hdfs-site.xml
│   ├── http-auth-signature-secret
│   ├── log4j.properties
│   ├── logs
│   │   ├── stderr.log
│   │   └── stdout.log
│   ├── ssl-client.xml
│   └── ssl-server.xml
```

Cluster-Wide Configurations

- **Access from Cloudera Manager's Cluster page Configuration menu**
- **Review or modify settings, including**
  - Non-default value configuration settings
  - Log directories
  - Disk space thresholds
  - Advanced Configuration Snippets



Chapter Topics

Cluster Configuration

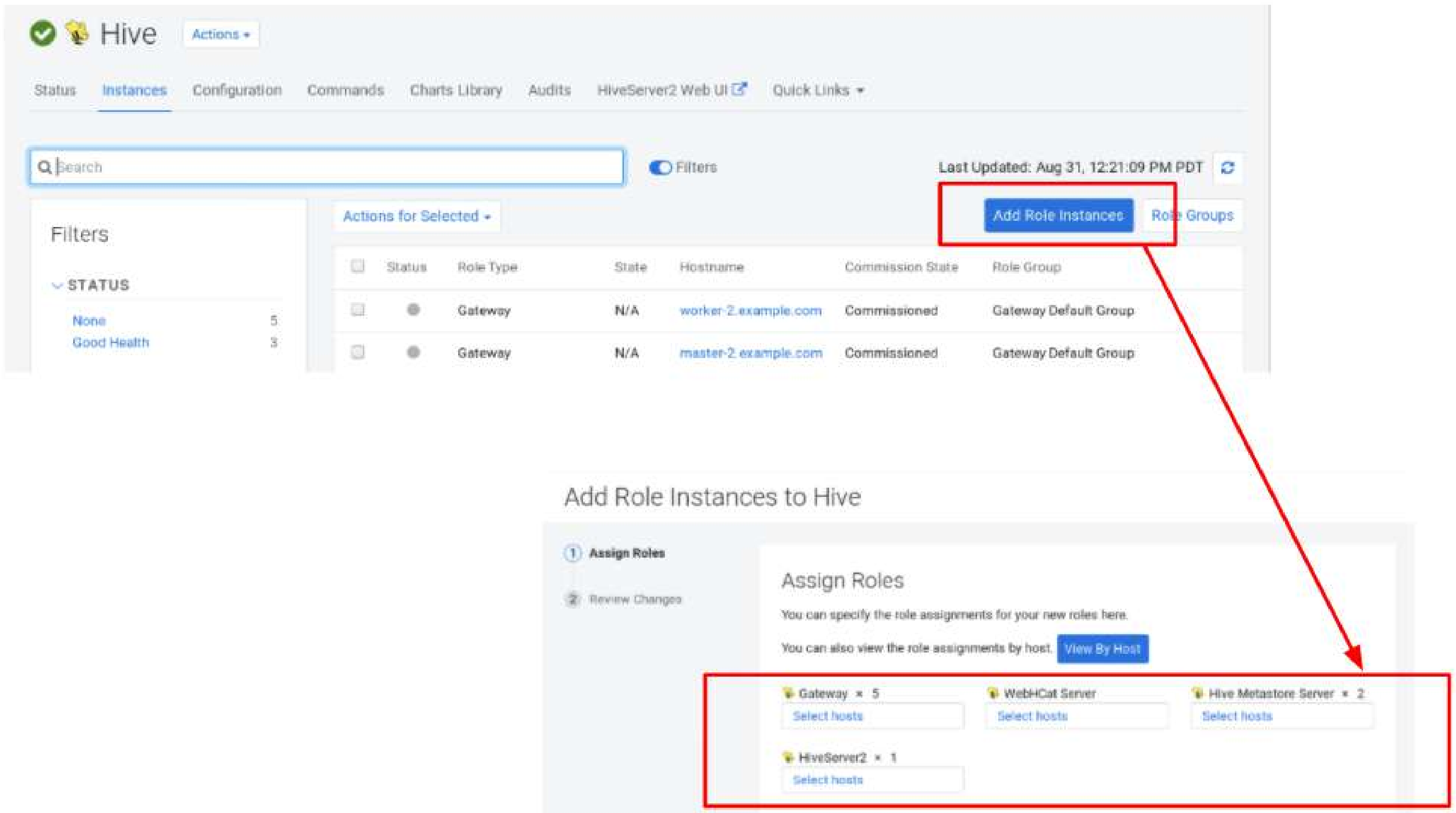
- Overview
- Configuration Settings
- Modifying Service Configurations
- Configuration Files
- **Managing Role Instances**
- Adding New Services
- Adding and Removing Hosts
- Essential Points
- Hands-On Exercise: Configuring a Hadoop Cluster

Role Instance Management Tasks

- **Add additional role instances after a service is created**
  - Example: Add a DataNode role instance to a new host
- **Start/stop/restart role instances**
- **Decommission a role instance**
  - Remove role instance while the cluster is running without losing data
- **Delete a role instance**
  - Decommission before deleting
  - Deleting a gateway role removes the client libraries
    - Does not delete the deployed client configurations

Example: Adding Hive Role Instances

- **On the Hive service's Instances tab, select Add Role Instances**



Gateway Role Instances

- **Service Gateway Role Instances**
  - Designate gateway hosts
    - Hosts receive client configuration files and libraries
    - Useful when the host does not have other roles for the service
  - Enable Cloudera Manager to control the client runtime environment
- **There is no process daemon associated with a gateway role**
- **Gateway hosts can be managed manually or by Cloudera Manager**
  - For manual management, Cloudera Manager provides a zip archive with configuration files

Chapter Topics

Cluster Configuration

Adding CDP Services to a Cluster

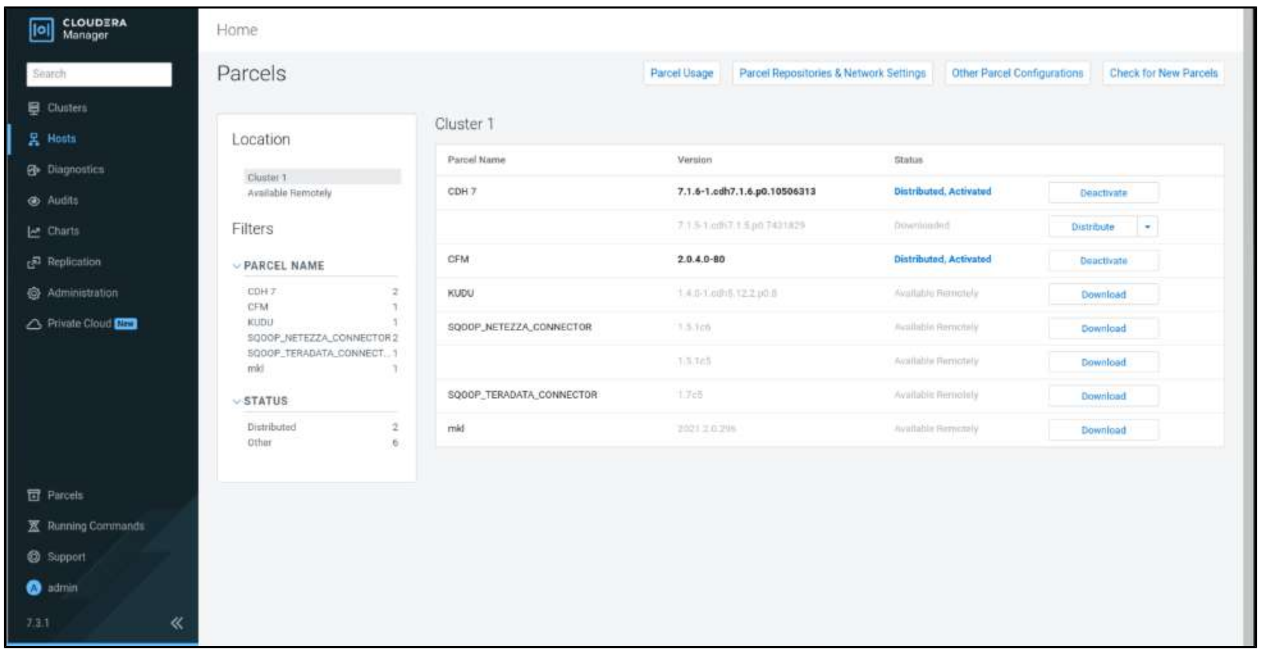
- **When creating a new cluster, use Add Cluster wizard**
  - All CDP services are automatically available
  - Multiple services can be added at once
- **When adding a service to an existing cluster, use Add Service wizard**
  - Choose **Add Service** from the cluster drop-down menu
  - Select the service you wish to install
  - Confirm dependencies (if required)
  - Choose which role instance(s) should be deployed to which host(s)

Add-on Services

- **Some services supported by Cloudera Manager are not part of CDP**
- **Common examples**
  - Anaconda
  - Cloudera Data Science Workbench (CDSW)
  - Apache Sqoop Connectors
- **Distributed as packages or Cloudera Manager parcels**
- **May be provided by Cloudera or by an independent software vendor**

Installing Add-on Services from Parcels (1)

- **Installation requires a parcel and usually a Custom Service Descriptor (CSD) file**
  - Contains the configuration needed to download and manage the new service
- **Place the CSD in the `/opt/cloudera/csd` directory**
- **Restart Cloudera Manager Server and the Cloudera Management Service**
- **Use Parcels page to download, distribute, and activate parcels**



Installing Add-on Service from Parcels (2)

1. **Add parcel URLs on** Configure **page**
2. **Click** Download
   - Downloads parcel to CM host parcel directory
   - `/opt/cloudera/parcel-repo` by default
3. **Click** Distribute
   - Pushes parcel to cluster hosts
   - `/opt/cloudera/parcels` by default
4. **Click** Activate
   - Adds the new service to available services list
5. **Use the standard** Add Service **wizard to deploy to cluster hosts**

Chapter Topics

Cluster Configuration

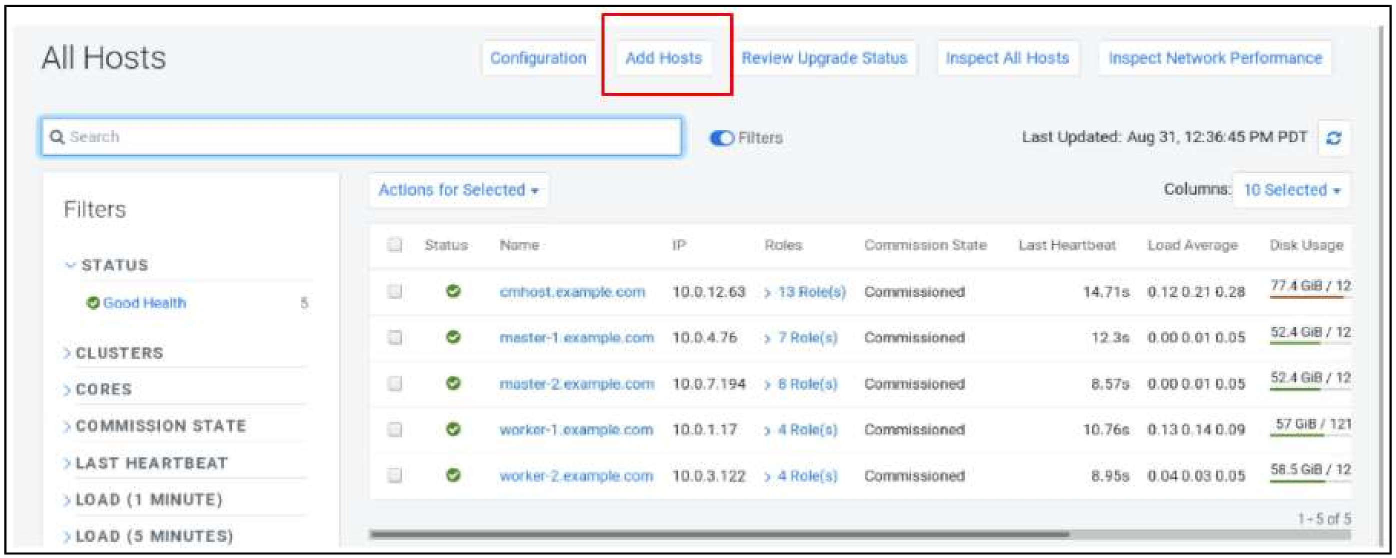
- Overview
- Configuration Settings
- Modifying Service Configurations
- Configuration Files
- Managing Role Instances
- Adding New Services
- **Adding and Removing Hosts**
- Essential Points
- Hands-On Exercise: Configuring a Hadoop Cluster

Cluster Size

- **A cluster can run on a single machine**
  - Only for testing, developing
  - Not for production clusters
- **Many organizations start with a small cluster and grow it as required**
  - Perhaps initially just eight or ten hosts
  - As the volume of data grows, more hosts can easily be added
- **Increase cluster size when you need to increase**
  - Computation power
  - Data storage
  - Memory

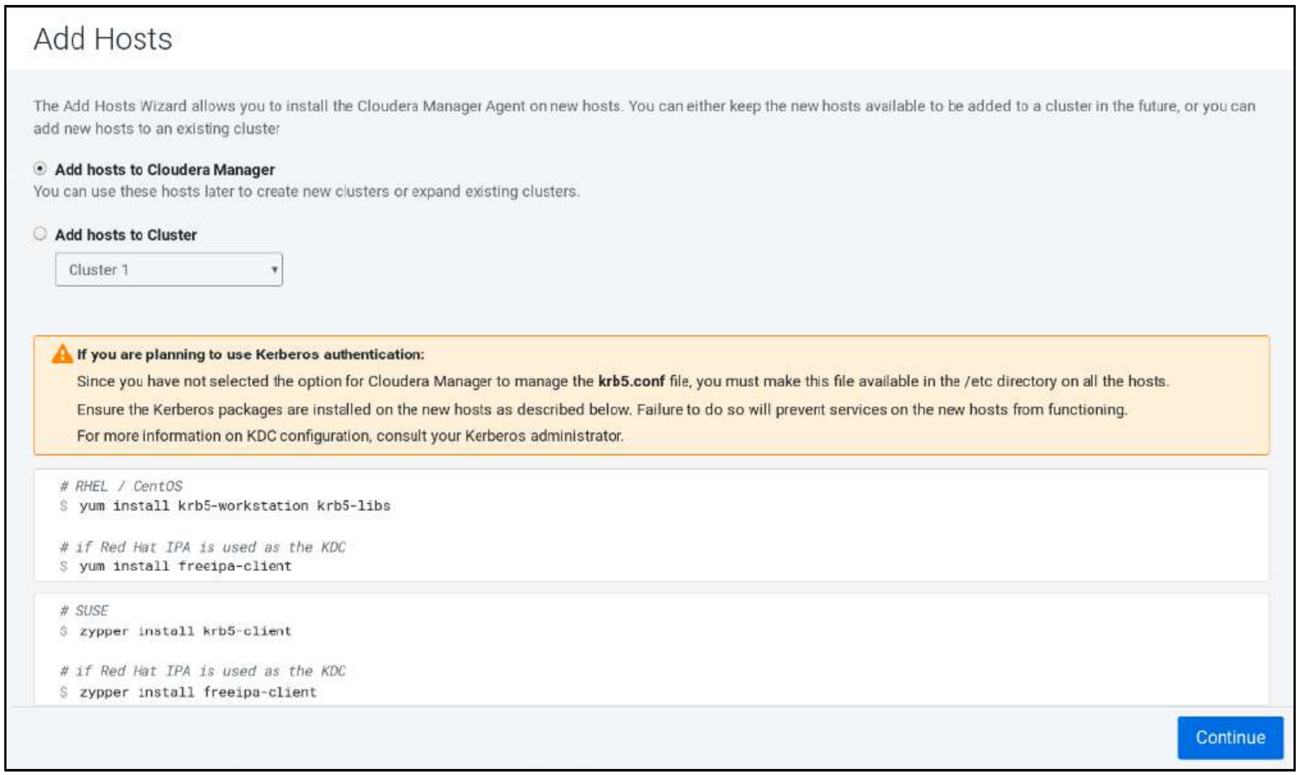
Adding a Host to the Cluster

- **Add new hosts from the Cloudera Manager** Hosts **page**
- **The host installation wizard will**
  - Install Cloudera Manager agent on new host
  - Provide the option to install JDK (three options provided)



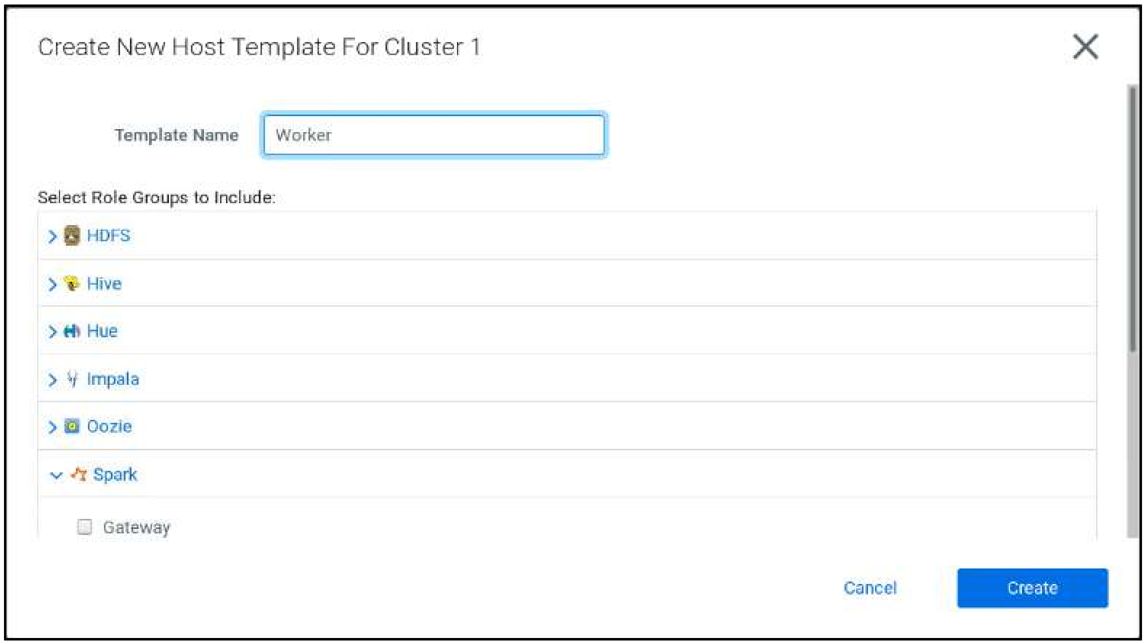
Add Host Wizard

- **The Add Hosts Wizard allows you to the Cloudera Manager Agent on new hosts for:**
  - Future use in a cluster
  - Add to an existing cluster



Host Templates

- **Helpful for deploying multiple roles to a host**
- **Streamlines the process of configuring and deploying new hosts**
- **Promotes standardization of hosts performing the same function**
  - Example: Define a “Worker” template and apply it to all new worker hosts



Removing a Host from the Cluster

- **Choose removal option from** All Hosts **page**
  - Remove From Cluster option
    - Keeps the host available to Cloudera Manager
  - Remove From Cloudera Manager option
    - Cloudera Manager will no longer manage the host
- **Both methods will**
  - Decommission and delete host's role instances
  - Remove managed service software
  - Preserve data directories

Chapter Topics

Cluster Configuration

Essential Points

- **Cloudera Manager organizes service configurations**
  - Uses constructs such as roles, role groups, and role instances
- **Cloudera Manager manages configuration changes across the cluster**
  - Provides tools for locating, modifying, and applying configuration changes
- **Use Cloudera Manager to manage hosts and services**
  - Commission/decommission hosts
  - Define and apply host templates
  - Add new services (including add-on services)

Chapter Topics

Cluster Configuration

- Overview
- Configuration Settings
- Modifying Service Configurations
- Configuration Files
- Managing Role Instances
- Adding New Services
- Adding and Removing Hosts
- Essential Points
- **Hands-On Exercise: Configuring a Hadoop Cluster**

Hands-On Exercise: Configuring a Hadoop Cluster

- **In this exercise, you will modify a configuration, add new services, create and apply a Host Template, and utilize an Advanced Configuration Snippet**
- **Please refer to the Hands-On Exercise Manual for instructions**

Data Storage

Chapter 5

Course Chapters

- Introduction
- Cloudera Data Platform
- CDP Private Cloud Base Installation
- Cluster Configuration
- **Data Storage**
- Data Ingest
- Data Flow
- Data Access and Discovery
- Data Compute
- Managing Resources
- Planning Your Cluster
- Advanced Cluster Configuration
- Cluster Maintenance
- Cluster Monitoring
- Cluster Troubleshooting
- Security
- Private Cloud / Public Cloud
- Conclusion
- Appendix: Cloudera Manager API
- Appendix: Ozone Overview

Data Storage

After completing this chapter, you will be able to

- **Summarize HDFS architecture, features, and benefits**
- **Explain how HDFS distributes and keeps track of data across machines**
- **Describe functionality of the NameNode and DataNodes in an HDFS deployment**
- **Use web UIs and command-line tools to load data into and interact with HDFS**
- **Summarize the use and administration of HBase**
- **Summarize the use and administration of Kudu**
- **Summarize the options and usage of Cloud Storage**

Chapter Topics

Data Storage

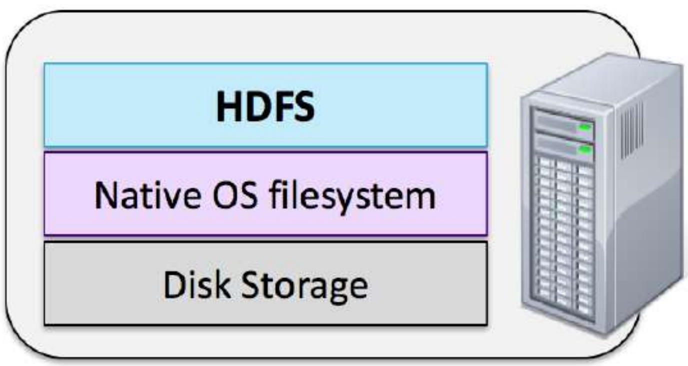
- **Overview**
- HDFS Topology and Roles
- HDFS Performance and Fault Tolerance
- HDFS and Hadoop Security Overview
- Working with Namenode UI
- Instructor-Led Demonstration: Namenode User Interface
- Working with HDFS
- Hands-On Exercise: Working with HDFS
- HBase Overview
- Kudu Overview
- Cloud Storage Overview
- Essential Points
- Hands-On Exercise: Storing Data in Amazon S3

Hadoop Data Storage

- **Hadoop supports a number of data storage platforms**
  - Hadoop Distributed File System (HDFS)
    - Hierarchical file-based storage
  - Apache HBase
    - NoSQL database, built on HDFS
  - Apache Kudu
    - Table-based storage for fast processing and SQL analytics
  - Cloud storage
    - Amazon S3, Microsoft Azure, Google GCS
  - Ozone
    - A scalable, redundant, and distributed object store for Hadoop
    - Ozone can function effectively in containerized environments such as Kubernetes and YARN

HDFS: The Hadoop Distributed File System

- **Emulates an OS filesystem**
- **A Java application running on cluster nodes**
  - Based on Google File System (GFS)
- **Sits on top of a native filesystem**
  - Such as ext3, ext4, or xfs
- **Redundant storage for massive amounts of data on industry-standard hardware**
- **Data is distributed at load time**



HDFS Features

- **High read throughput performance**
- **Fault tolerance**
- **Relatively simple centralized management**
  - Master-worker architecture
- **Security**
  - Optionally configured with Kerberos for secure authentication
- **Optimized for distributed processing**
  - Data locality
- **Scalability**

HDFS Characteristics

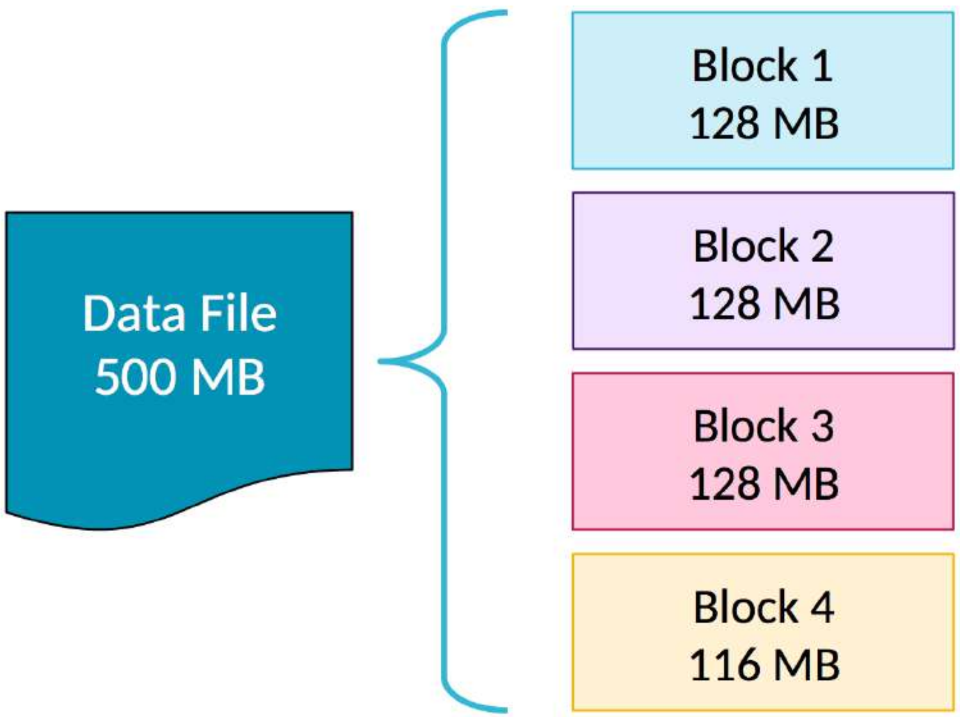
- **Fault tolerant to handle component failure**
- **Optimized for “modest” number of large files**
  - Millions of large files, not billions of small files
  - Each file is likely to be 128MB or larger
    - Multi-gigabyte files typical
- **Files are immutable**
  - Data can be appended, but a file’s existing contents cannot be changed
- **Designed for large streaming reads**
  - Favors high sustained throughput over low latency

Options for Accessing HDFS

- **From the command line**
  - `hdfs dfs`
    - Synonym for `hadoop fs`
- **From a web browser**
  - Hue
  - Cloudera Manager
  - NameNode web UI
- **Other programs**
  - NFS Gateway
    - Allows a client to mount HDFS as part of the local file system
  - Java API
    - Used by MapReduce, Spark, Impala, Hue, Sqoop, and so on
  - RESTful interface
    - WebHDFS and HttpFS

HDFS Blocks

- **When a file is added to HDFS, it is split into blocks**
  - Similar concept to native filesystems, but *much* larger block size
  - Default block size is 128MB (configurable)
  - HDFS only uses as much disk space as there is data in the block



HDFS Replication

- **Blocks are replicated to multiple hosts based on *replication factor***
  - Default replication factor is three
- **Replication increases reliability and performance**
  - Reliability—data can tolerate loss of all but one replica
  - Performance—more opportunities for data locality
- **To set the replication factor for the cluster:**
  - Set the HDFS property `dfs.replication`
- **To set the replication factor for a specific directory:**
  - Any new files added to the directory will use the global replication factor
  - Example command will set replication factor of 8 to all files in the pathway indicated, recursively

```
$ hdfs dfs -setrep -R 8 /pathway
```

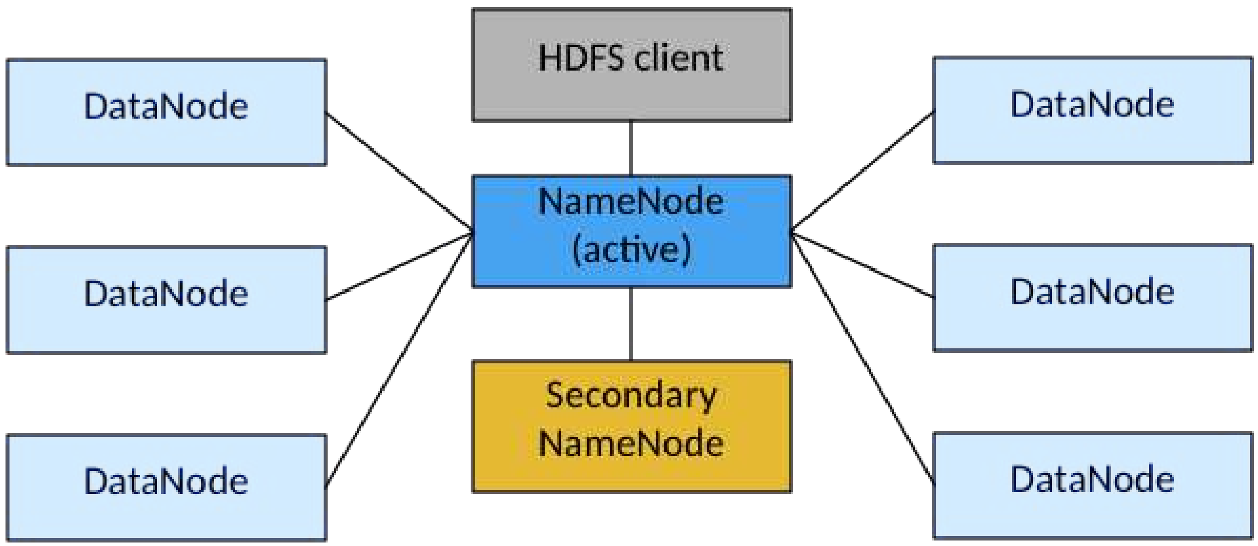
Chapter Topics

Data Storage

- Overview
- **HDFS Topology and Roles**
- HDFS Performance and Fault Tolerance
- HDFS and Hadoop Security Overview
- Working with Namenode UI
- Instructor-Led Demonstration: Namenode User Interface
- Working with HDFS
- Hands-On Exercise: Working with HDFS
- HBase Overview
- Kudu Overview
- Cloud Storage Overview
- Essential Points
- Hands-On Exercise: Storing Data in Amazon S3

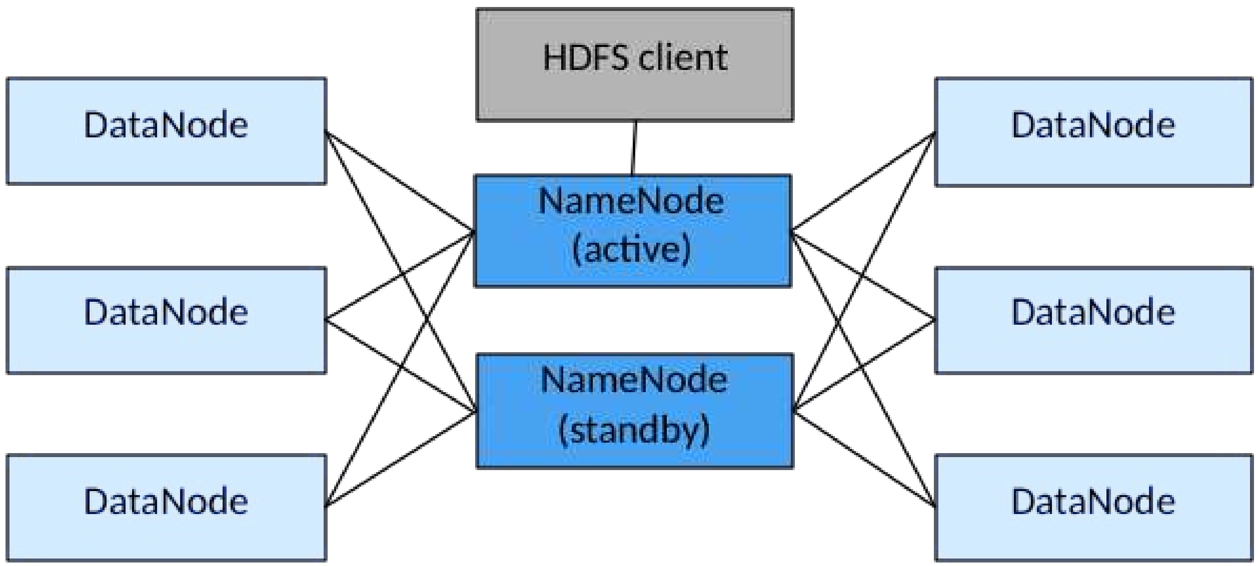
HDFS Without High Availability

- **You can deploy HDFS with or without high availability**
- **Without high availability, there are three daemons**
  - NameNode (master)
  - Secondary NameNode (master)
  - DataNode (worker)
- **The Secondary NameNode is *not* a failover NameNode**
  - It only handles checkpointing the file metadata



HDFS With High Availability

- **Eliminates the NameNode as a single point of failure**
- **Two NameNodes: one active and one standby**
  - Standby NameNode takes over when active NameNode fails
- **Secondary NameNode is not used in high availability mode**



HDFS DataNodes

- **Contents of files in HDFS are stored as *blocks* on the worker hosts**
- **Each worker host runs a DataNode daemon**
  - Controls access to the blocks
  - Communicates with the NameNode
- **Blocks are simply files on the worker hosts' underlying filesystem**
  - Named `blk_`*`xxxxxxx`*
  - The location on disk on each DataNode defaults to `/dfs/dn`
    - Set `dfs.data.dir` property to change
  - DataNodes are unaware of which stored file a block is part of
    - That information is *only* stored as metadata on the NameNode

HDFS NameNode

- **The NameNode holds all *metadata* about files and blocks**
  - Stored in RAM and persisted to disk
- **Metadata is loaded from disk when the NameNode daemon starts up**
  - Filename is `fsimage`
  - Note: block locations are *not* stored in `fsimage`
- **Changes to the metadata are stored in RAM**
  - Changes are also written to an edits log
- **Note: the data stored in blocks *never* passes through the NameNode**
  - For writes, reads, or during re-replication

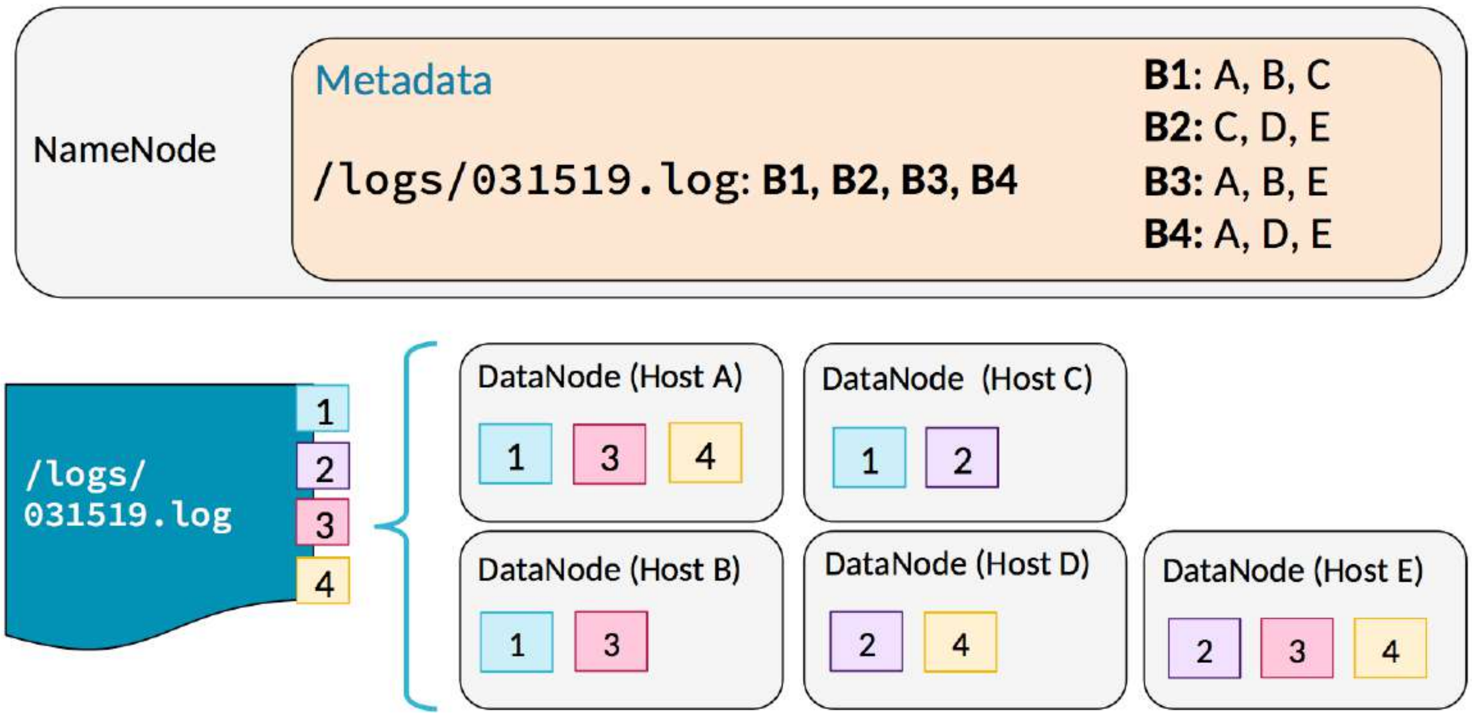
NameNode Memory Allocation (1)

- **Default Java heap size on the NameNode is 1 GB**
  - At least 1GB recommended for every million HDFS blocks
- **Items stored by the NameNode**
  - Filename, file ownership, and permissions
  - Name and location of the individual blocks
- **Each item uses approximately 150 to 250 bytes of memory**

NameNode Memory Allocation (2)

- **Fewer files requires less NameNode memory**
  - Which is why HDFS prefers fewer, larger files
- **Example: 1GB of data, HDFS block size 128 MB**
  - Stored as 1 x 1GB file
    - Name: 1 item
    - Blocks: 8 items
    - *Total items in memory: 9*
  - Stored as 1000 x 1MB files
    - Names: 1000 items
    - Blocks: 1000 items
    - *Total items in memory: 2000*

HDFS Topology and Replication



Chapter Topics

Data Storage

- Overview
- HDFS Topology and Roles
- **HDFS Performance and Fault Tolerance**
- HDFS and Hadoop Security Overview
- Working with Namenode UI
- Instructor-Led Demonstration: Namenode User Interface
- Working with HDFS
- Hands-On Exercise: Working with HDFS
- HBase Overview
- Kudu Overview
- Cloud Storage Overview
- Essential Points
- Hands-On Exercise: Storing Data in Amazon S3

File System Metadata Snapshot and Edit Log

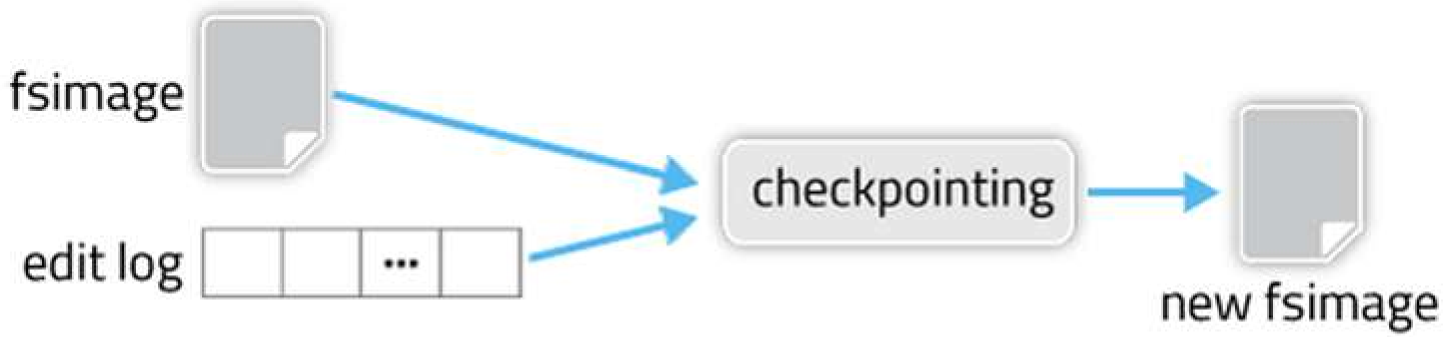
- **The `fsimage` file contains a file system metadata snapshot**
  - It is *not* updated with every write
- **NameNode records HDFS write operations in an edit log file**
  - Also updates in-memory representation of file system metadata
- **At start-up, NameNode reads metadata from `fsimage`, then applies edits from edit log**
  - More efficient than rewriting huge `fsimage` file with every write operation

Checkpointing the File System Metadata

- **A checkpoint consists of**
  - The most recent `fsimage` file
  - Edit logs for write operations since `fsimage` was last save
- **HDFS creates checkpoints periodically**
  1. Merges edits with the most recent **`fsimage`** file
  2. Replaces it with a new **`fsimage`** file
  3. Clears the edits log

Benefit of Checkpointing

- **Checkpointing speeds up NameNode restarts**
  - Prevents edit log from growing very large
  - NameNode takes less time to apply smaller edit log at restart



Edit Log Location and Secondary NameNode

- **When NameNode high availability is *not* configured**
  - Checkpointing performed by Secondary NameNode
  - Secondary NameNode pulls the latest `edit` log(s) from the NameNode and rolls the edit log
  - After completing the checkpoint, new `fsimage` file is copied back to NameNode
- **When NameNode high availability is configured**
  - Checkpoints conducted by an ensemble of JournalNode role instances
  - Edit log files are stored in a shared location
  - The active NameNode can write to the log files
  - Standby NameNodes have read access
  - Checkpointing is conducted by the Standby NameNode in HA
- **The Secondary NameNode is *not* a failover NameNode**
  - The Secondary NameNode only used when high availability is not configured

HDFS Read Caching (1)

- **Applications can instruct HDFS to *cache* blocks of a file**
  - Blocks are stored on the DataNode in off-heap RAM
  - Cache-aware applications will read cached blocks if available
    - Such as Impala
- **HDFS caching provides benefits over standard OS-level caching**
  - Avoids memory-to-memory copying
- **Cloudera Manager enables HDFS caching by default**
  - Set `dfs.datanode.max.locked.memory` to control amount of memory per DataNode for caching

HDFS Read Caching (2)

- **Files can be cached manually**

  1. Create a cache pool

```
$ hdfs cacheadmin -addPool testPool
```

  2. Add files to cache pools

```
$ hdfs cacheadmin -addDirective -path /myfile -pool
 testPool
```

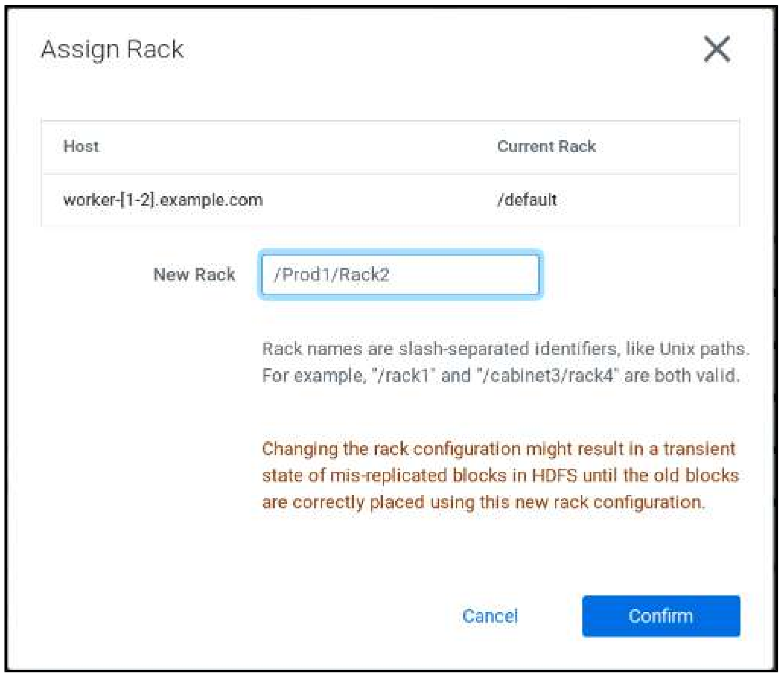
Hadoop is "Rack-Aware"

- **Hadoop can be configured to know how "close" hosts are to one another**
  - Closest: on the same host, or within the same rack
- **Client read operations run on closest node when possible**
- **HDFS replicates data blocks on hosts on different racks**
  - Provides extra data security in case of catastrophic hardware failure
- **Rack-awareness is an important feature for on-premises deployments**
  - Not applicable for cloud deployments

Configuring Rack Topology

- **To maximize performance, specify topology of hosts and racks**
  - Important for clusters that span more than one rack
  - If the cluster has more than 10 hosts, you should specify the rack for each host
  - Specify Rack ID in the form `/datacenter/rack`
  - Any host without a specified rack location assigned location `/default`

All Hosts > Assign Rack



Dealing with Data Corruption

- **Clients create checksums for each block**
  - First when block is written, again when block is read
  - Client compares checksums when reading the block
- **If read and write checksums do not match, client**
  - Informs NameNode of a corrupted version of the block
    - NameNode re-replicates that block elsewhere
  - Reads a copy of the block from the another DataNode
- **The DataNode verifies checksums for blocks periodically, to avoid “bit rot”**
  - Set `dfs.datanode.scan.period.hours` to configure
    - Default value is every three weeks

Data Reliability and Recovery

- **DataNodes send *heartbeats* to the NameNode periodically**
  - Configure frequency with `dfs.heartbeat.interval`
  - Default is every three seconds
- **If heartbeats are not received, a DataNode is:**
  - Declared stale after 30 seconds and used last
    - `dfs.namenode.stale.datanode.interval`
  - Declared dead after 10.5 minutes and not used
    - `dfs.namenode.heatbeat.recheck-interval` and `dfs.heatbeat.interval`
  - A dead DataNode forces the NameNode to re-replicate the data blocks
- **A DataNode can rejoin a cluster after being down for a period**
  - The NameNode ensures blocks are not over-replicated by instructing DataNodes to remove excess copies

Chapter Topics

Data Storage

- Overview
- HDFS Topology and Roles
- HDFS Performance and Fault Tolerance
- **HDFS and Hadoop Security Overview**
- Working with Namenode UI
- Instructor-Led Demonstration: Namenode User Interface
- Working with HDFS
- Hands-On Exercise: Working with HDFS
- HBase Overview
- Kudu Overview
- Cloud Storage Overview
- Essential Points
- Hands-On Exercise: Storing Data in Amazon S3

HDFS File Permissions

- **Files and directories have an owner, a group, and permissions**
  - Very similar to UNIX file permissions
- **HDFS file permissions are set for each of owner, group, and other**
  - read (`r`), write (`w`), and execute (`x`)
  - For files, execute setting is ignored
  - For directories, execute setting means that its children can be accessed
- **HDFS enforces permissions by default**
  - Configure with `dfs.permissions` property
- **HDFS permissions are designed to stop good people doing foolish things**
  - Not to stop bad people doing bad things
  - HDFS believes you are who you tell it you are

Superuser and Supergroup (1)

- **Superuser**
  - The user with the same identity as the process
  - Examples: hdfs, yarn, nifi, etc.
  - The super-user can do anything
  - Permissions checks never fail for the super-user
- **Supergroup**
  - Used to ensure that the Hadoop Client has superuser access
  - Can be configured using dfs.permissions.superusergroup property in core-site.xml

Superuser and Supergroup (2)

- **To add a group called supergroup:**

```
$ ssh training@master-1 sudo /usr/sbin/groupadd supergroup
```

- **To view the list of groups for a user:**

- **To add a user to the group:**

```
$ ssh master-1 sudo /usr/sbin/usermod -aG supergroup training
```

- **To view the list of groups for a user:**

```
$ ssh master-1 groups training
```

Result example: `training : training wheel supergroup`

Hadoop Security Overview

- **Authentication**
  - Ensures systems or users are who they claim to be
  - Hadoop can provide strong authentication control using Kerberos
    - Cloudera Manager simplifies Kerberos deployment
  - Authentication using LDAP is available
- **Authorization (access control)**
  - Allowing people or systems to do some things but not other things
  - Hadoop has traditional POSIX-style permissions for files and directories
  - Access Control Lists (ACLs) for HDFS
  - Atrribute-based access control provided with Apache Ranger
- **Data encryption levels**
  - File system (for data at rest), HDFS, and network levels

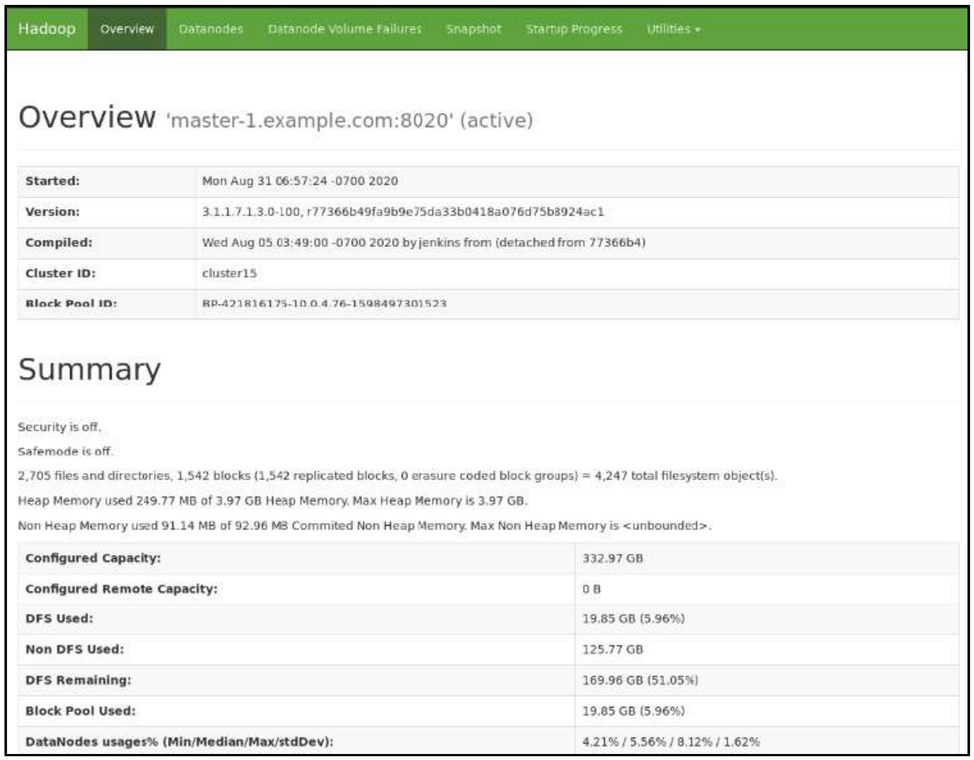
Chapter Topics

Data Storage

- Overview
- HDFS Topology and Roles
- HDFS Performance and Fault Tolerance
- HDFS and Hadoop Security Overview
- **Working with Namenode UI**
- Instructor-Led Demonstration: Namenode User Interface
- Working with HDFS
- Hands-On Exercise: Working with HDFS
- HBase Overview
- Kudu Overview
- Cloud Storage Overview
- Essential Points
- Hands-On Exercise: Storing Data in Amazon S3

NameNode Web User Interface

- **Key features**
  - HDFS service status and reports
  - NameNode health and status
  - DataNode storage capacity and status
  - File details such as block IDs and locations



Chapter Topics

Data Storage

- Overview
- HDFS Topology and Roles
- HDFS Performance and Fault Tolerance
- HDFS and Hadoop Security Overview
- Working with Namenode UI
- **Instructor-Led Demonstration: Namenode User Interface**
- Working with HDFS
- Hands-On Exercise: Working with HDFS
- HBase Overview
- Kudu Overview
- Cloud Storage Overview
- Essential Points
- Hands-On Exercise: Storing Data in Amazon S3

Instructor-Led Demonstration of Namenode WebUI (optional)

- **This demonstration will introduce several of the read-only functions of the NameNode UI**
- **Options include:**
  - HDFS service and status
  - DataNode information
  - Browse files
  - Browse logs

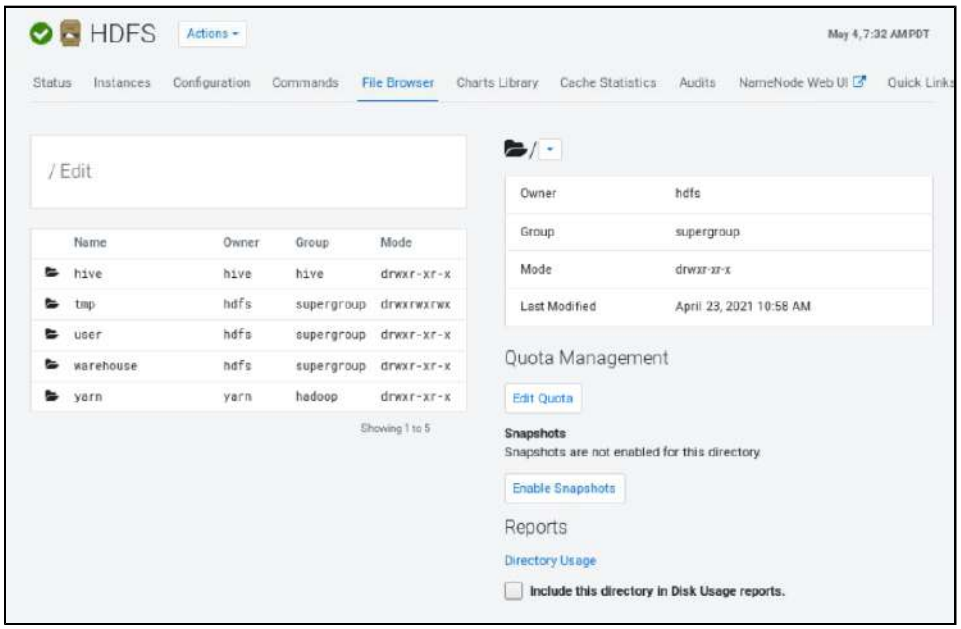
Chapter Topics

Data Storage

- Overview
- HDFS Topology and Roles
- HDFS Performance and Fault Tolerance
- HDFS and Hadoop Security Overview
- Working with Namenode UI
- Instructor-Led Demonstration: Namenode User Interface
- **Working with HDFS**
- Hands-On Exercise: Working with HDFS
- HBase Overview
- Kudu Overview
- Cloud Storage Overview
- Essential Points
- Hands-On Exercise: Storing Data in Amazon S3

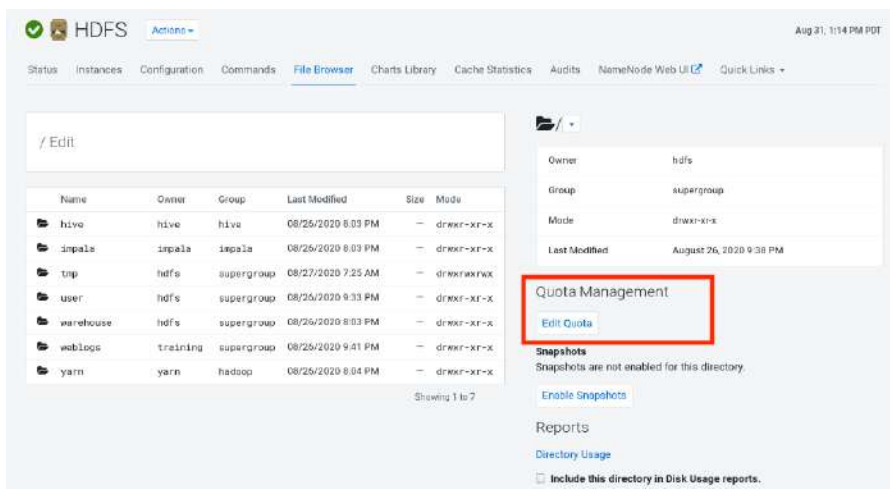
The Cloudera Manager HDFS File Browser

- **Provides file system administration features**
  - Utilization reports
  - Storage quotas
  - Snapshot management
- **This feature is also available through CLI**



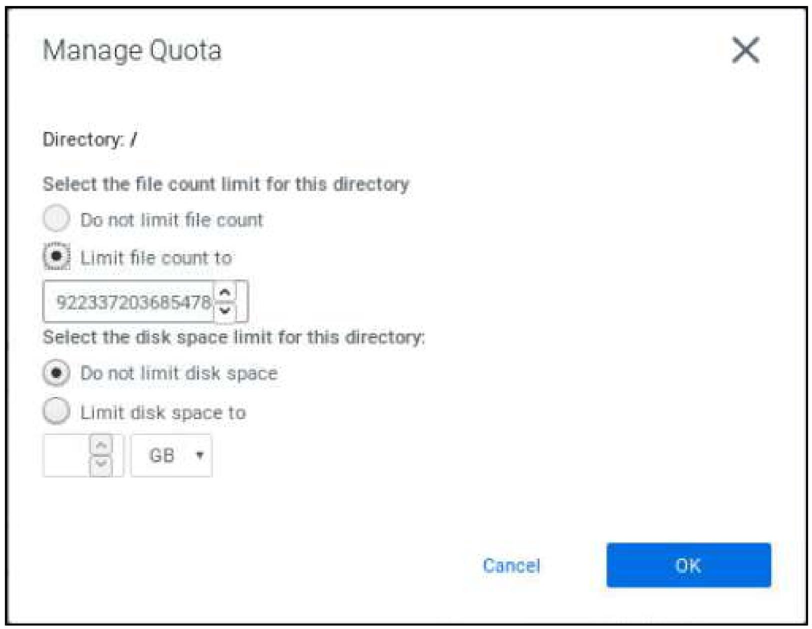
HDFS Quotas

- **As an administrator, you can set up HDFS quotas for directories**
- **You can use set file count or space quotas**
- **Considerations for working with HDFS quotas:**
  - Quotas for names and quotas for space are independent of each other
  - Block allocation fails if the quota prevents a full block from being written
  - You cannot create more files and directories if their creation would cause the quotas to exceed
  - If you are using replication, each replica counts against the quota



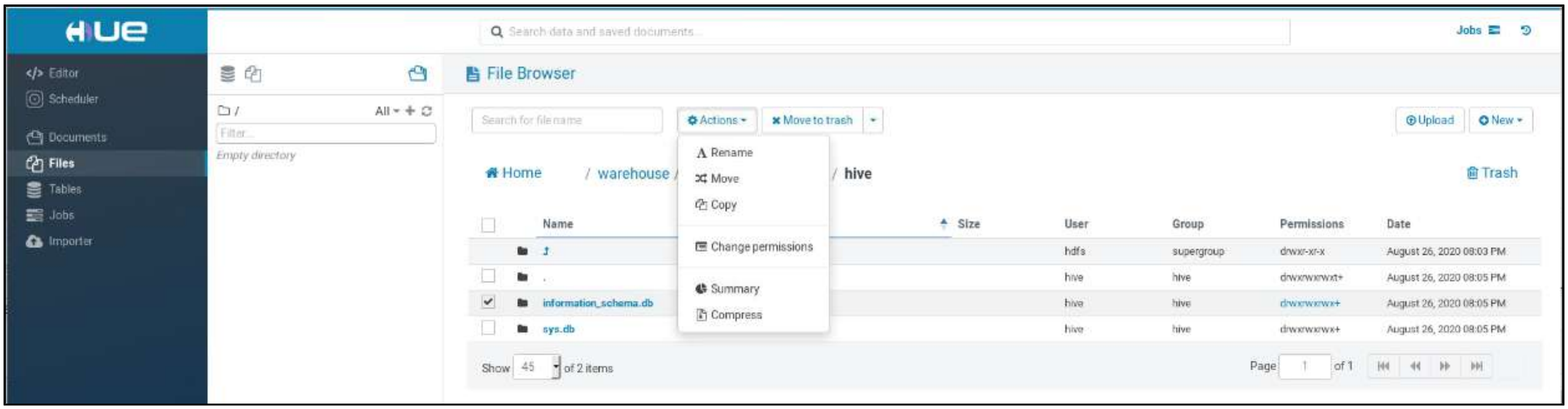
Quota Types

- **Name Quotas**
  - The name quota is a hard limit on the number of file and directory name
  - File and directory creations fail if the quota would be exceeded
- **Space Quotas**
  - The space quota is a hard limit on the number of bytes used in that directory
  - Each replica of a block counts against the quota



The Hue File Browser

- **Focused on the needs of the end user**
  - Upload, copy, delete, move, rename directories or files
  - Update permissions on files or directories
  - Set replication factor on files
  - Explore contents of text-based files



Accessing HDFS from the Command Line

- **HDFS is not a general purpose filesystem**
  - Not built into the OS, so only specialized tools can access it
- **End users typically use the `hdfs dfs` command**
  - Actions are specified with subcommands (prefixed with a hyphen)
  - Most subcommands are similar to corresponding UNIX commands
- **Display the contents of the `/user/fred/sales.txt` file**

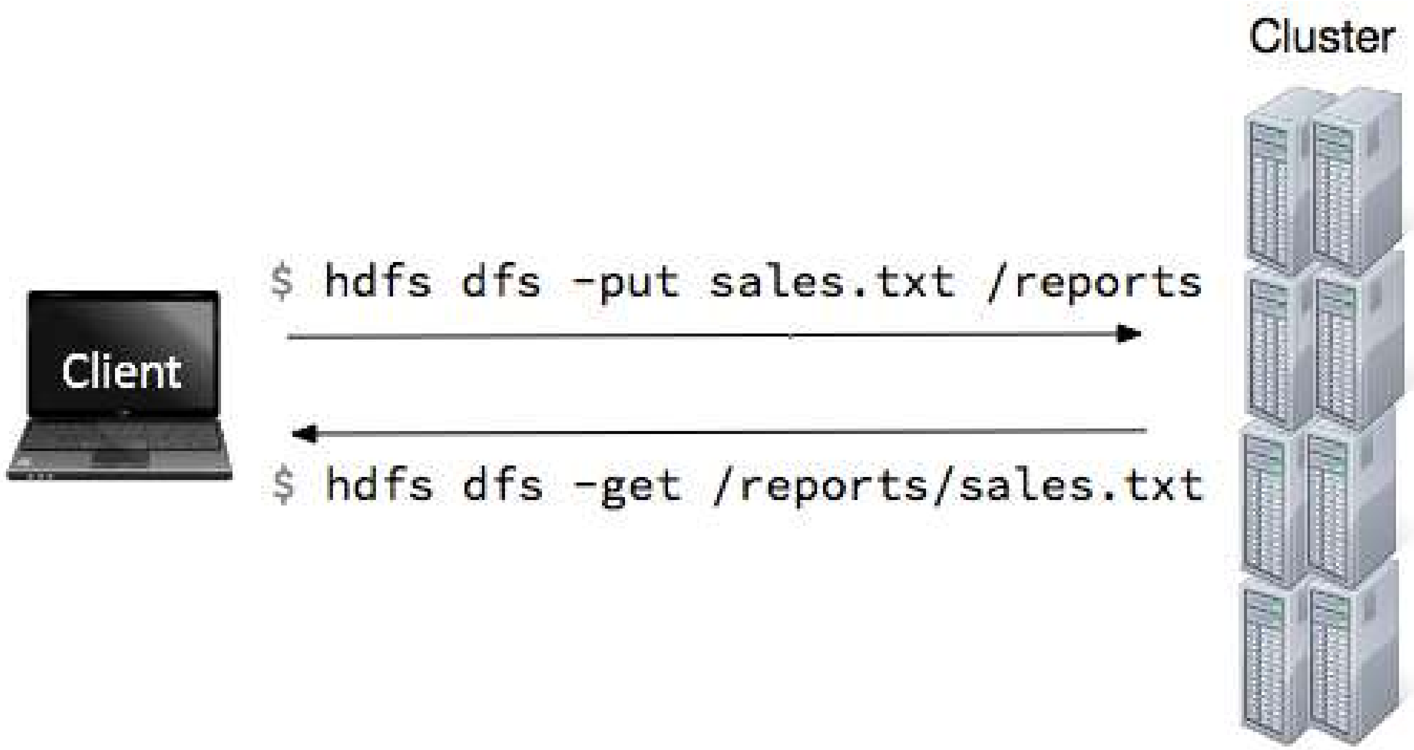
```
$ hdfs dfs -cat /user/fred/sales.txt
```

- **Create a directory called `reports` below the root**

```
$ hdfs dfs -mkdir /reports
```

Copy Local Data to and from HDFS

- The `hdfs dfs -put` copies local files to HDFS
- The `hdfs dfs -get` command copies files from HDFS to local file system



Some Common `hdfs dfs` Commands

- **Copy file `input.txt` from local disk to the current user's directory in HDFS**

```
$ hdfs dfs -put input.txt input.txt
```

  - This will copy the file to `/user/`*`username`*`/input.txt`

- **Get a directory listing of the HDFS root directory**

```
$ hdfs dfs -ls /
```

- **Delete the file `/reports/sales.txt`**

```
$ hdfs dfs -rm /reports/sales.txt
```

Filesystem Check: `hdfs fsck`

- **The `hdfs fsck` utility can be used to check the health of files in HDFS**
- **It also will report missing blocks and over- or under-replicated blocks**
- **Common commands**
  - Check the entire filesystem and provide a block replication summary

```
$ hdfs fsck /
```

  - Find all the blocks that make up a specific file

```
$ hdfs fsck /somedir/part-00001 -files -blocks -racks
```

NFS Gateway (1)

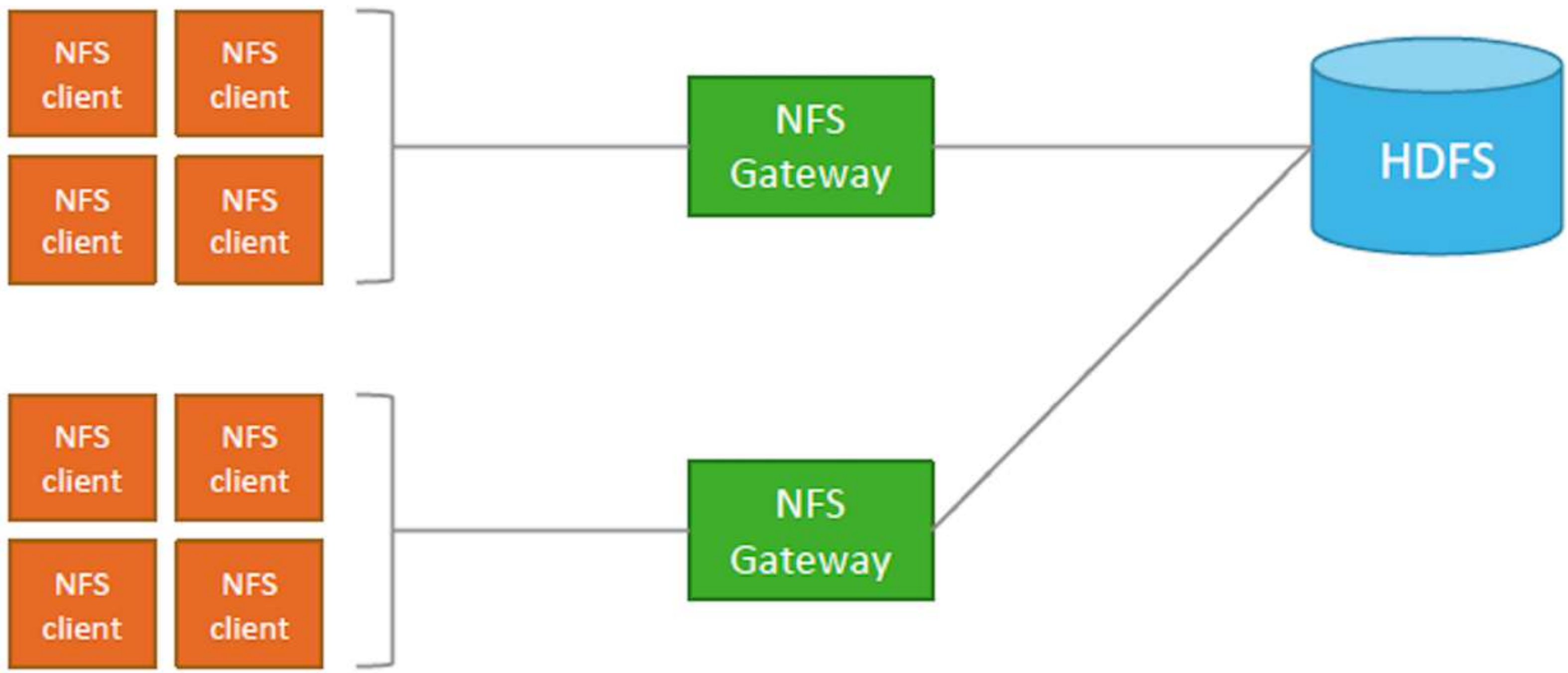
- **NFS Gateway for HDFS allows clients to mount HDFS and interact with it through NFS**
- **Interact with it as if it were part of the local file system**
- **After mounting HDFS, a client user can perform the following tasks:**
  - Browse the HDFS file system through the local file system
  - Upload and download files between the HDFS file system and local file system
  - Stream data directly to HDFS through the mount point
  - File append is supported, but random write is not supported
- **Prerequisites for using NFS Gateway**
  - NFS Gateway machine must be running all components that are necessary for running an HDFS client, such as a Hadoop core JAR file and a HADOOP_CONF directory
  - NFS Gateway can be installed on any DataNode, NameNode, or CDP client
  - Start the NFS server on that machine

NFS Gateway (2)

- **Configure the NFS gateway**
  - Ensure that the proxy user for the NFS Gateway can proxy all the users
  - Configure settings specific to the Gateway
- **Start and stop the NFS Gateway services**
  - rpcbind (or portmap)
  - mountd
  - nfsd
- **Access HDFS from the NFS Gateway**
  - Mount the namespace
    ```
    mount -t nfs -o
    vers=3,proto=tcp,nolock,sync,rsize=1048576,wsize=1048576
    $server:/ $mount_point
    ```
  - Set up the NFS client hosts

NFS Gateway (3)

- **Each NFS Gateway has finite network, CPU and memory resources**
- **Multiple gateways increase scalability**
- **NFS client mounts do not failover between gateways**



Optimizing Data Storage

- **You can consider various options to optimize data storage:**
  - Setting HDFS quotas (chapter 5)
  - Erasure coding (chapter 12)
  - HDFS compression (chapter 12)
  - Balancing data across disks of a DataNode (chapter 13)
  - Balancing data across an HDFS cluster (chapter 13)

Chapter Topics

Data Storage

- Overview
- HDFS Topology and Roles
- HDFS Performance and Fault Tolerance
- HDFS and Hadoop Security Overview
- Working with Namenode UI
- Instructor-Led Demonstration: Namenode User Interface
- Working with HDFS
- **Hands-On Exercise: Working with HDFS**
- HBase Overview
- Kudu Overview
- Cloud Storage Overview
- Essential Points
- Hands-On Exercise: Storing Data in Amazon S3

Hands-On Exercise: Working with HDFS

- **In this exercise, you will practice working with HDFS and add a new DataNode to your cluster**
- **Please refer to the Hands-On Exercise Manual for instructions**

Chapter Topics

Data Storage

- Overview
- HDFS Topology and Roles
- HDFS Performance and Fault Tolerance
- HDFS and Hadoop Security Overview
- Working with Namenode UI
- Instructor-Led Demonstration: Namenode User Interface
- Working with HDFS
- Hands-On Exercise: Working with HDFS
- **HBase Overview**
- Kudu Overview
- Cloud Storage Overview
- Essential Points
- Hands-On Exercise: Storing Data in Amazon S3

HBase Overview

- **HBase is a NoSQL database that runs on top of HDFS**
- **HBase is:**
  - Highly available and fault tolerant
  - Very scalable, and can handle high throughput
  - Able to handle massive tables with ease
  - Well suited to sparse rows where the number of columns varies
  - An open-source Apache project
  - HDFS provides:
    - Fault tolerance
    - Scalability

What Differentiates HBase?

- **HBase helps solve data access issues where random access is required**
- **HBase scales easily, making it ideal for Big Data storage and processing needs**
- **Columns in an HBase table are defined dynamically, as required**

HBase Usage Scenarios (1)

- **High capacity**
  - Massive amounts of data
  - Hundreds of terabytes to multiple petabytes
- **High write throughput - 60,000 ops/second on a 5 node cluster**
  - Servicing search requests
  - Message storage
  - Event stream storage
  - Metrics storage
- **High read throughput - 6,000 ops/second on a 5 node cluster**
  - User profile cache
  - Banking system: Accessing / viewing acct. statements
  - SMS message storage / retrieval for high number of concurrent users
- **Reference:** `Operational Database Performance Improvements in CDP Private Cloud`

HBase Usage Scenarios (2)

- **Scalable in-memory caching**
  - Adding nodes adds to available cache
- **Large amount of stored data, but queries often access a small subset**
  - Data is cached in memory to speed up queries by reducing disk I/O
- **Data layout**
  - HBase excels at key lookup
  - No penalty for sparse columns

HBase Use Cases

- **You can use HBase in CDP Public Cloud alongside your on-prem HBase clusters for disaster recovery use cases**
- **As an operational data store, you can run your applications on top of HBase**
- **Some of the other use cases of HBase in CDP include:**
  - Support mission-important/mission-critical scale-out applications
  - Query data with millisecond latency
  - Operationalize AI/Machine Learning to drive revenue or manage operational cost
  - Bring together data spanning sources, schemas and data types and leverage in your applications
  - Use as a small file store, for example, use HBase to store logs

HBase Training Course

- **Installation of HBase is performed by adding the HBase Service to the cluster**
- **HBase is a very large subject. For details on use and administration of this complex system:**
  - `Click here for:` Cloudera Training for Apache HBase
  - Learning topics include:
    - The use cases and usage occasions for HBase, Hadoop, and RDBMS
    - Using the HBase shell to directly manipulate HBase tables
    - Designing optimal HBase schemas for efficient data storage and recovery
    - How to connect to HBase using the Java API to insert and retrieve real time data
    - Best practices for identifying and resolving performance bottlenecks
  - This course is appropriate for developers and administrators who intend to use HBase

Chapter Topics

Data Storage

- Overview
- HDFS Topology and Roles
- HDFS Performance and Fault Tolerance
- HDFS and Hadoop Security Overview
- Working with Namenode UI
- Instructor-Led Demonstration: Namenode User Interface
- Working with HDFS
- Hands-On Exercise: Working with HDFS
- HBase Overview
- **Kudu Overview**
- Cloud Storage Overview
- Essential Points
- Hands-On Exercise: Storing Data in Amazon S3

Kudu Overview

- **Columnar storage manager**
- **Kudu's benefits include:**
  - Fast processing of OLAP workloads
  - Integration with MapReduce, Spark, and other Hadoop ecosystem components
  - Tight integration with Apache Impala
  - Strong performance for running sequential and random workloads
  - Easy administration and management through Cloudera Manager
- **Kudu gives you capability to stream inputs with near real-time availability**

Kudu Use Cases

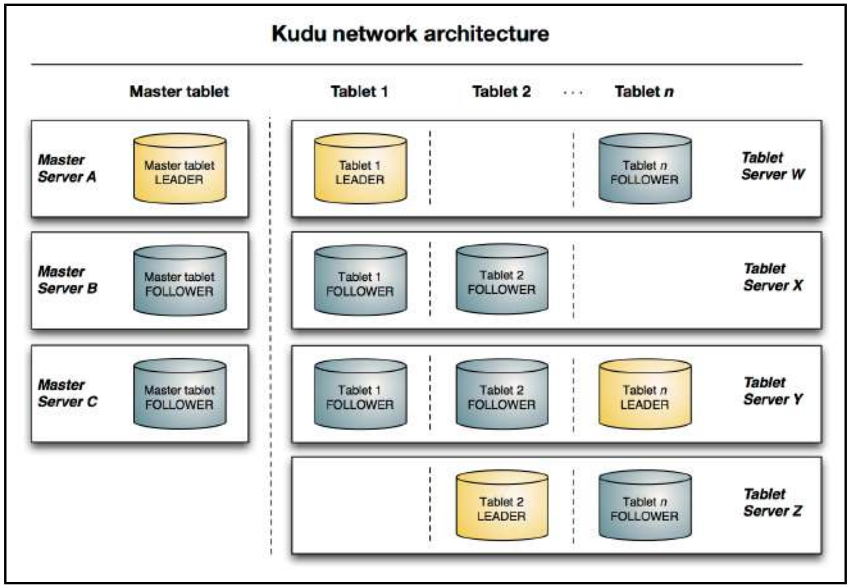
- **Streaming input with near real time availability**
- **Time-Series application with widely varying access patterns**
- **Data scientists can develop predictive learning models from large sets of data that need to be updated or modified often**
- **Combining Data in Kudu with legacy systems using Impala without the need to change legacy systems**

Kudu Concepts

- **Columnar datastore**
- **Raft consensus algorithm**
- **Table - where the data is stored**
- **Tablet - a contiguous segment of a table**
- **Tablet Server - stores and serves tablets to clients**
- **Catalog table - central location of metadata storing information about tables and tablets**

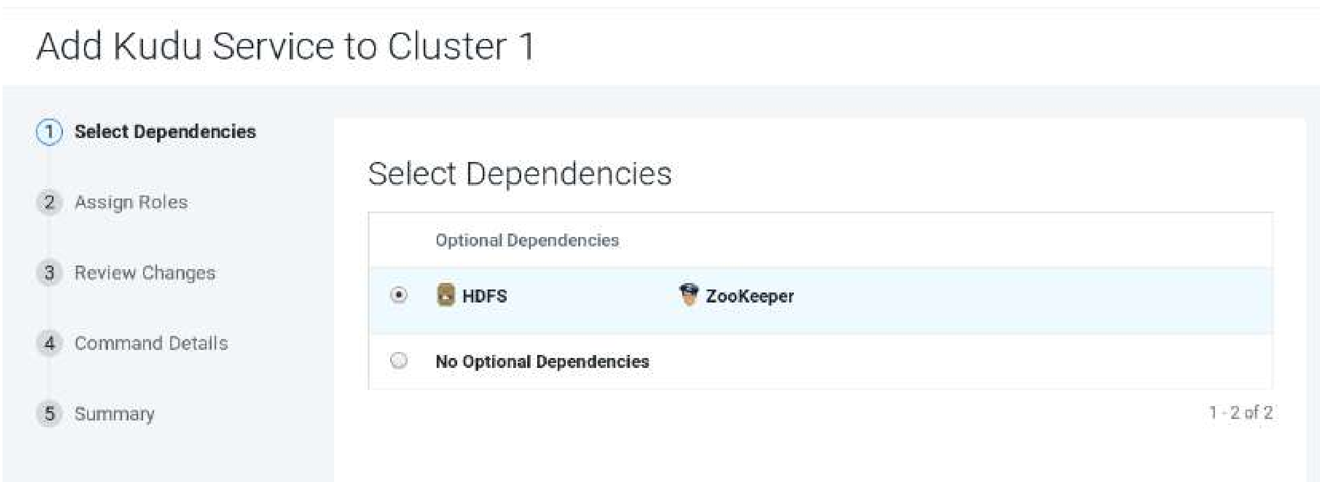
Kudu Architecture

- Diagram in this topic shows a Kudu cluster with three masters and multiple tablet servers
- Each tablet server serves multiple tablets
- Raft consensus is used to allow for both leaders and followers
- A tablet server can be a leader for some tablets and a follower for others
- Leaders are shown in gold, while followers are shown in grey



Kudu Service

- **Kudu can be installed by adding the service using Cloudera Manager**



Kudu Training - Private or On-Demand

- **Through instructor-led discussion, as well as hands-on exercises, participants will learn topics including:**
  - A high-level explanation of Kudu
  - How does it compares to other relevant storage systems and which use cases
  - Learn about Kudu's architecture as well as how to design tables
  - Learn data management techniques using Impala
  - Develop Apache Spark applications with Apache Kudu

Chapter Topics

Data Storage

- Overview
- HDFS Topology and Roles
- HDFS Performance and Fault Tolerance
- HDFS and Hadoop Security Overview
- Working with Namenode UI
- Instructor-Led Demonstration: Namenode User Interface
- Working with HDFS
- Hands-On Exercise: Working with HDFS
- HBase Overview
- Kudu Overview
- **Cloud Storage Overview**
- Essential Points
- Hands-On Exercise: Storing Data in Amazon S3

Cloud Storage

- **To assess the hardware and resource allocations for your cluster:**
  - Before cloud computing, companies had to store all their data and software on their own hard drives and servers
  - The bigger the company, the more storage needed
  - This way of treating data is not scalable at speed
  - Cloud technology means that companies can scale and adapt
  - Companies can accelerate innovation, drive business agility, streamline operations, and reduce costs

Object Store Support

- **Benefits of Object Store**
  - Durability and scalability
  - Cost
- **Challenge of Object Store vs HDFS**
  - Higher latency
- **Cloudera support of Object Store**
  - Impala on Amazon S3
  - Spark on Amazon S3
  - Hive on Amazon S3
  - Hive-on-Tez on Amazon S3
  - S3a connector (such as, for `distcp` copying between HDFS and S3)

Cloud Storage Connections

- **Integration of a CDP cluster to object storage services is through cloud storage connectors**
- **Cloud connectors are included with CDP**
- **Use case examples:**
  - Collect data for analysis and then load it into Hadoop ecosystem applications such as Hive or Spark directly from cloud storage services
  - Persist data to cloud storage services for use outside of CDP clusters
  - Copy data stored in cloud storage services to HDFS for analysis and then copy back to the cloud when done
  - Share data between multiple CDP clusters – and between various external non-CDP systems
  - Back up CDP clusters using `distcp`
- **The cloud object store connectors are implemented as modules whose libraries are automatically placed on the classpath**

Amazon S3 Cloud Storage

- **Amazon S3 is an object store**
  - The S3A connector implements the Hadoop filesystem interface
  - Through the interface you can see a filesystem view of buckets
  - Buckets are used to organize S3 cloud data
  - Applications can access data stored in buckets with `s3a://bucket/dir/files`
- **S3 can not be used as a replacement for HDFS as the cluster filesystem in CDP**
- **Amazon S3 can be used as a source and destination of work**

Limitations of S3

- Operations on directories are potentially slow and non-atomic
- Not all file operations are supported
- Data is not visible in the object store until the entire output stream has been written
- Amazon S3 is eventually consistent
- Neither per-file, per-directory permissions, nor ACL's are supported; set up policies in Ranger to include S3 URLs
- Bandwidth between your workload clusters and Amazon S3 is limited and can vary based on network and load

Access to Amazon S3

- **For Apache Hadoop applications to be able to interact with Amazon S3, they must know the AWS access key and the secret key**
- **This can be achieved in multiple ways**
  - Configuration properties (recommended for Private Cloud Base clusters)
  - Environmental variables
  - EC2 instance metadata (if cluster running on EC2)
- **By default, the S3A filesystem client authentication chain is:**
  - The AWS login details are looked for in the job configuration settings
  - The AWS environment variables are then looked for
  - An attempt is made to query the Amazon EC2 Instance Metadata Service to retrieve credentials published to EC2 VMs

Azure Data Lake Storage

- **ADLS more closely resembles native HDFS behavior than S3**
  - Provides consistency, directory structure, and POSIX-compliant ACLs
- **Accessible though an HDFS-compatible API**
- **Do not configure ADLS as the *default* filesystem**
- **Overview of how to connect**
  - Create a service principal in the Azure portal
  - Grant the service principal permission to access the ADLS account
  - Configure cluster access to your ADLS account

Google Cloud Storage

- **Objects stored in buckets**
- **Different storage classes available (multi-regional, regional, nearline, coldline)**
- **Follow the steps documented at [Google's cloud computing site] to**
  - Ensure the GCS API is enabled
  - Create a service account
  - Obtain the credentials to connect to cloud storage buckets

Accessing Data in Object Storage

- **Example: Storing Impala/Hive table data in S3**

```
CREATE EXTERNAL TABLE ages (
   name STRING, age INT)
 ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
 LOCATION 's3a://myBucket/accounts/';

 SELECT * FROM ages WHERE age > 50;
```

- **Example: Reading and writing ADLS data in Spark**

```
accountsDF = spark.read.csv("adl://myBucket/accounts/")
 accountsDF.where("age > 18").write.
   save("adl://myBucket/adult_accounts/")
```

Options for Connecting to S3

- **Provide the credentials on the command line**
  - Example:

```
$ hadoop distcp \
 -Dfs.s3a.access.key=myAccessKey \
 -Dfs.s3a.secret.key=mySecretKey \
 /user/hdfs/mydata s3a://myBucket/mydata_backup
```

- **Use a Hadoop Credential Provider**
  - Stores encrypted credentials in a Java keystore in HDFS
  - Create the credential provider using the `hadoop credential create` command
  - For *per-user* access, pass the path to the keystore at job run time
  - For *system-wide* access, reference the keystore in the advanced configuration snippet

Options for Connecting to ADLS

- **Configure the ADLS Connector service (from the External Accounts page in Cloudera Manager)**
- **User-supplied key—pass credentials on command line**

```
$ hadoop command
 -Dfs.adls.oauth2.access.token.provider=ClientCredential \
 -Dfs.adls.oauth2.client.id=CLIENT_ID \
 -Dfs.adls.oauth2.credential=CLIENT_SECRET \
 -Dfs.adls.oauth2.refresh.url=REFRESH_URL \
 adl://store.azuredatalakestore.net/src \
 hdfs://namenode/target-location
```

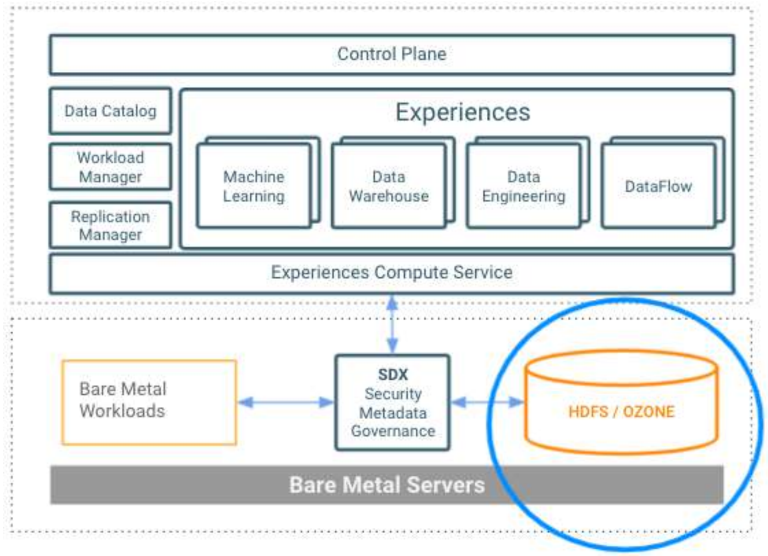
- **Single master key for cluster-wide access (grants access to all users)**
  - Set connection properties in **Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml**
- **User-supplied key stored in a Hadoop Credential Provider**
  - Provision a credential store for the access key and secret key in HDFS
- **Create a Credential Provider and reference the keystore**

Ozone Preview

- **Ozone was designed to address the scale limitation of HDFS with respect to small files**
- **On current Private Cloud Base hardware, HDFS has a limit of:**
  - 350 million files
  - 700 million file system objects
- **Ozone's architecture addresses these limitations**
- **Hive on Ozone works faster than HDFS**
- **Ozone can function effectively in containerized environments such as Kubernetes and YARN**
- **When using CDP Private Cloud, Ozone can offer a solution to the small files problem**
- **See Appendix to this course titled *Ozone Review*.**

Apache Ozone

- Ozone is a distributed Key Value Object Store that provide 20x the scalability of traditional HDFS
- Increase storage node configurations by 350% reducing storage costs by 50% and cost per TB
- Ozone is designed and optimized for Big Data workloads providing the scale of a modern object store
- Native support for S3 API that enables cloud native architectures



Chapter Topics

Data Storage

- Overview
- HDFS Topology and Roles
- HDFS Performance and Fault Tolerance
- HDFS and Hadoop Security Overview
- Working with Namenode UI
- Instructor-Led Demonstration: Namenode User Interface
- Working with HDFS
- Hands-On Exercise: Working with HDFS
- HBase Overview
- Kudu Overview
- Cloud Storage Overview
- **Essential Points**
- Hands-On Exercise: Storing Data in Amazon S3

Essential Points (1)

- **HDFS distributes blocks of data across a set of machines**
  - Supports running computations where the data is located
  - HDFS works best with a modest number of large files
  - Block size is configurable, default is 128MB
- **The NameNode maintains HDFS metadata in memory and stores it on disk**
  - Edit logs and checkpointing make HDFS more efficient
- **HDFS provides fault tolerance with built-in data redundancy**
  - The number of replicas per block is configurable (default is three)
- **Many options available for connecting to HDFS**
  - Command-line interface, Java API, and multiple web UIs

Essential Points (2)

- **HBase is a NoSQL database that runs on top of HDFS**
- **Kudu is a big data relational type columnar storage**
- **Object Storage can be utilized**
  - Amazon S3 storage
  - Microsoft Azure Data Lake Storage

Chapter Topics

Data Storage

- Overview
- HDFS Topology and Roles
- HDFS Performance and Fault Tolerance
- HDFS and Hadoop Security Overview
- Working with Namenode UI
- Instructor-Led Demonstration: Namenode User Interface
- Working with HDFS
- Hands-On Exercise: Working with HDFS
- HBase Overview
- Kudu Overview
- Cloud Storage Overview
- Essential Points
- **Hands-On Exercise: Storing Data in Amazon S3**

Hands-On Exercise: Storing Data in Amazon S3

- **In this exercise, you will you will copy data from AWS S3 to HDFS**
- **Please refer to the Hands-On Exercise Manual for instructions**

Data Ingest

Chapter 6

Course Chapters

- Introduction
- Cloudera Data Platform
- CDP Private Cloud Base Installation
- Cluster Configuration
- Data Storage
- **Data Ingest**
- Data Flow
- Data Access and Discovery
- Data Compute
- Managing Resources
- Planning Your Cluster
- Advanced Cluster Configuration
- Cluster Maintenance
- Cluster Monitoring
- Cluster Troubleshooting
- Security
- Private Cloud / Public Cloud
- Conclusion
- Appendix: Cloudera Manager API
- Appendix: Ozone Overview

Data Ingest

After completing this chapter, you will be able to

- **Describe the features, pros, and cons of various methods for ingesting data into HDFS**
- **Import data into HDFS from external file systems**
- **Import data from a relational-database into HDFS using Sqoop**
- **Describe the features of NiFi to transform and ingest data**
- **Summarize how time requirements and data sources determine the best data ingest tool**

Chapter Topics

Data Ingest

- **Data Ingest Overview**
- File Formats
- Ingesting Data using File Transfer or REST Interfaces
- Importing Data from Relational Databases with Apache Sqoop
- Hands-On Exercise: Importing Data Using Sqoop
- Ingesting Data Using NiFi
- Instructor-Led Demonstration: NiFi User Interface
- Best Practices for Importing Data
- Essential Points
- Hands-On Exercise: NiFi Verification

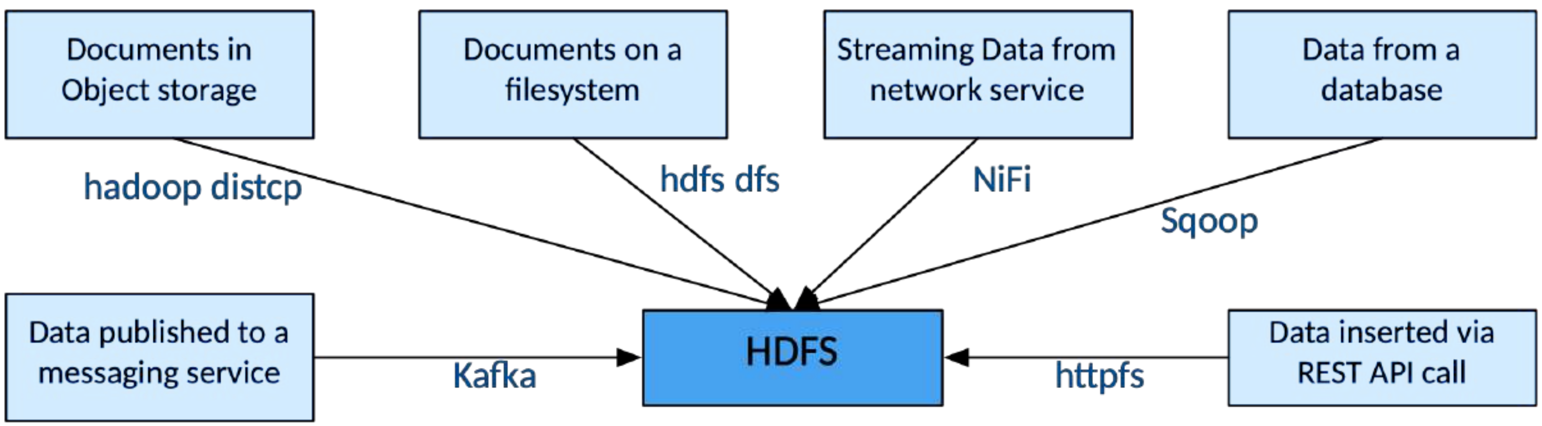
Data Ingestion Introduction

- **Data ingestion and transformation is the first step in big data projects**
  - Transformation can be done during the initial ingest or later
- **Key tools for data ingest**
  - File transfer
  - REST interfaces
  - Apache Sqoop
  - Apache NiFi
  - Apache Kafka *
  - Apache Spark Streaming *
- **These same tools can also be used to copy data out of the cluster**

* Not covered in this chapter

Example Data Sources

- **Example Data Sources**
  - Web server log files
  - Financial transactions
  - Mobile phone activations, device status
  - RDBMS customer records
  - Sensor-generated data associated with IoT
  - Social media data collection
- **Different tools exist for ingesting data from different sources**



Chapter Topics

Data Ingest

- Data Ingest Overview
- **File Formats**
- Ingesting Data using File Transfer or REST Interfaces
- Importing Data from Relational Databases with Apache Sqoop
- Hands-On Exercise: Importing Data Using Sqoop
- Ingesting Data Using NiFi
- Instructor-Led Demonstration: NiFi User Interface
- Best Practices for Importing Data
- Essential Points
- Hands-On Exercise: NiFi Verification

Hadoop File Formats

- **HDFS can store *any* type of file**
- **Other Hadoop tools support a variety of file formats such as**
  - Text
    - Including CSV, JSON, and plain text files
  - Apache Parquet
    - Optimized, binary, columnar storage of structured data
  - Apache Avro Data Format
    - Apache Avro is a data serialization system
    - Stores data in a compact, structured, binary, row-oriented format
  - Apache ORC
    - Optimized for both column-based and record-based access, specific to the Hadoop ecosystem
  - SequenceFile Format
    - The original Hadoop binary storage format
- **Not all Hadoop components support all formats natively**

Apache Parquet

- **Parquet is a very common storage format**
  - Supported by Sqoop, Spark, Hive, Impala, and other tools
- **Key Features**
  - Optimized binary storage of structured data
  - Schema metadata is embedded in the file
  - Efficient performance and size for large amounts of data
  - Parquet works well with Impala
- **Use `parquet-tools` to view Parquet file schema and data**
  - Use `head` to display the first few records

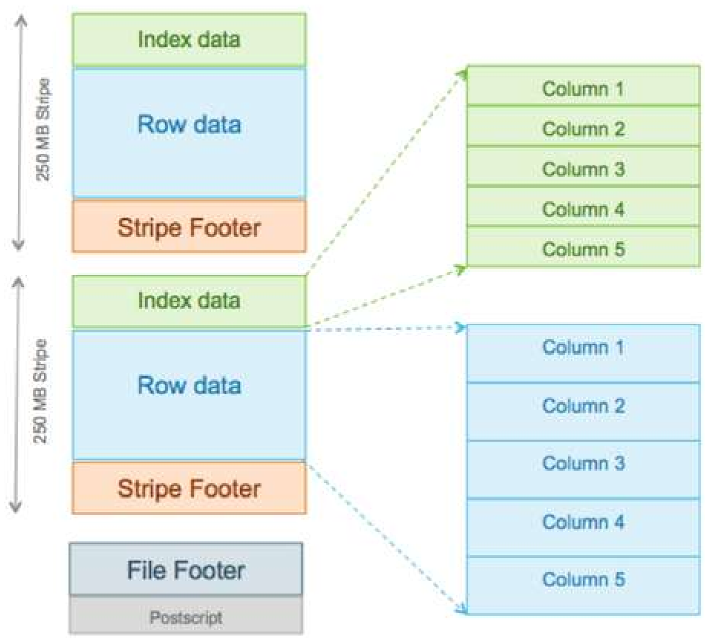
```
$ parquet-tools head /mydir/mydatafile.parquet
```

  - Use `schema` to view the schema

```
$ parquet-tools schema /mydir/mydatafile.parquet
```

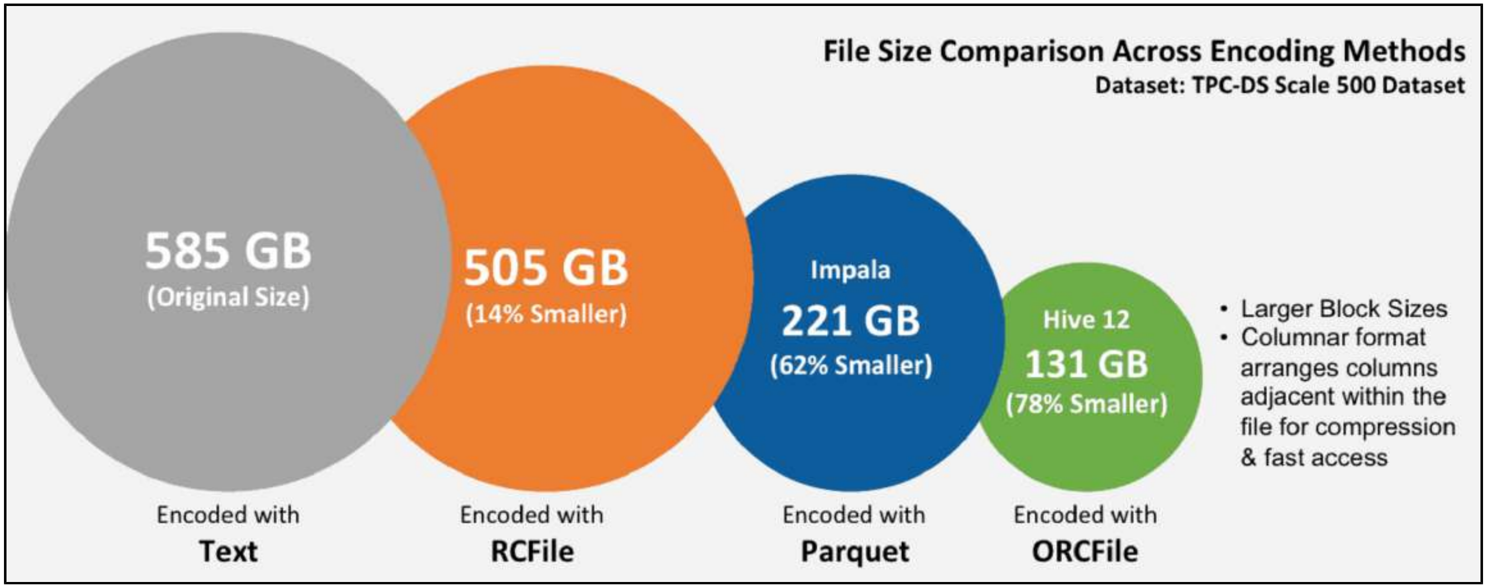
ORC File Format (Optimized Record Columnar)

- ORC is a row columnar data format
- Highly optimized for reading, writing, and processing data in Hive
- ORC files are made of stripes of data
- Each stripe contains index, row data, and footer



ORC File Format

- **Designed to speed up usage of Hive**
- **Key statistics are conveniently cached (count, max, min, and sum)**
- **You can insert data into an ORC Hive table**
- **Hive 3 improves the ACID qualities and performance of transactional tables**



ORC File Format

- **ORC reduces I/O overhead by accessing only the columns that are required for the current query**
- **Columnar storage for high performance**
  - Efficient reads: Break into large “stripes” of data
- **Fast filtering**
  - Built-in aggregates per block (MIN, MAX, COUNT, SUM, and so on)
  - Built-in light indexing
- **Efficient compression**
  - Decompose complex row types into primitives
  - Block-mode compression based on data type

Data Compression

- **Compression reduces amount of disk space required to store data**
- **Trades off between CPU time and bandwidth/storage space**
  - Aggressive algorithms are slower but save more space
  - Less aggressive algorithms save less space but are much faster
- **Can significantly improve performance**
  - Many Hadoop jobs are I/O-bound
  - Using compression allows you to handle more data per I/O operation
  - Compression can also improve the performance of network transfers
- **Supported formats include BZip2, and Snappy**
  - Not all file formats support all compression formats

Choosing and Configuring Data Compression

- **Guidelines for Choosing a Compression Type**
  - GZIP compression uses more CPU resources than Snappy, but provides a higher compression ratio
  - GZip is often a good choice for cold data, which is accessed infrequently
  - Snappy is a better choice for hot data, which is accessed frequently
  - BZip2 can also produce more compression than GZip for some types of files, at the cost of some speed when compressing and decompressing
  - HBase does not support BZip2 compression

Chapter Topics

Data Ingest

- Data Ingest Overview
- File Formats
- **Ingesting Data using File Transfer or REST Interfaces**
- Importing Data from Relational Databases with Apache Sqoop
- Hands-On Exercise: Importing Data Using Sqoop
- Ingesting Data Using NiFi
- Instructor-Led Demonstration: NiFi User Interface
- Best Practices for Importing Data
- Essential Points
- Hands-On Exercise: NiFi Verification

File Transfer Tools

- `hdfs dfs -put` or `hdfs dfs -get`
  - For copying data between the Linux filesystem and HDFS
- `hadoop distcp`
  - For copying data between object storage and HDFS
  - For copying data between and within clusters
- **Mountable HDFS (Fuse-DFS, NFS Gateway)**
  - HDFS can be mounted as a remote Linux file system
  - The NFS Gateway is sometimes used to move data from Windows clients
- **Limitations**
  - If errors occur during transfer, the transfer fails
  - No data transformation as part of the ingest process
  - Can only specify a single target location

The WebHDFS REST API

- **WebHDFS provides an HTTP/HTTPS REST interface to HDFS**
  - REST: REpresentational State Transfer
  - WebHDFS supports reads and writes both from and to HDFS
  - Can be accessed from within a program or script
  - Can be accessed using command-line tools such as `curl` and `wget`
- **Installs with the HDFS service in Cloudera Manager**
  - Enabled by default (`dfs.webhdfs.enabled`)
- **Clients must be able to access to every DataNode in the cluster**
- **Does not support HDFS high availability deployments**

The HttpFS REST API

- **Provides an HTTP/HTTPS REST interface to HDFS**
  - The interface is identical to the WebHDFS REST interface
- **Optional part of the HDFS service**
  - Add the HttpFS role to the HDFS service deployment
    - Installs and configure an HttpFS server
    - Enables proxy access to HDFS for the `httpfs` user
- **Client only needs access to the HttpFS server only**
  - The HttpFS server then accesses HDFS
- **Supports HDFS HA deployments**

WebHDFS/HttpFS REST Interface Examples

- **These examples will work with either WebHDFS or HttpFS**
  - For WebHDFS, specify the NameNode host and port (default: 9870)
  - For HttpFS, specify the HttpFS server and port (default: 14000)
- **Open and get the `shakespeare.txt` file**

```
$ curl -i -L "http://host:port/webhdfs/v1/tmp/\
shakespeare.txt?op=OPEN&ampuser.name=training"
```

- **Make the `mydir` directory**

```
$ curl -i -X PUT "http://host:port/webhdfs/v1/user/\
training/mydir?op=MKDIRS&user.name=training"
```

Chapter Topics

Data Ingest

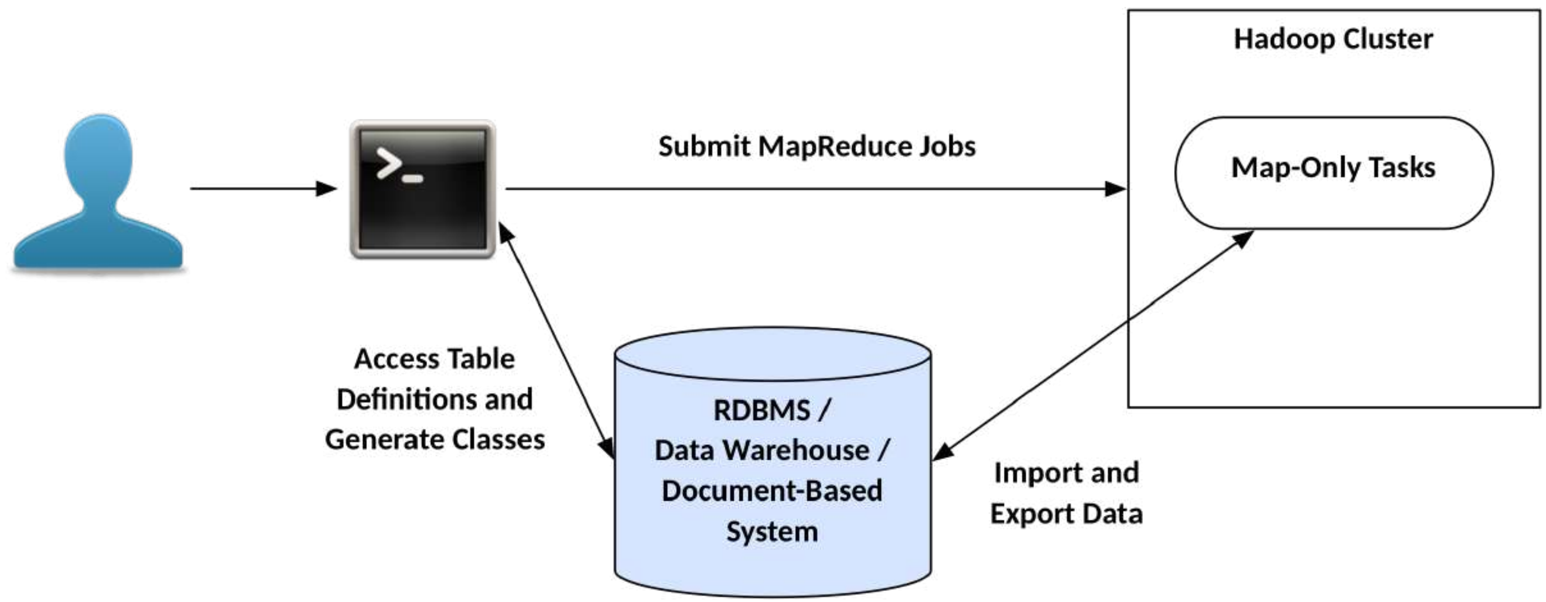
- Data Ingest Overview
- File Formats
- Ingesting Data using File Transfer or REST Interfaces
- **Importing Data from Relational Databases with Apache Sqoop**
- Hands-On Exercise: Importing Data Using Sqoop
- Ingesting Data Using NiFi
- Instructor-Led Demonstration: NiFi User Interface
- Best Practices for Importing Data
- Essential Points
- Hands-On Exercise: NiFi Verification

What is Apache Sqoop?

- **Sqoop is "the SQL-to-Hadoop database import tool"**
  - Open source Apache project, originally developed at Cloudera
  - Included in CDP
- **Imports and exports data between database systems and HDFS**
- **Supports several Hadoop file types**
  - Delimited text files such as CSV
  - Hive tables
  - Avro data format files
  - Parquet files
  - HBase tables
- **Uses JDBC (Java Database Connectivity) to connect to database**
  - JDBC drivers available for common RDBMSs as a separate download
    - For example: MySQL, Oracle, SQL Server, PostgreSQL

How Does Sqoop Work?

1. **Generates a Java class to import data**
2. **Runs a Hadoop MapReduce job**
   - Map-only job
   - By default, four mappers connect to the RDBMS
   - Each mapper imports a quarter of the data



Sqoop Features

- **Imports a single table or all tables in a database**
- **Can specify which *rows* to import by using a `WHERE` clause**
- **Can specify which *columns* to import**
- **Allows an arbitrary `SELECT` statement**
- **Can automatically create a Hive table based on the imported data**
- **Supports incremental imports**

Sqoop Usage Examples

- **List all databases**

```
$ sqoop list-databases --username fred -P \
  --connect jdbc:mysql://dbserver.example.com/
```

- **List all tables in the `world` database**

```
$ sqoop list-tables --username fred -P \
  --connect jdbc:mysql://dbserver.example.com/world
```

- **Import all tables in the `world` database**

```
$ sqoop import-all-tables --username fred --password derf \
  --connect jdbc:mysql://dbserver.example.com/world
```

- **Sqoop import options:**
  - To designate the table: --table movie
  - To identify the field delimiter :--fields-terminated-by ’\t’

- To identify the location to place files in hdfs: --target-dir /tmp/moviedata/movie

Chapter Topics

Data Ingest

- Data Ingest Overview
- File Formats
- Ingesting Data using File Transfer or REST Interfaces
- Importing Data from Relational Databases with Apache Sqoop
- **Hands-On Exercise: Importing Data Using Sqoop**
- Ingesting Data Using NiFi
- Instructor-Led Demonstration: NiFi User Interface
- Best Practices for Importing Data
- Essential Points
- Hands-On Exercise: NiFi Verification

Hands-On Exercise: Importing Data Using Sqoop

- **In this exercise, you will Install Sqoop and ingest data from a MySQL server**
- **Please refer to the Hands-On Exercise Manual for instructions**

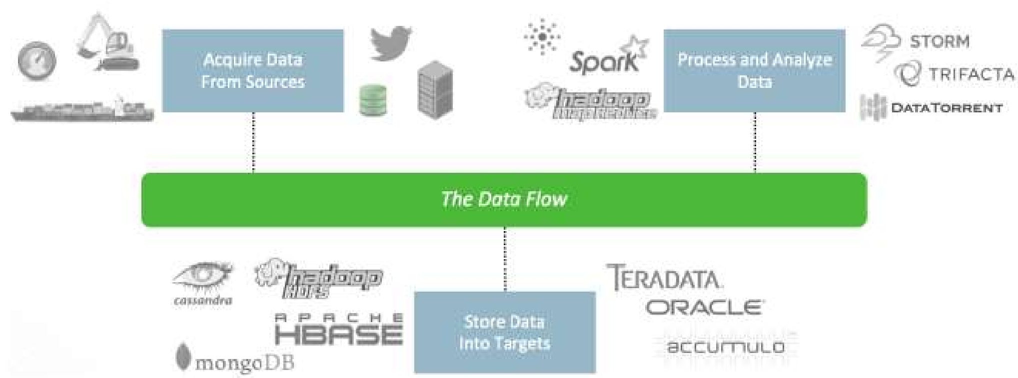
Chapter Topics

Data Ingest

- Data Ingest Overview
- File Formats
- Ingesting Data using File Transfer or REST Interfaces
- Importing Data from Relational Databases with Apache Sqoop
- Hands-On Exercise: Importing Data Using Sqoop
- **Ingesting Data Using NiFi**
- Instructor-Led Demonstration: NiFi User Interface
- Best Practices for Importing Data
- Essential Points
- Hands-On Exercise: NiFi Verification

NiFi Basics

- Apache NiFi is a general purpose tool for data ingest
- NiFi automates the movement of data between disparate data sources
- Making data ingestion fast, easy and secure
- NiFi is data source agnostic
- Trace your data in real time
- Provides a web-based UI for creating, monitoring, and controlling data flow
- Visual Programming paradigm allows for non-code implementation

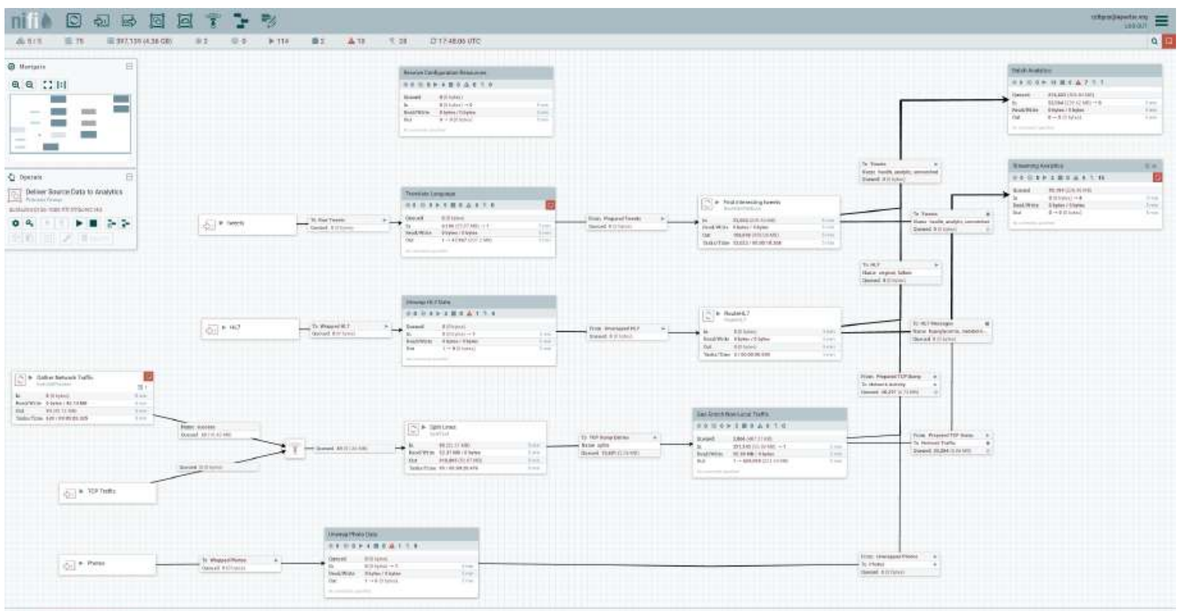


NiFi Key Features

- Guaranteed delivery
- Data buffering with back pressure and pressure release
- Control quality of service for different flows
- Data provenance
- Flow templates
- Security and multi-tenant authorization
- Clustering
- NiFi is considered to be the successor of Flume

Install NiFi

- **Use the Add Service wizard to install NiFi**
- **NiFi is part of the CFM parcel**
- **When selecting the set of dependencies for NiFi, you must select ZooKeeper**
- **If the cluster is not configured to use JDK 8, use the Java Home Path Override configuration field**
- **Specify the security settings appropriate for your installation**



Chapter Topics

Data Ingest

- Data Ingest Overview
- File Formats
- Ingesting Data using File Transfer or REST Interfaces
- Importing Data from Relational Databases with Apache Sqoop
- Hands-On Exercise: Importing Data Using Sqoop
- Ingesting Data Using NiFi
- **Instructor-Led Demonstration: NiFi User Interface**
- Best Practices for Importing Data
- Essential Points
- Hands-On Exercise: NiFi Verification

Instructor-Led Demonstration of NiFi to Ingest Data

- **The instructor will demonstrate how to build a dataflow in NiFi to ingest data into HDFS**
- **Additional administration information regarding NiFi will be covered in the following chapter**

Chapter Topics

Data Ingest

- Data Ingest Overview
- File Formats
- Ingesting Data using File Transfer or REST Interfaces
- Importing Data from Relational Databases with Apache Sqoop
- Hands-On Exercise: Importing Data Using Sqoop
- Ingesting Data Using NiFi
- Instructor-Led Demonstration: NiFi User Interface
- **Best Practices for Importing Data**
- Essential Points
- Hands-On Exercise: NiFi Verification

What Do Others See as Data Is Imported to HDFS?

- **When a client starts to write data to HDFS**
  - The NameNode marks the file as existing with size zero
  - Other clients will see it as an empty file
- **After each block is written, other clients will see that block**
  - They will see the file growing as it is being created, one block at a time
- **Other clients may begin to process a file as it is being written**
  - Clients will read a partial file
  - Not a best practice

Best Practices: Importing Data to HDFS

- **Import data into a temporary directory**
- **Move data to target directory after file is imported**
  - Moving is an atomic operation
  - The blocks on disk are not moved
    - Only requires an update of the NameNode's metadata
- **Many organizations standardize on a directory structure. Example:**
  - `/incoming/<import_job_name>/<files>`
  - `/for_processing/<import_job_name>/<files>`
  - `/completed/<import_job_name>/<files>`
- **Jobs move the files from `for_processing` to `completed`**

Best Practices: Ingest Frequency

- **Determine the best ingest approach**
  - How soon will the data need to be processed
- ***Less frequent***
  - Periodic batch data dumps
  - Likely storage layer: HDFS or Object storage
  - Likely ingest tools: File transfer or Sqoop
- ***More frequent* (such as less than two minutes)**
  - Streaming data feeds
  - Likely storage layer: HBase, Solr, or Kudu
  - Likely ingest tools: NiFi (with or without Kafka)
- ***Near-real-time***
  - Streaming data feeds
  - Likely storage layer: HBase, Solr, or Kudu
  - Likely ingest tools: Kafka with Spark Streaming

Chapter Topics

Data Ingest

- Data Ingest Overview
- File Formats
- Ingesting Data using File Transfer or REST Interfaces
- Importing Data from Relational Databases with Apache Sqoop
- Hands-On Exercise: Importing Data Using Sqoop
- Ingesting Data Using NiFi
- Instructor-Led Demonstration: NiFi User Interface
- Best Practices for Importing Data
- **Essential Points**
- Hands-On Exercise: NiFi Verification

Essential Points

- **There are many options available for moving data into or out of HDFS**
- **For batch ingest, when transformation is not needed**
  - File transfer tools such as the HDFS CLI and distcp work well
- **A REST interface is available for accessing HDFS**
  - Enable WebHDFS or add the HttpFS role to the HDFS service
  - The REST interface is identical whether you use WebHDFS or HttpFS
- **Use Sqoop to import data from a relational database into HDFS**
- **NiFi automates the movement of data between disparate data sources**
  - Using a no-code visual programming tool

Chapter Topics

Data Ingest

- Data Ingest Overview
- File Formats
- Ingesting Data using File Transfer or REST Interfaces
- Importing Data from Relational Databases with Apache Sqoop
- Hands-On Exercise: Importing Data Using Sqoop
- Ingesting Data Using NiFi
- Instructor-Led Demonstration: NiFi User Interface
- Best Practices for Importing Data
- Essential Points
- **Hands-On Exercise: NiFi Verification**

Hands-On Exercise: NiFi Verification

- **Instructor-Led Exercise of NiFi to verify Nifi**
- **Please refer to the Hands-On Exercise Manual for instructions**

Data Flow

Chapter 7

Course Chapters

Data Flow

After completing this chapter, you will be able to

- **Describe how flow management fits into an enterprise data solution**
- **Summarize how Cloudera Flow Management uses NiFi to manage data flow**
- **Explain the major areas of the NiFi web user interface**
- **Describe typical Kafka use cases**
- **Summarize how Kafka brokers, consumers, and producers work together**

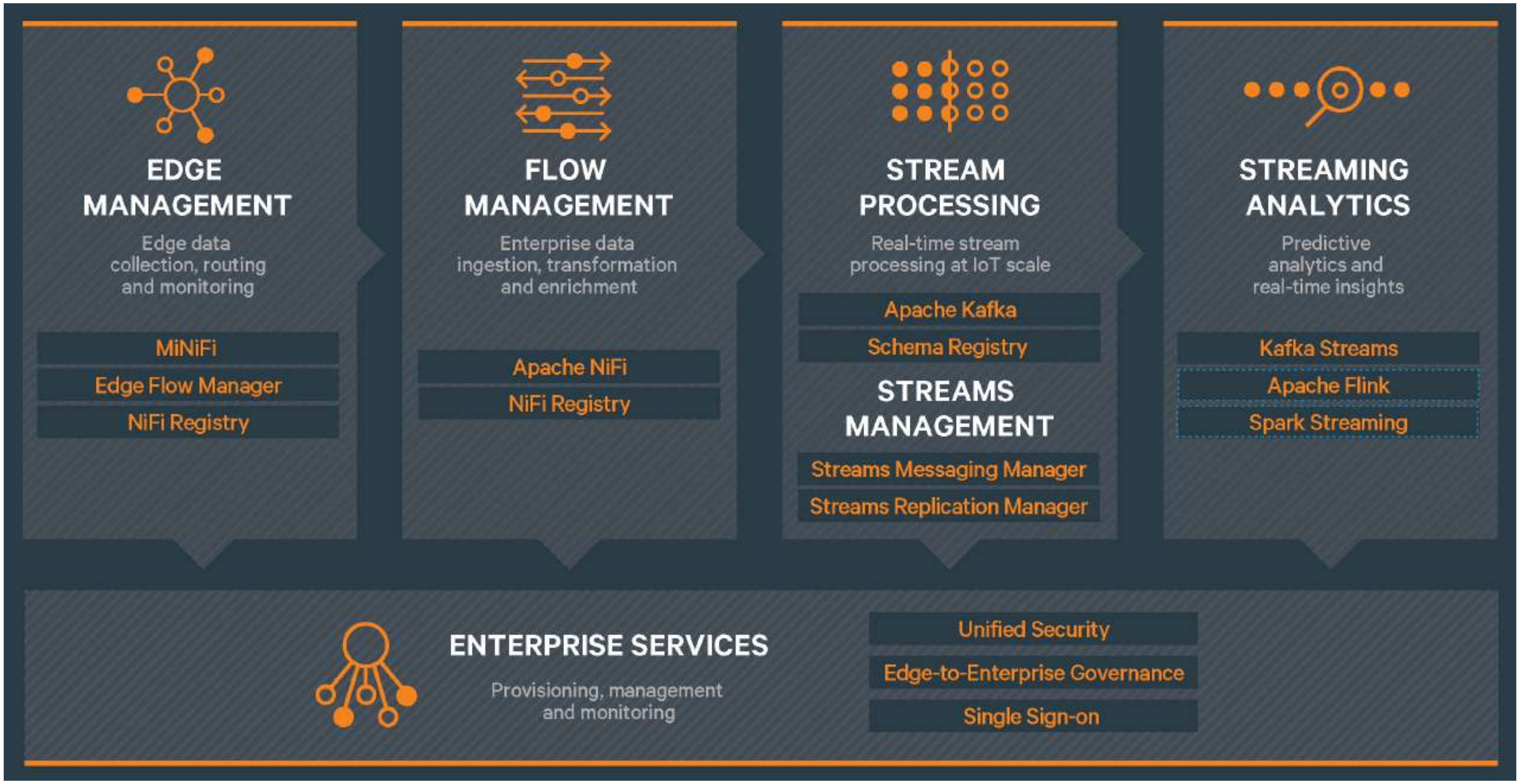
Chapter Topics

Data Flow

- **Overview of Cloudera Flow Management and NiFi**
- NiFi Architecture
- Cloudera Edge Flow Management and MiNiFi
- Instructor-Led Demonstration: NiFi Usage
- Apache Kafka Overview
- Apache Kafka Cluster Architecture
- Apache Kafka Command Line Tools
- Essential Points
- Hands-On Exercise: Working with Kafka

Cloudera Flow Management

- **Cloudera Flow Management (CFM) is an Enterprise Data-in-Motion platform**
  - Scalable, real-time streaming analytics platform
  - Ingests, curates, and analyzes data for key insights and immediate actionable intelligence



Why Use Cloudera Flow Management and NiFi?

- **Runtime configuration of the flow of data**
- **Keeps detailed a history of each data item through entire flow**
- **Extensible through development of custom components**
- **Secure communication with other NiFi instances and external systems**

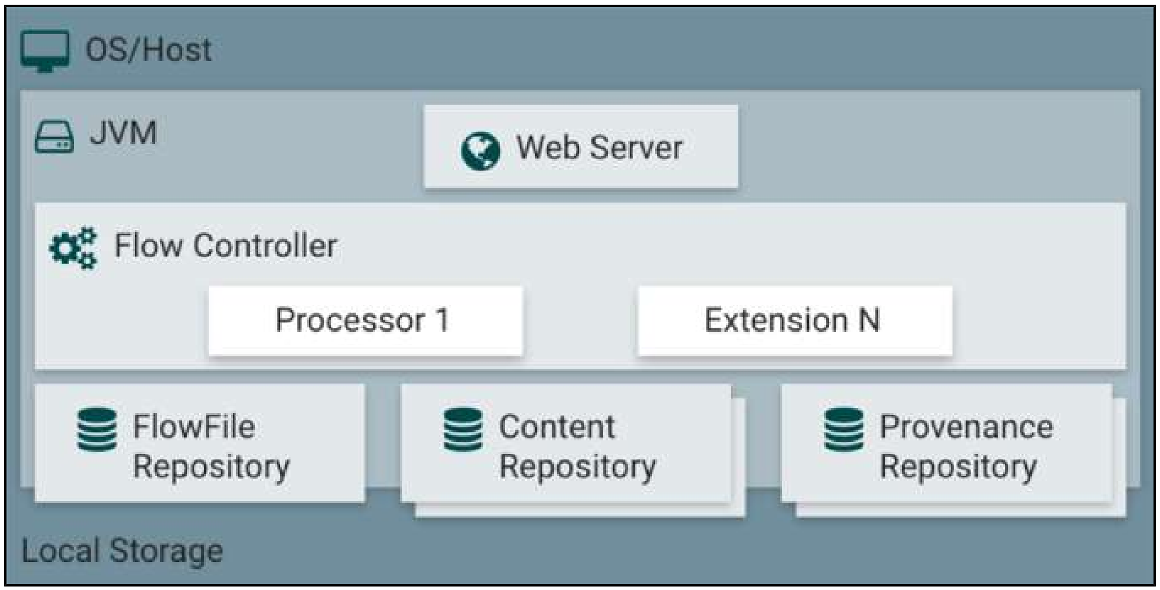
Chapter Topics

Data Flow

- Overview of Cloudera Flow Management and NiFi
- **NiFi Architecture**
- Cloudera Edge Flow Management and MiNiFi
- Instructor-Led Demonstration: NiFi Usage
- Apache Kafka Overview
- Apache Kafka Cluster Architecture
- Apache Kafka Command Line Tools
- Essential Points
- Hands-On Exercise: Working with Kafka

NiFi Architecture

- **NiFi can run on a single node in standalone mode or on multiple nodes in a cluster**
- **Individual nodes have the same basic architecture in both modes**



NiFi Primary Storage Components

- **FlowFile Repository—where NiFi keeps track of the state of active FlowFiles**
  - The default approach is a persistent Write-Ahead Log on a specified disk partition
- **Content Repository—where the actual contents of FlowFiles are stored**
  - The default approach stores blocks of data a file system
  - Supports multiple locations in different physical volumes for performance
- **Provenance Repository—where provenance event data is stored**
  - By default, located on one or more physical disk volumes
  - Event data is indexed and searchable
- **flow.xml.gz—contains information about everything on the canvas**
  - Includes templates, versioning, and controller settings

Primary Components in the JVM

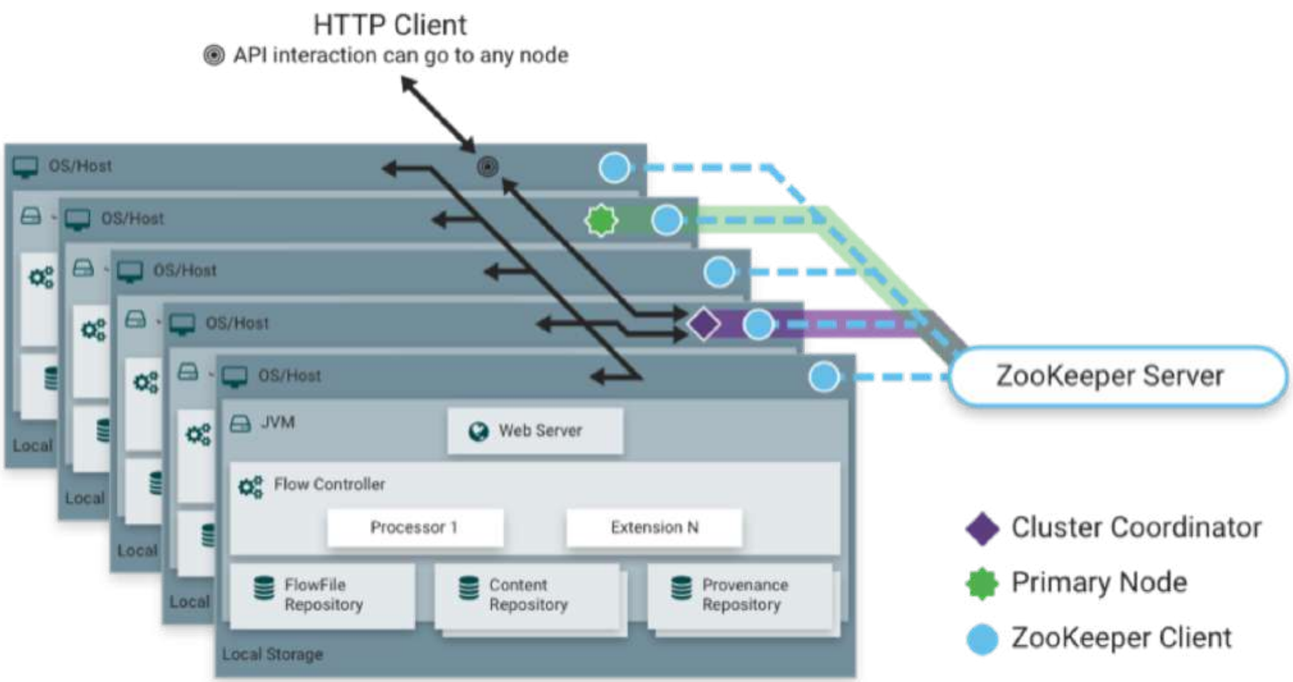
- **NiFi executes within a JVM running on a host**
- **The primary components of NiFi running in the JVM are**
  - Web Server—hosts NiFi’s HTTP-based command and control API
  - Flow Controller—manages threads and schedules execution and resources
    - The “brains” of the the operation
  - Extensions such as custom processors and NiFi plugins

Why Use NiFi in a Cluster?

- **Physical resource exhaustion can occur even with an optimized dataflow**
  - One instance of NiFi on a single server might not be enough to process all required data
- **Installing NiFi in a cluster solves this problem**
  - Spreads the data load across multiple NiFi instances
- **NiFi provides a single interface to**
  - Make dataflow changes and replicate them throughout the cluster
  - Monitor all dataflows running across the cluster

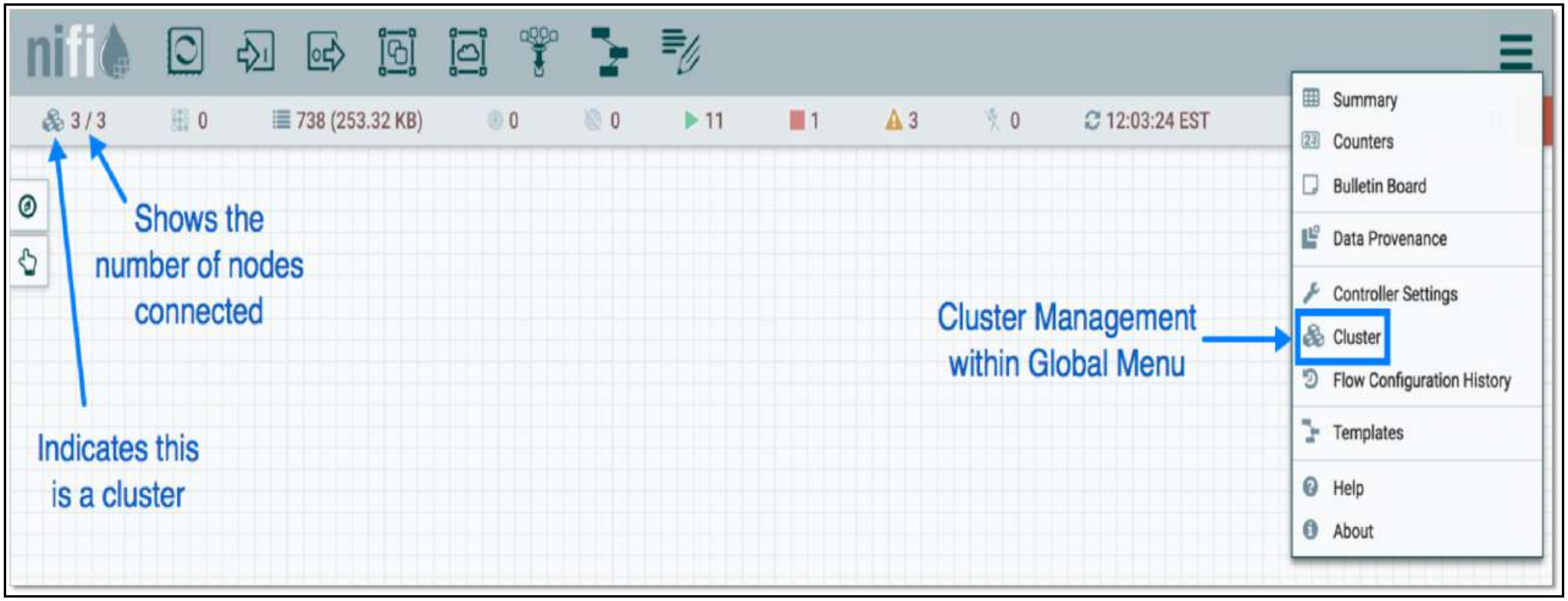
NiFi Clustering Architecture Overview

- A cluster is a set of "nodes"—separate NiFi instances working together to process data
- Each node in the cluster performs the same tasks on a different dataset



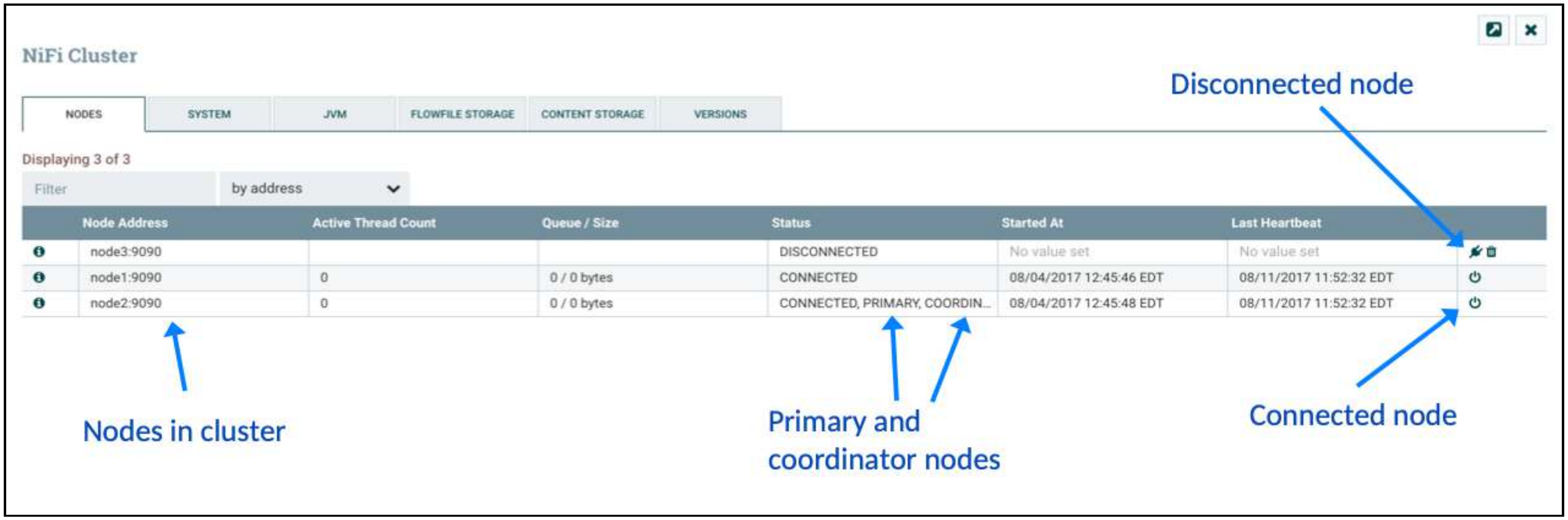
Cluster Management UI (1)

- **On the main canvas page, view the number of nodes in the cluster**
  - Show total count and currently connected count



Cluster Management UI (2)

- **The NiFi Cluster window lets you manage the cluster**
  - View cluster node details, disconnect and remove nodes, and so on



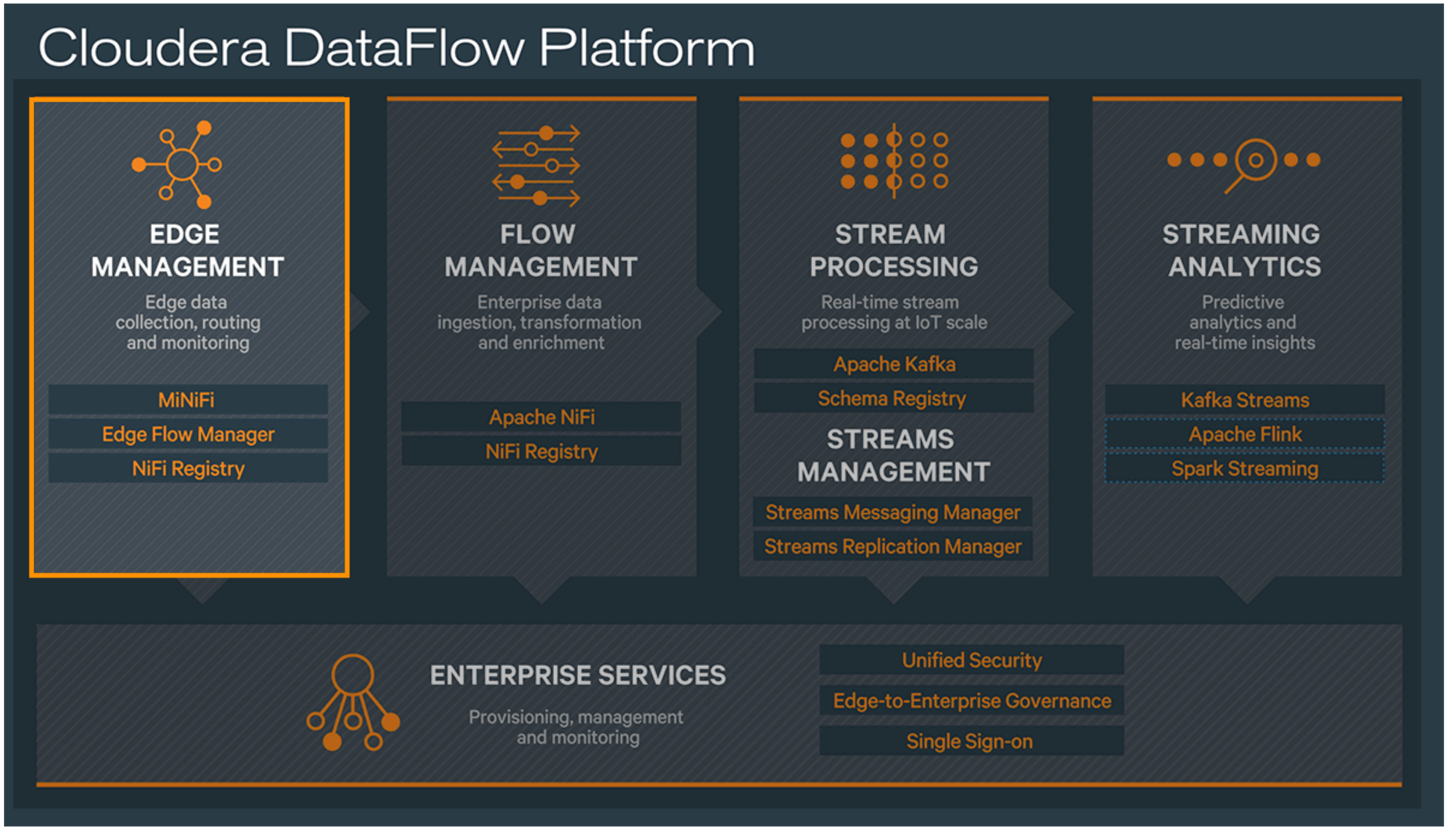
Chapter Topics

Data Flow

- Overview of Cloudera Flow Management and NiFi
- NiFi Architecture
- **Cloudera Edge Flow Management and MiNiFi**
- Instructor-Led Demonstration: NiFi Usage
- Apache Kafka Overview
- Apache Kafka Cluster Architecture
- Apache Kafka Command Line Tools
- Essential Points
- Hands-On Exercise: Working with Kafka

Cloudera Edge Management

- **Cloudera Edge Management is made up of MiNiFi edge agents and an edge management hub—Edge Flow Manager**
  - Manages, controls, and monitors edge agents to collect data from edge devices and push intelligence back to the edge



What is MiNiFi?

- **MiNiFi is focused on collecting data at the source**
  - Sub-project of Apache NiFi

- NiFi lives in the data center
- Runs on enterprise servers

- MiNiFi lives as close to the source of the data as possible
- Agent runs as a guest on that device or system

Key MiNiFi Features

- **Design and deploy**
- **Warm re-deploys**
- **Guaranteed delivery**
- **Data buffering (back pressure)**
- **Security and data provenance**
- **Maintains fine-grained history of data**
- **Extensible**

Chapter Topics

Data Flow

- Overview of Cloudera Flow Management and NiFi
- NiFi Architecture
- Cloudera Edge Flow Management and MiNiFi
- **Instructor-Led Demonstration: NiFi Usage**
- Apache Kafka Overview
- Apache Kafka Cluster Architecture
- Apache Kafka Command Line Tools
- Essential Points
- Hands-On Exercise: Working with Kafka

Instructor-Led Demonstration of NiFi (Optional)

- **The instructor will demonstrate how to create a fairly complex dataflow**
- **The processors added in this exercise will implement the following scenario:**
  - Collect the output of an application log file
  - Split the contents into multiple files
  - Compress the files
  - Save the files in a destination directory
- **For further information consider the `https://www.cloudera.com/about/training/courses/dataflow-flow-managment-with-nifi.html` course**

Chapter Topics

Data Flow

- Overview of Cloudera Flow Management and NiFi
- NiFi Architecture
- Cloudera Edge Flow Management and MiNiFi
- Instructor-Led Demonstration: NiFi Usage
- **Apache Kafka Overview**
- Apache Kafka Cluster Architecture
- Apache Kafka Command Line Tools
- Essential Points
- Hands-On Exercise: Working with Kafka

What Is Apache Kafka?

- **Apache Kafka is a distributed “commit log” service**
  - Conceptually similar to a publish-subscribe messaging system
  - Offers scalability, performance, reliability, and flexibility
  - Widely used for data ingest
- **Originally created at LinkedIn, now an open source Apache project**
  - Donated to the Apache Software Foundation in 2011
  - Graduated from the Apache Incubator in 2012
  - Supported by Cloudera for production use in 2015

Advantages of Kafka

- **Scalable**
  - Kafka is a distributed system that supports multiple nodes
- **Fault-tolerant**
  - Data is persisted to disk and can be replicated throughout the cluster
- **High throughput**
  - Each broker can process hundreds of thousands of messages per second*
- **Low latency**
  - Data is delivered in a fraction of a second
- **Flexible**
  - Decouples the production of data from its consumption

* Using modest hardware, with messages of a typical size

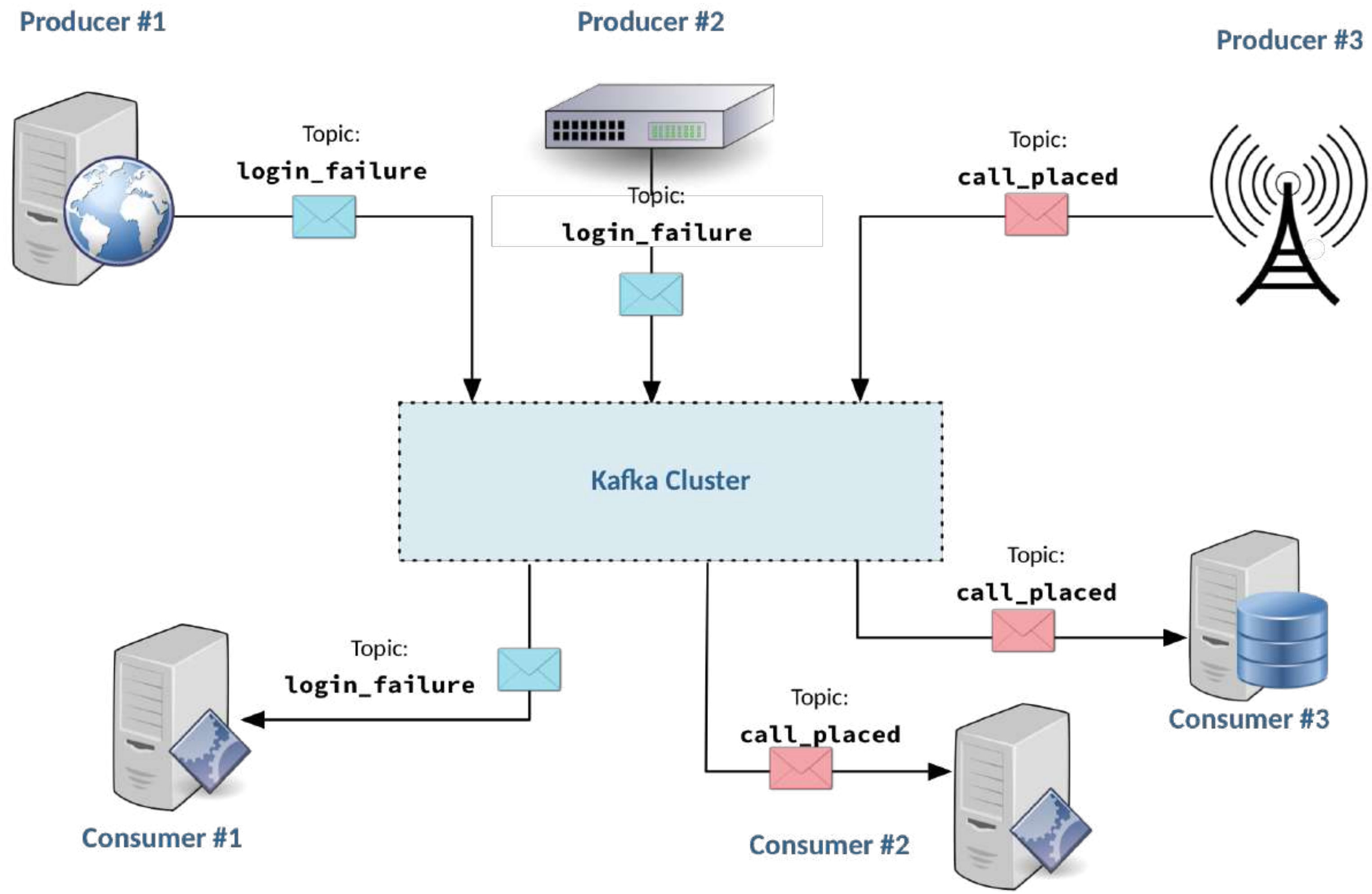
Kafka Use Cases

- **Kafka is deployed in a variety of use cases, such as**
  - Log aggregation
  - Messaging
  - Web site activity tracking
  - Stream processing
  - Event sourcing

Key Terminology

- **Message**
  - A single data record passed by Kafka
- **Topic**
  - A named log or feed of messages within Kafka
- **Producer**
  - A program that writes messages to Kafka
- **Consumer**
  - A program that reads messages from Kafka

Example: High-Level Architecture



Messages

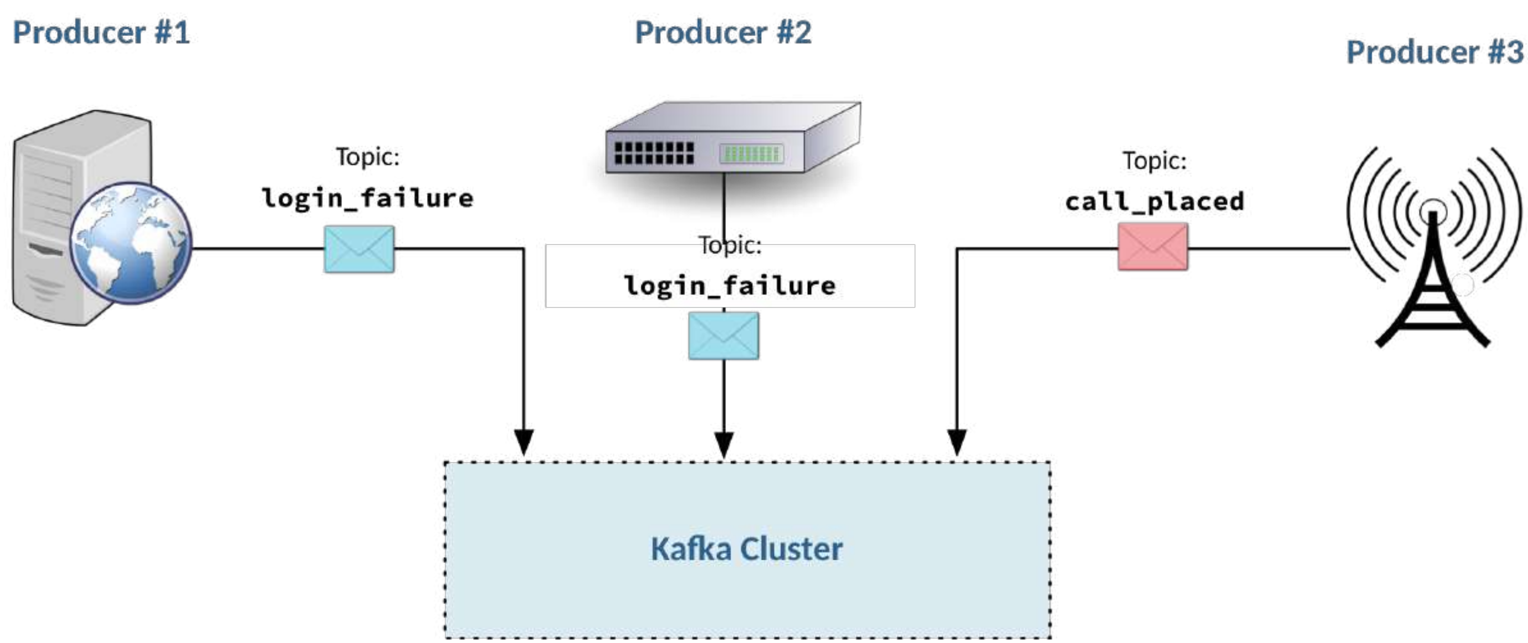
- **Messages in Kafka are variable-sized byte arrays**
  - Represent arbitrary user-defined content
- **Performs best with small message**
  - Optimal performance up to 10KB per message
  - No technical limit on message size, but practical limit of 1MB per message
- **Kafka retains all messages in a *log directory* for a configurable time period and/or total size**
  - Administrators can specify retention on global or per-topic basis
  - Kafka will retain messages regardless of whether they were read
  - Kafka discards messages automatically after the retention period or total size is exceeded (whichever limit is reached first)
  - Default retention is one week
  - Retention can reasonably be one year or longer

Topics

- **There is no explicit limit on the number of topics**
  - However, Kafka works better with a few large topics than many small ones
- **Creating topics**
  - Topics can be created using the Kafka command-line interface, API or SMM
  - By default, topics are also created automatically when an application publishes a message to a non-existent topic
    - Cloudera recommends topic auto-creation be disabled to prevent accidental creation of large numbers of topics

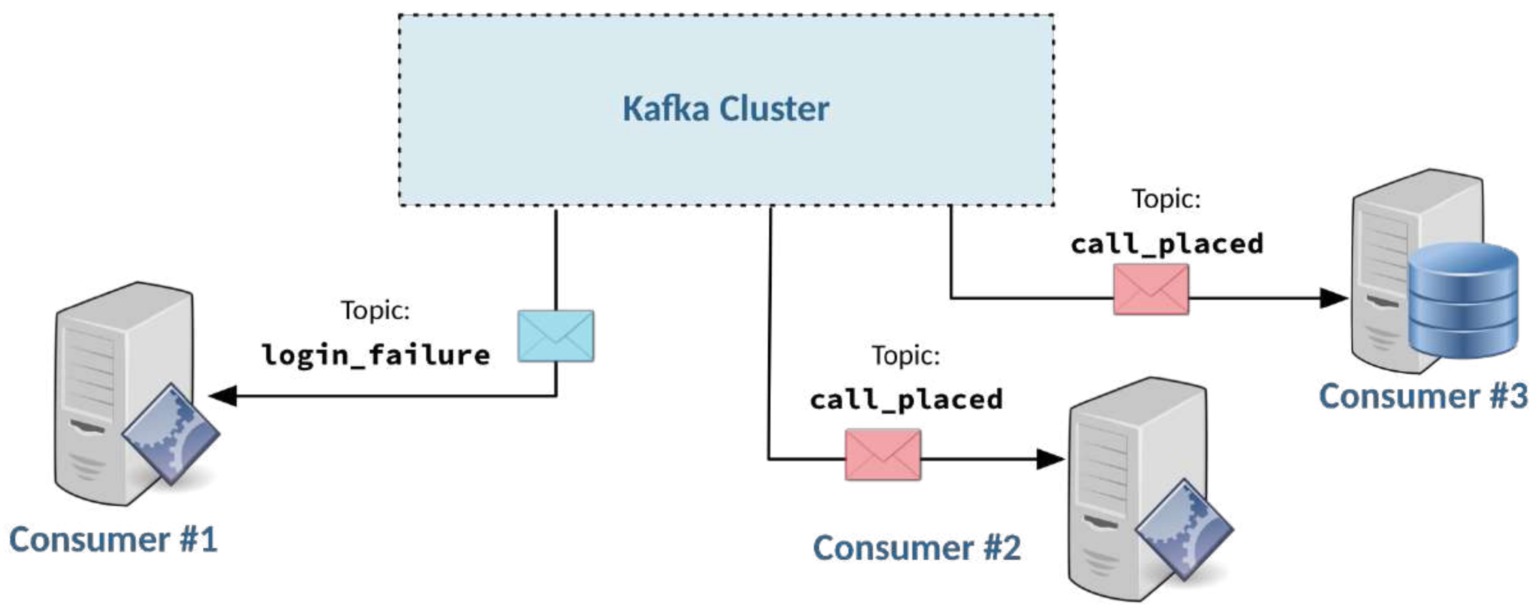
Producers

- **Producers publish messages to Kafka topics**
  - They communicate with Kafka, not a consumer
  - Kafka persists messages to disk on receipt



Consumers

- **A consumer reads messages that were published to Kafka topics**
  - They communicate with Kafka, not any producer
- **Consumer actions do not affect other consumers**
  - For example, having one consumer display the messages in a topic as they are published does not change what is consumed by other consumers
- **They can come and go without impact on the cluster or other consumers**

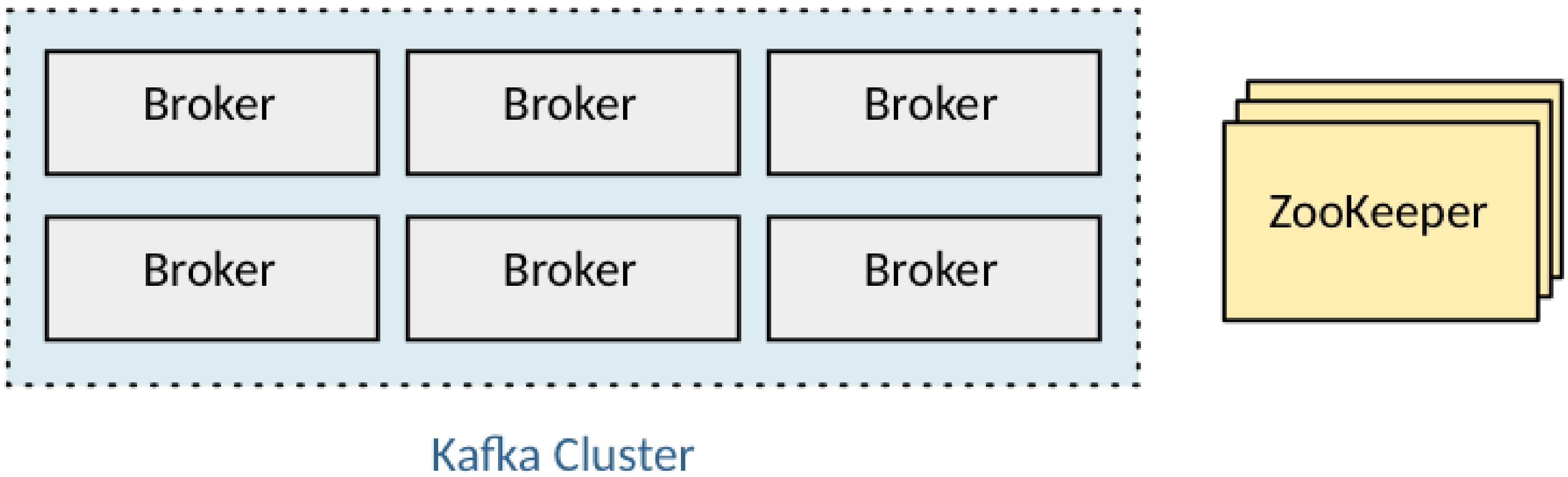


Chapter Topics

Data Flow

Kafka Clusters

- **A Kafka cluster consists of one or more *brokers*—hosts running the Kafka broker daemon**
  - Kafka clusters are separate from Hadoop clusters
  - Cloudera recommends not colocating Hadoop services with Kafka brokers
- **Kafka depends on the Apache ZooKeeper service for coordination**

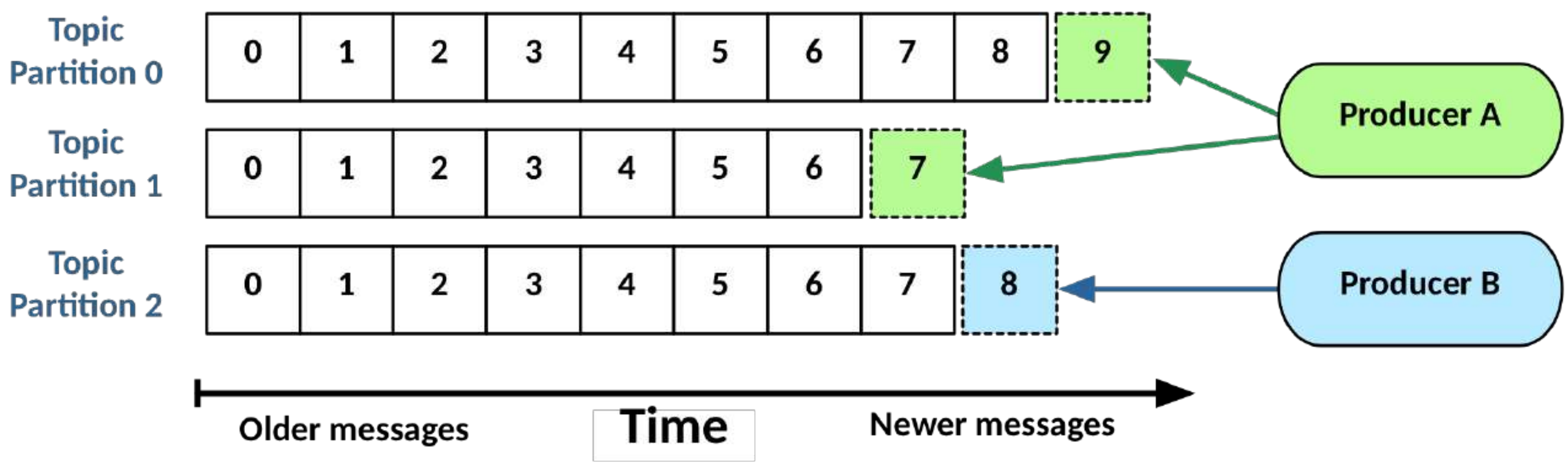


Kafka Brokers

- **Brokers are the daemons that make up a Kafka cluster**
- **A broker stores a topic partition on disk**
  - OS-level disk caching improves performance
- **Kafka messages are separated into partitions for performance**
  - Partitions are hosted by different brokers
  - A single broker can reasonably host 1000 topic partitions
- **One broker is elected controller of the cluster**
  - For assignment of topic partitions to brokers, and so on
- **Each broker daemon runs in its own JVM**

Topic Partitioning

- **Kafka divides each topic into some number of partitions***
  - Topic partitioning improves scalability and throughput
- **A topic partition is an ordered and immutable sequence of messages**
  - New messages are appended to the partition as they are received
  - Each message is assigned a unique sequential ID known as an *offset*



* Note that the Kafka topic partitioning is independent of partitioning in HDFS, Kudu, or Spark

Chapter Topics

Data Flow

- Overview of Cloudera Flow Management and NiFi
- NiFi Architecture
- Cloudera Edge Flow Management and MiNiFi
- Instructor-Led Demonstration: NiFi Usage
- Apache Kafka Overview
- Apache Kafka Cluster Architecture
- **Apache Kafka Command Line Tools**
- Essential Points
- Hands-On Exercise: Working with Kafka

Creating Topics from the Command Line

- **Kafka includes a convenient set of command line tools**
  - These are helpful for testing, exploring, and experimentation
- **The `kafka-topics` command offers a simple way to create Kafka topics**
  - Provide the topic name of your choice, such as `device_status`
  - You must also specify the *bootstrap server* (one or more brokers) for your cluster

```
$ kafka-topics --create \
--bootstrap-server broker1:9092,broker2:9092,broker3:9092 \
--replication-factor 3 \
--partitions 5 \
--topic topic-name
```

Displaying Topics from the Command Line

- Use the `--list` option to list all topics

```
$ kafka-topics --list \
--bootstrap-server broker1:9092,broker2:9092,broker3:9092
```

- Use the `--delete` option to delete a topic
  - `delete.topic.enable` must be set

```
$ kafka-topics --delete \
--bootstrap-server broker1:9092,broker2:9092,broker3:9092 \
--topic topic-name
```

Running a Producer from the Command Line

- **You can run a producer using the `kafka-console-producer` tool**
- **Required arguments**
  - `topic`—the topic name
  - `broker-list`—one or more brokers
- **Reads input from stdin**

```
$ kafka-console-producer \
--broker-list broker1:9092,broker2:9092,broker3:9092 \
--topic topic-name
>input message 1
>input message 2
```

Running a Consumer from the Command Line

- **You can run a consumer with the `kafka-console-consumer` tool**
  - Primarily for testing purposes
- **Arguments**
  - `topic`
  - `bootstrap-server`
  - `from-beginning`
    - Optional—if not specified, reads only new messages

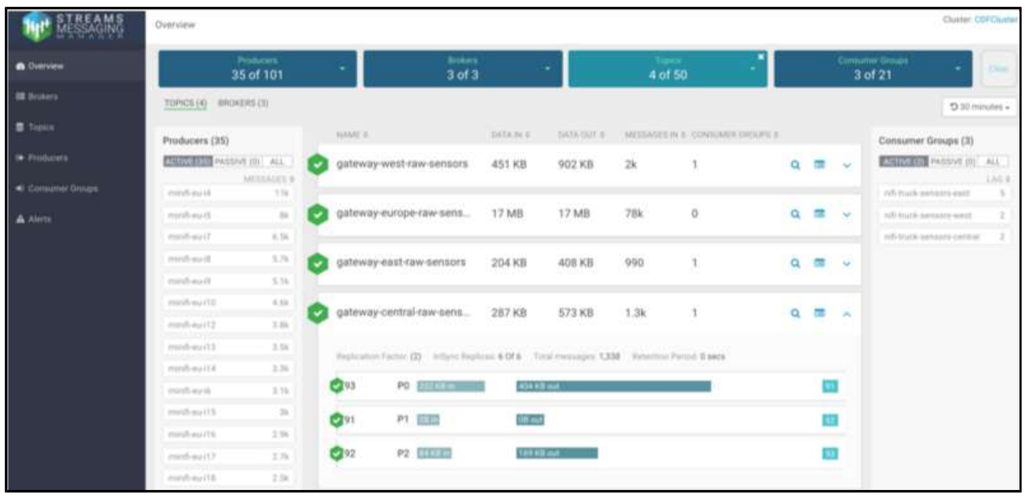
```
$ kafka-console-consumer \
--bootstrap-server broker1:9092,broker2:9092,broker3:9092 \
--topic topic-name \
--from-beginning
```

Streams Messaging Manager Overview

- **Streams Messaging Manager (SMM) is an operations monitoring and management tool**
  - Provides end-to-end visibility into Apache Kafka
  - Gain clear insights about your Kafka clusters
  - Troubleshoot your Kafka environment to identify bottlenecks, throughputs, consumer patterns, traffic flow etc.
  - Analyze the stream dynamics between producers and consumers
  - Optimize your Kafka environment based on the key performance insights

Streams Messaging Manager Benefits

- **Provides a dashboard to visually monitor all of your Kafka clusters**
- **Cures "Kafka Blindness"**
  - Struggle to understand what is happening inside Kafka
  - Difficulty knowing the state of topics and brokers
  - Inability to see who is producing and consuming the data
  - Difficulty monitoring and troubleshooting the flow of data

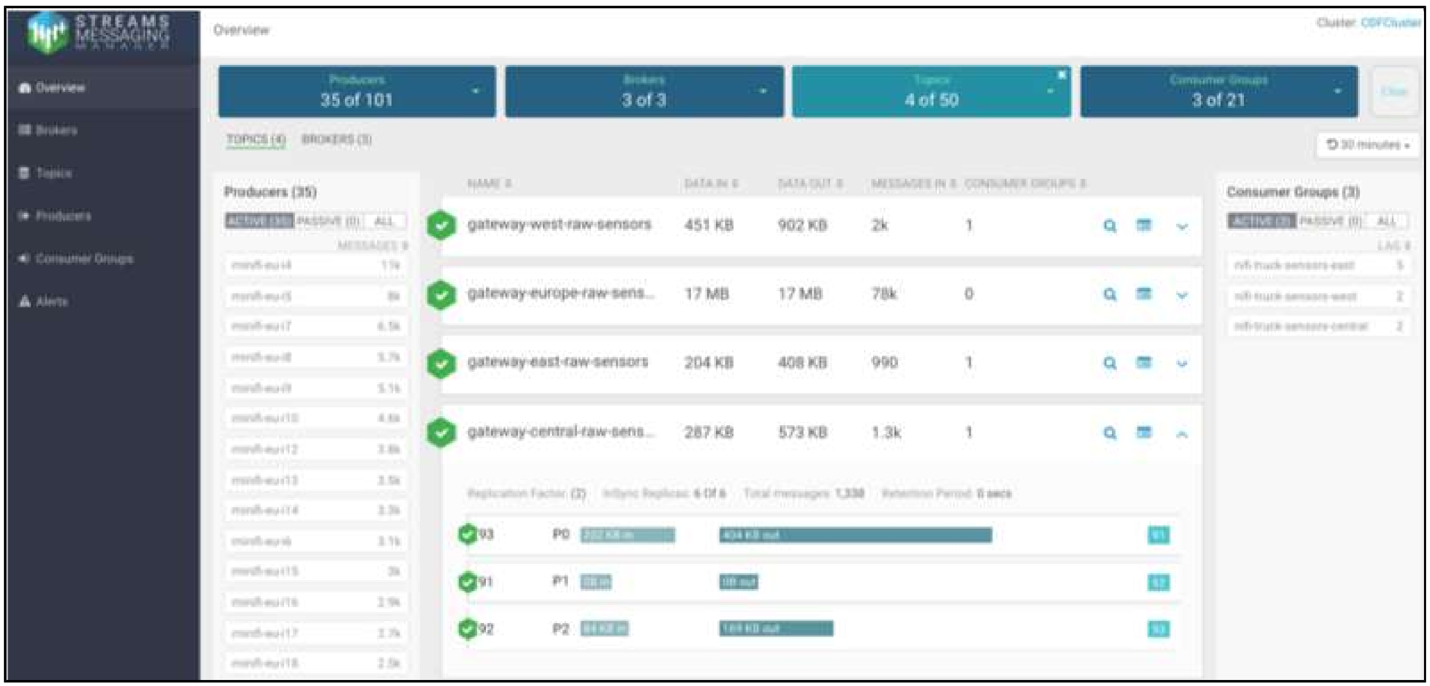


Designed for the Enterprise

- Single monitoring dashboard
- Support for multiple Kafka clusters
- Intelligent filtering
- Topic management
- Alert policies and notifications
- Authentication and authorization

SMM Components

- **SMM consists of two components**
  - SMM UI Server
  - SMM REST Admin Server
- **SMM UI powered by first class REST Services via the SMM REST Admin Server**



Streams Replication Manager (SRM)

- **Kafka supports internal replication to support data availability within a cluster**
- **However with Kafka based applications becoming critical, enterprises require that the data availability and durability guarantees span entire cluster and site failures**
- **Replication of data across clusters and sites is key for the following use cases:**
  - Disaster recovery
  - Aggregation for analytics
  - Data deployment after analytics
  - Isolation for access control
  - Geo Proximity
  - Cloud Migration
  - Legal and Compliance

Streams Replication Manager (SRM)

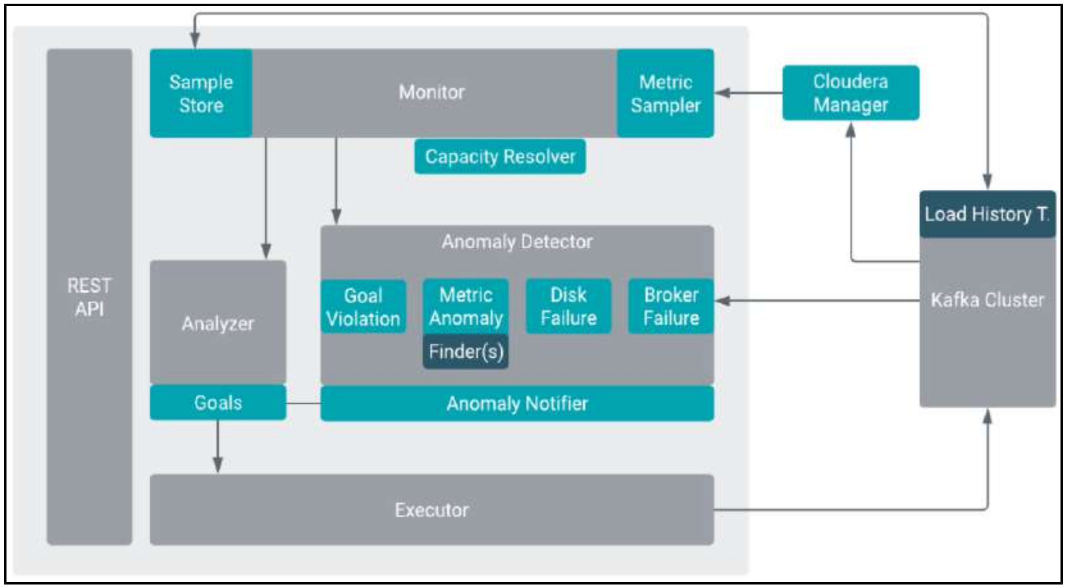
- **A replication solution that enables fault tolerant, scalable and robust cross-cluster Kafka topic replication**
  - Provides the ability to dynamically change configurations
  - Keeps the topic properties in sync across clusters
  - Delivers custom extensions that facilitate installation, management and monitoring
  - Kafka supports internal replication to ensure data availability within a cluster
  - SRM can span the entire cluster as well as site failures in regard to data availabiity and durability

Streams Replication Manager (SRM)

- **Consists of two main components:**
  - Stream Replication Engine
    - A next generation multi-cluster and cross-datacenter replication engine for Kafka clusters
  - Stream Replication Management Services
    - Services powered by open source Cloudera extensions which utilize the capabilities of the Stream Replication Engine

Cruise Control

- **Cruise Control is a Kafka load balancing component that can be used in large Kafka installations**
- **Cruise Control can automatically balance the partitions**
- **Based on specific conditions when adding or removing Kafka brokers**
- **The architecture of Cruise Control consists of the Load Monitor, Analyzer, Anomaly Detector and Executor**



Chapter Topics

Data Flow

- Overview of Cloudera Flow Management and NiFi
- NiFi Architecture
- Cloudera Edge Flow Management and MiNiFi
- Instructor-Led Demonstration: NiFi Usage
- Apache Kafka Overview
- Apache Kafka Cluster Architecture
- Apache Kafka Command Line Tools
- **Essential Points**
- Hands-On Exercise: Working with Kafka

Essential Points

- **Cloudera Flow Management uses NiFi to manage data flow**
- **In Kafka, Producers publish messages to categories called *topics***
- **Messages in a topic are read by consumers**
- **Topics are divided into partitions for performance and scalability**
  - Partitions are replicated for fault tolerance
- **Kafka *brokers* receive messages from producers, store them, and pass them to consumers**

Chapter Topics

Data Flow

- Overview of Cloudera Flow Management and NiFi
- NiFi Architecture
- Cloudera Edge Flow Management and MiNiFi
- Instructor-Led Demonstration: NiFi Usage
- Apache Kafka Overview
- Apache Kafka Cluster Architecture
- Apache Kafka Command Line Tools
- Essential Points
- **Hands-On Exercise: Working with Kafka**

Hands-On Exercise: Working with Kafka

- **In this exercise, you will Install Kafka. Then you will use Kafka's command line tool to create a Kafka topic and use the command line producer and consumer clients to publish and read messages.**
- **Please refer to the Hands-On Exercise Manual for instructions**

Data Access and Discovery

Chapter 8

Course Chapters

Data Access and Discovery

After completing this chapter, you will be able to

- **Explain the roles of the metastore, Apache Hive, and Apache Impala in a cluster deployment**
- **Describe the function of each Hive and Impala service role in Cloudera Manager**
- **Distinguish the differences between internal and external tables**
- **Explain the advantages that Impala provides relative to Hive**
- **Describe the usage of Atlas**
- **Describe Search and its features**
- **Describe the usage of Cloudera Data Science Workbench**

Chapter Topics

Data Access and Discovery

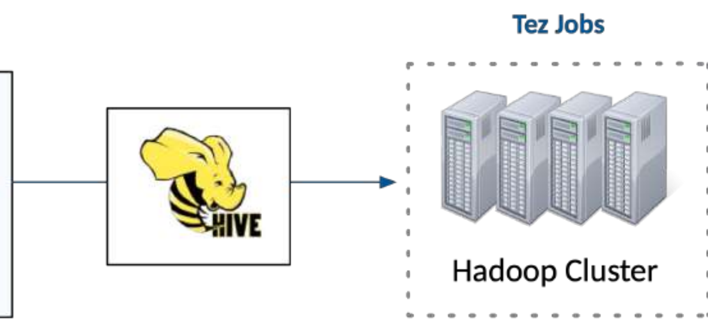
- **Apache Hive**
- Apache Impala
- Apache Impala Tuning
- Hands-On Exercise: Install Impala and Hue
- Search Overview
- Hue Overview
- Managing and Configuring Hue
- Hue Authentication and Authorization
- CDSW Overview
- Essential Points
- Hands-On Exercise: Using Hue, Hive and Impala

What is Apache Hive?

- **Uses a SQL-like language called HiveQL**
  - Query data in HDFS, HBase, S3, or ADLS
- **Fulfills queries by running Tez jobs by default**
  - Automatically runs the jobs, returns the results

HiveQL Query

```
  SELECT id, fname, lname, city, unpaid
    FROM accounts
    JOIN billing
      ON accounts.acct_num = billing.acct_num
   WHERE state = 'CA'
     AND unpaid > 100
ORDER BY unpaid DESC;
```



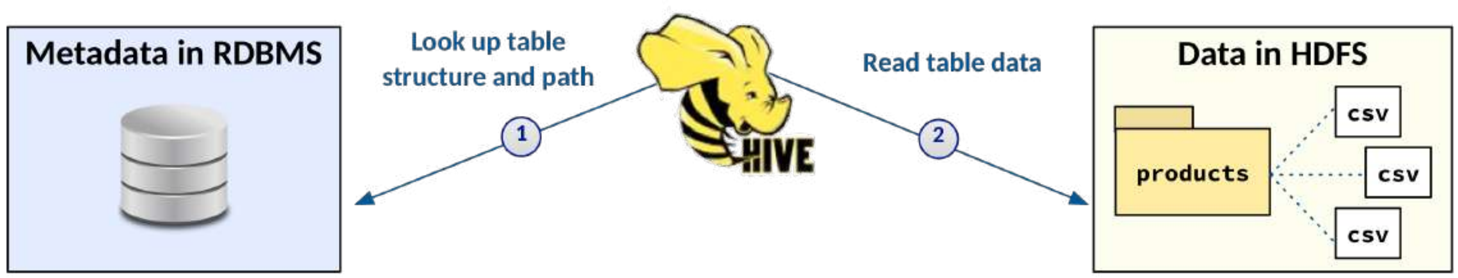
- **Motivation**
  - SQL is the standard language for querying data in relational databases
  - Data analysts may not know programming languages like Java, Python

How Hive Works

- **Hive stores data in tables like RDBMS systems**
- **A table is a data directory in HDFS, S3, or ADLS with associated metadata**
- **Tables can created for pre-existing data**

```
CREATE TABLE products (
  id INT,
  name STRING,
  price INT
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';
```

- ***Metadata* (table structure and path to data) is stored in an RDBMS**



Hive Tables

- **A Hive table points to data in a directory**
  - Hive interprets all files in the directory as the contents of the table
  - The metastore describes how and where the data is stored
- **Use the `CREATE TABLE` command to define an *internal* table**
  - Hive will manage the files, metadata and statistics
  - CAUTION: If the table is dropped, schema *and data* are both deleted
- **Use the `CREATE EXTERNAL TABLE` command to define an *external* table**
  - Hive does not manage the data
  - If the table is dropped, only the table schema is deleted
- **Use the `DESCRIBE FORMATTED` *`table_name`* command to show table details**

Hive Table Data Location

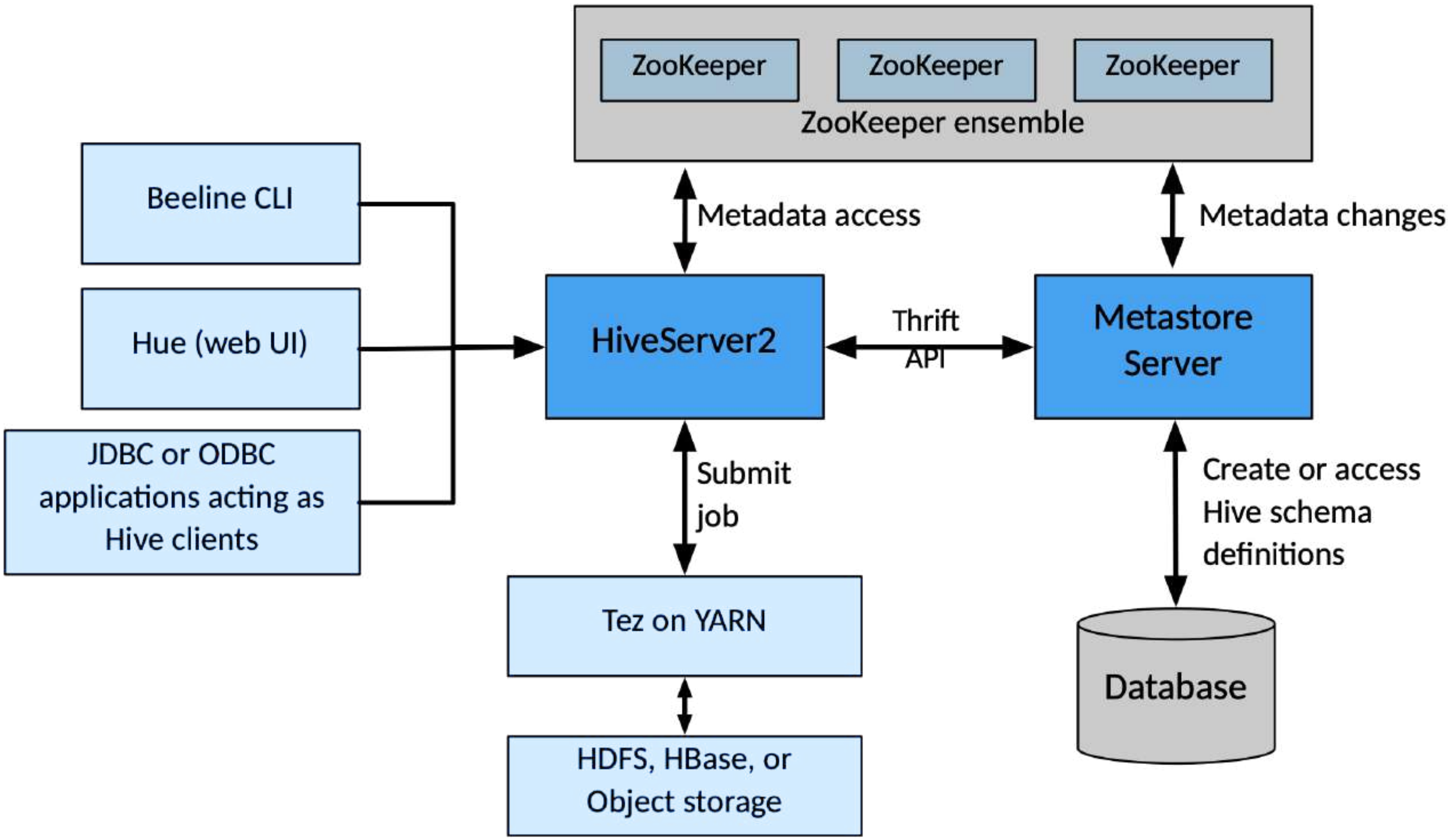
- **The `hive.metastore.warehouse.dir` property**
  - Specifies the default table data location in HDFS
  - Defaults to `/warehouse/tablespace/external/hive/`
  - Can be overridden by the `LOCATION` keyword when creating the table
- **Example of setting a non-default table data storage location**

```
CREATE EXTERNAL TABLE sample_table (
  name string, value int)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
LOCATION '/mydata/sample_table';
```

- **Use the `LOAD DATA` command to move data to table location**

```
LOAD DATA INPATH '/source-hdfs-directory/*'
INTO TABLE existing_table;
```

Hive Basic Architecture



Hive Service Roles Installed by Cloudera Manager

- **Hive Metastore Server**
  - Manages the metastore database
  - Hive Metastore does the following
    - Stores metadata for tables and partitions in a relational database
    - Provides clients (including Hive) access by way of the metastore API
- **HiveServer2**
  - Supports the Thrift API used by clients
  - Provides the Beeline Hive CLI
- **Hive Gateway**
  - Provides client configurations, including the libraries to access Beeline

Hive Metastore Server and Deployment Mode Options

- **Hive Metastore Server (remote mode)**
  - Recommended deployment mode
  - Metastore data is stored in a standalone RDBMS such as MySQL
  - Metastore Server uses JDBC to connect to RDBMS
  - Advantage over local mode
    - No need to share metastore JDBC login details with each Hive user
- **Local mode**
  - Functionality of Metastore Server embedded in the HiveServer process
  - Database runs separately, accessed using JDBC
- **Embedded mode**
  - Supports only one active user at a time—for experimental purposes only
  - Uses Apache Derby, a Java-based RDBMS

HiveServer2

- **HiveServer2**
  - A container for the Hive execution engine
  - Enables remote clients to run queries against Hive and retrieve the results
  - Accessible using JDBC, ODBC, or Thrift
  - Example clients: Beeline (the Hive CLI), Hue (Web UI)
- **Supports concurrent queries from multiple Hive clients**
  - Example of why concurrency support is needed
    - A Hive client issues a `DROP TABLE` command
    - At the same time another client on a different machine runs a `SELECT` query against the same table
  - Hive concurrency requires Apache ZooKeeper
    - To guard against data corruption while supporting concurrent clients

Using the Beeline CLI

- **Start Beeline connected to HiveServer2**

```
$ beeline -u jdbc:hive2://master-2:10000 -n training
```

- **Define tables or run HiveQL queries**

```
0: jdbc:hive2://master-2:10000> SELECT COUNT(*) FROM
 movierating;
+----------+
|   _c0    |
+----------+
| 1000205  |
+----------+
1 row selected (83.514 seconds)
0: jdbc:hive2://master-2:10000> !quit
Closing: org.apache.hive.jdbc.HiveConnection
```

Hive 3 Key Features (1)

- **Enhancements in Hive 3 can improve SQL query performance, security, and auditing capabilities**
  - ACID transaction processing (details in Ch. 9)
  - Shared Hive metastore
  - Low-latency analytical processing - LLAP (not in CDP PvC Base)
  - Spark integration with Hive
  - Security improvements (details in Ch 16)
  - Workload management at the query level
  - Materialized views
  - Query results cache

Hive 3 Key Features (2)

- **Be aware of the unavailable or unsupported interfaces for Hive 3**
  - Hive CLI (replaced by Beeline)
  - WebHCat
  - Hcat CLI
  - SQL Standard Authorization
  - Hive Indexes
  - MapReduce execution engine (replaced by Tez)
  - Spark execution engine (replaced by Tez)

Chapter Topics

Data Access and Discovery

- Apache Hive
- **Apache Impala**
- Apache Impala Tuning
- Hands-On Exercise: Install Impala and Hue
- Search Overview
- Hue Overview
- Managing and Configuring Hue
- Hue Authentication and Authorization
- CDSW Overview
- Essential Points
- Hands-On Exercise: Using Hue, Hive and Impala

What Is Apache Impala?

- **Allows users to query data using SQL**
  - Data can be in HDFS, Kudu, Amazon S3, HBase or Microsoft ADLS
- **Impala does not run queries as MapReduce, Spark or Tez jobs (unlike Hive)**
  - Queries run on an additional set of daemons on the cluster
  - Impala supports many simultaneous users more efficiently than Hive
- **Impala is best for interactive, ad hoc queries**
  - Hive is better for large, long-running batch processes
- **Impala uses the same shared metastore as Hive**
  - Tables created in Hive are visible in Impala (and the other way around)

The Impala Shell

- The Impala Shell is an interactive tool similar to the Beeline shell for Hive
- Start the `impala-shell` connected to an Impala daemon

```
$ impala-shell -i worker-1.example.com:21000
```

- Define tables or run queries

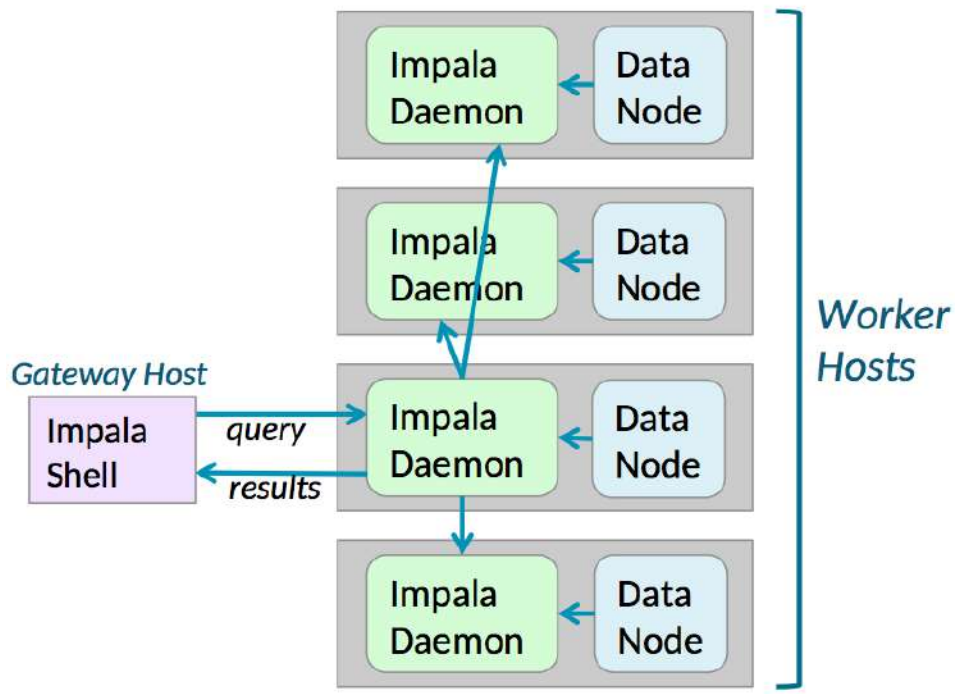
```
[worker-1.example.com:21000] > SELECT acct_num,
  first_name, last_name FROM accounts
  WHERE zipcode='90210';
+----------+-------------+-------------+
| acct_num | first_name  | last_name   |
+----------+-------------+-------------+
| 6029     | Brian       | Ferrara     |
| 9493     | Mickey      | Kunkle      |
+----------+-------------+-------------+
Fetched 2 row(s) in 0.01s
[worker-1.example.com:21000] > quit;
```

Impala and the Metastore

- **Impala uses the same metastore as Hive**
- **Can access tables defined or loaded by Hive**
- **Tables must use data types, file formats, and compression codecs supported by Impala**
- **Impala supports read-only access for some data formats**
  - Avro Data Format, RCFile, or SequenceFile
  - Load data using Hive, then query with Impala

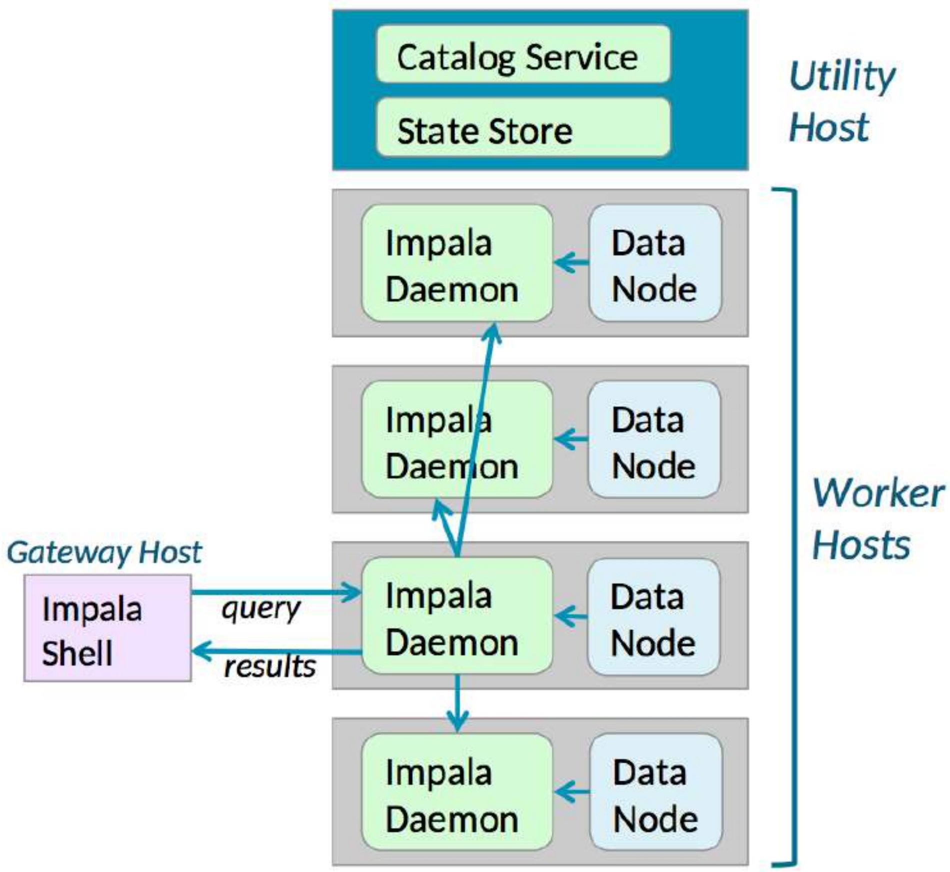
Impala Daemons

- **Read and write to data files and stream results to the client**
- **Accept queries from clients**
  - Such as the Impala shell, Hue, JDBC, and ODBC clients
- **The daemon that receives the query is the query's *coordinator***
- **The coordinator distributes work to other Impala daemons**
  - Daemons transmit intermediate results back to coordinator
  - Coordinator returns results to client



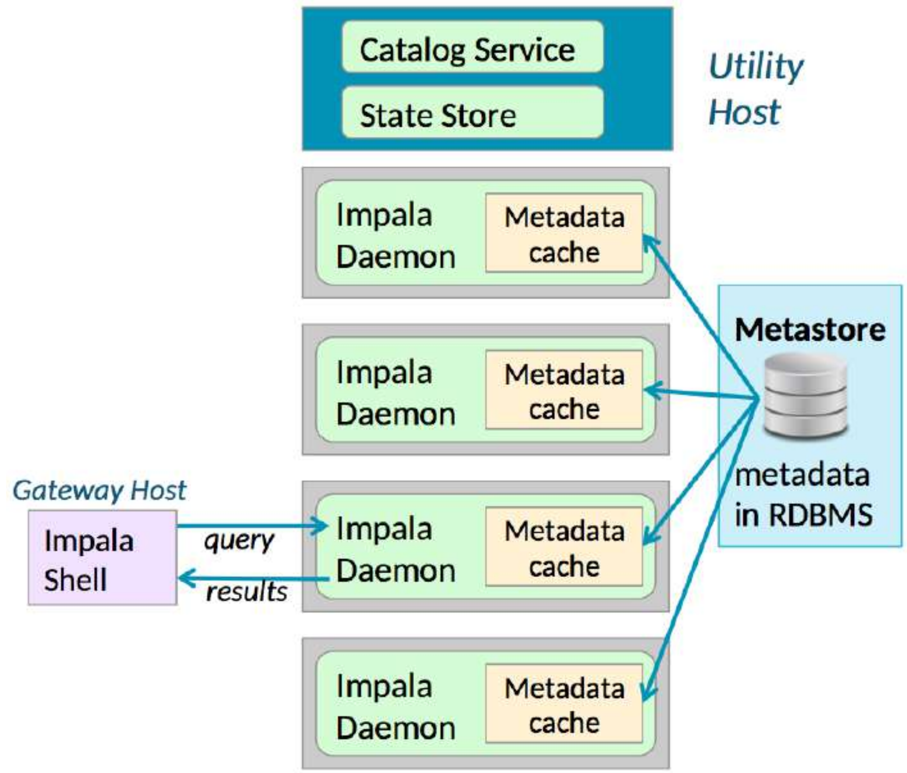
Impala State Store and Catalog Service

- **State Store**
  - Provides lookup service and status checks for Impala daemons
  - One per cluster
- **Catalog Server**
  - Relays metadata changes to all Impala daemons
  - One per cluster



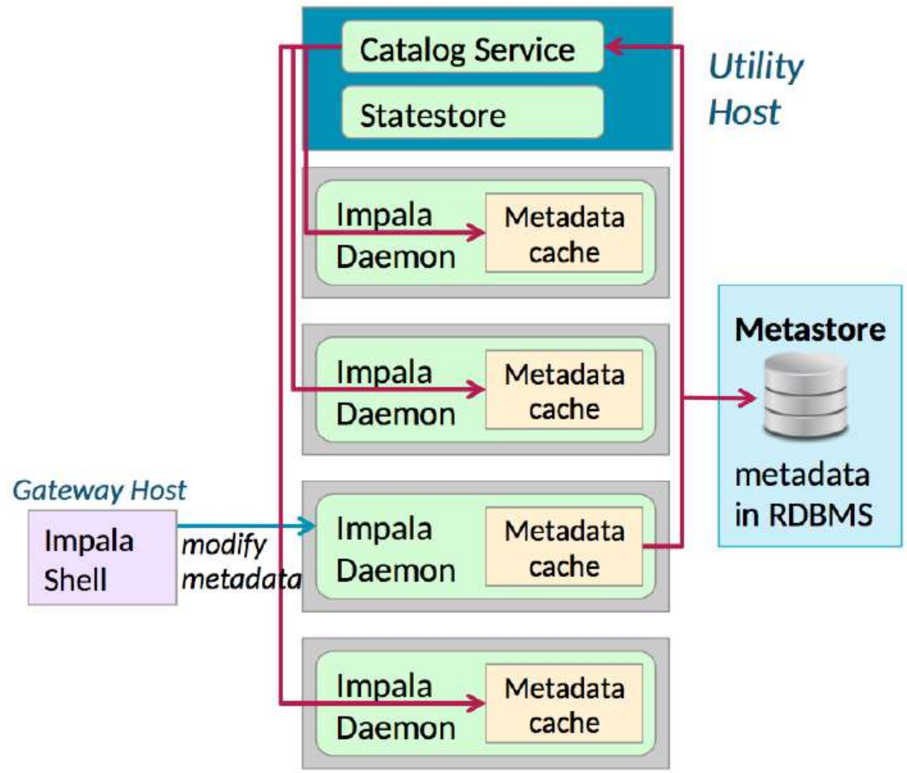
Metadata Caching (1)

- **Impala daemons cache metadata**
  - Table schema definitions
  - Locations of HDFS blocks containing table data
- **Metadata is cached from the metastore and HDFS at startup**
- **Reduces query latency**



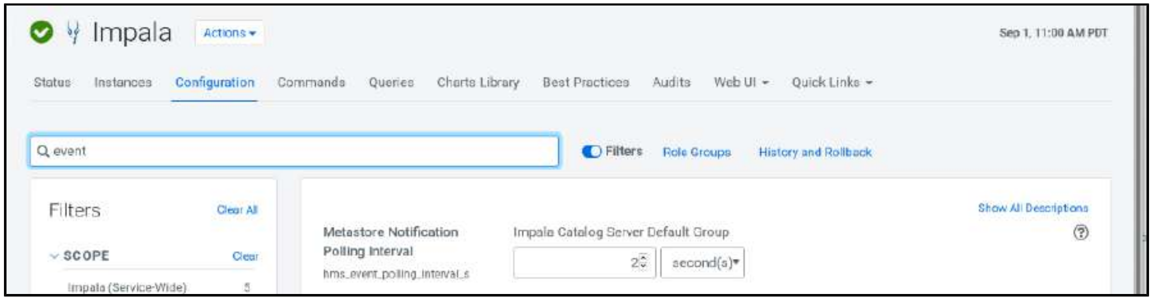
Metadata Caching (2)

- **When one Impala daemon changes the metastore or the location of HDFS blocks, it notifies the catalog service**
- **The catalog service notifies all Impala daemons to update their cache**



Automatic Invalidation or Refresh of Metadata

- **In previous versions of Impala, users needed to manually issue an INVALIDATE or REFRESH commands**
- **Impala Catalog Server polls and processes the following changes:**
  - Invalidates the tables when it receives the `ALTER TABLE` event
  - Refreshes the partition when it receives the `ALTER`, `ADD`, or `DROP` partitions
  - Adds the tables or databases when it receives the `CREATE TABLE` or `CREATE DATABASE` events
  - Removes the tables from catalogd when it receives the `DROP TABLE` or `DROP DATABASE` events
  - Refreshes the table and partitions when it receives the `INSERT` events



Installing Impala

- **Impala is part of the Runtime**
- **Install using Cloudera Manager**
  - HDFS and Hive are prerequisites
  - Cloudera Manager installs the Impala service
    - Includes Impala Catalog Server, Impala StateStore, and Impala Daemon roles
    - Creates `/user/impala` directory in HDFS
    - Deploys the Impala client to all hosts that have the Impala daemon
- **Cloudera recommends running Hive Metastore, Impala State Store, and Impala Catalog Server on the same host**
  - Typically on a utility host
  - May be on a gateway or master host for a small cluster, but not the NameNode host

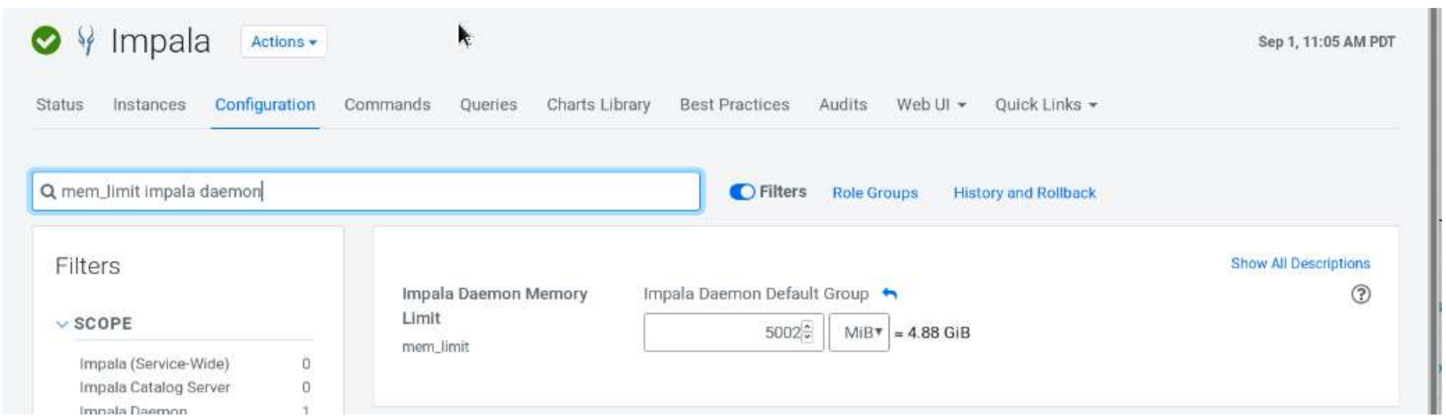
Recommendations to Improve Impala Performance

- **Use binary file formats instead of text-based formats**
  - Parquet is the most efficient
  - Avro is also common
- **Snappy compression provides a good balance between size and speed**
- **Store data as numeric types instead of strings when possible**
- **Partition the largest and most frequently queried tables**
  - But avoid creating many very small partitions (over-partitioning)
- **Always compute statistics after loading data using `COMPUTE STATS`**
  - Use `COMPUTE INCREMENTAL STATS` when data is appended to existing table
- **Avoid ingest process that create many small files**
  - Performance is best with larger files
- **Verify execution plans with `EXPLAIN` and `SUMMARY`**

Impala Daemon Memory Configuration

- **Configure the amount of system memory available to the Impala daemon**
  - Controlled by `Impala Daemon Memory Limit` parameter (`mem-limit`)
  - Default value: (empty)—allows Impala to pick its own limit
  - Set based on system activity
  - Setting is ignored when Dynamic Resource Pools is enabled

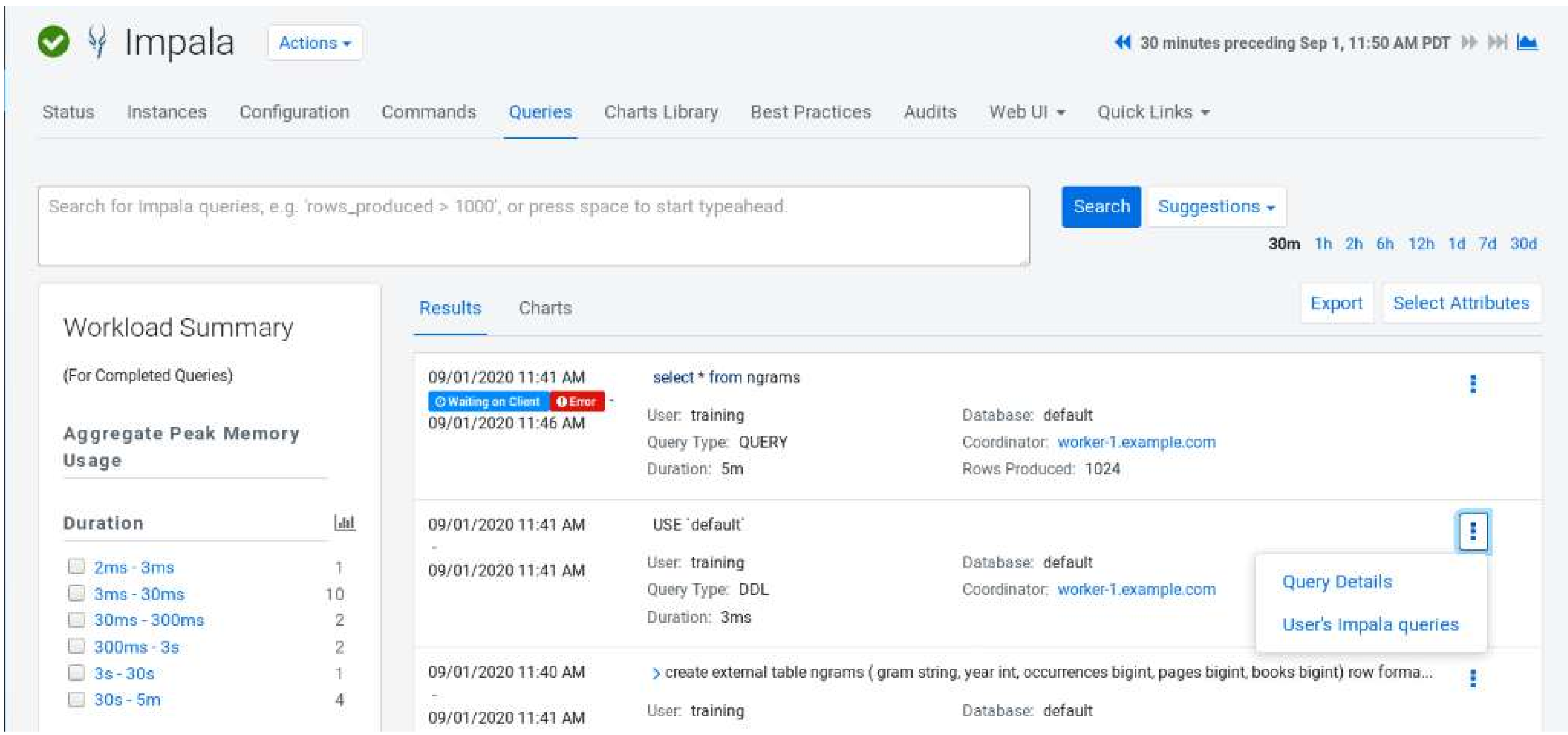
Impala Configuration Tab



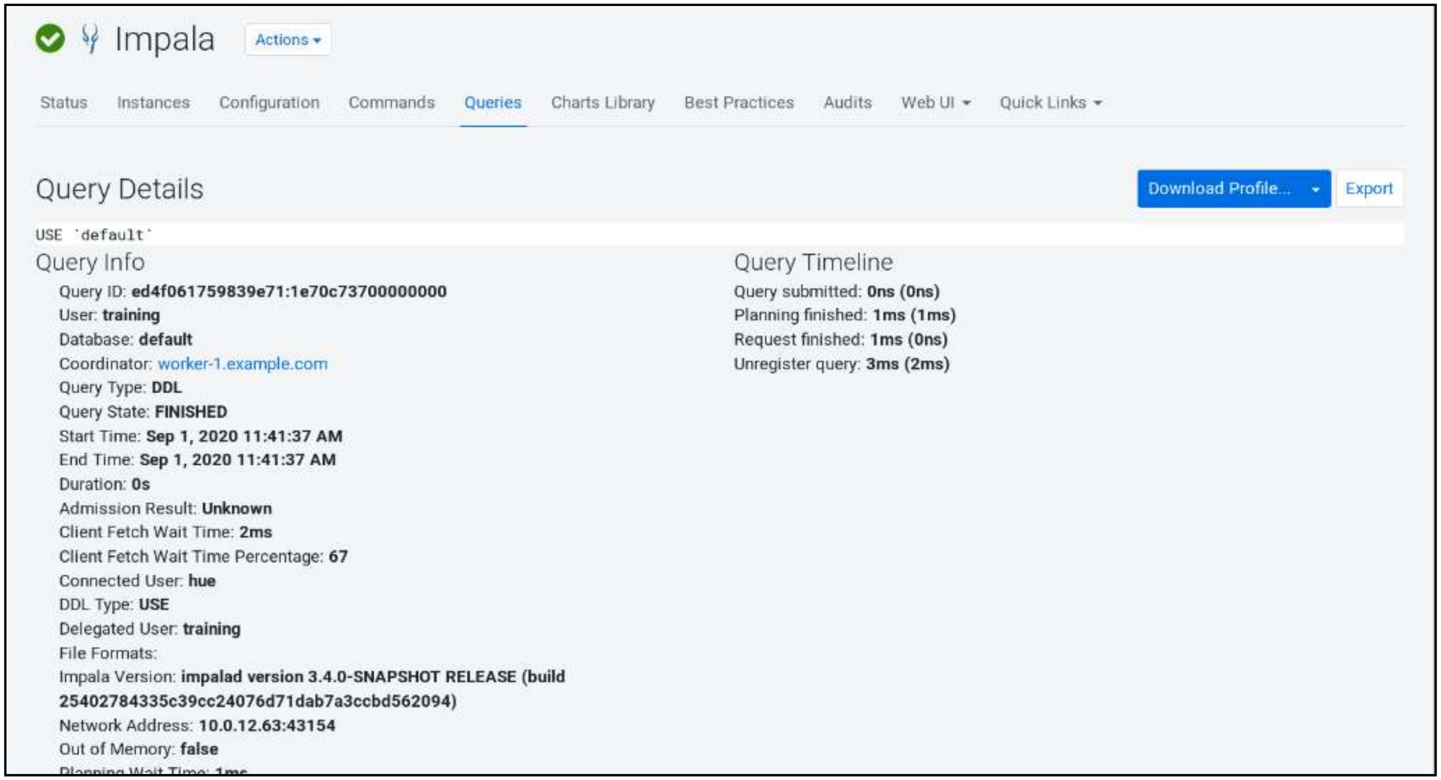
Viewing Impala Queries in Cloudera Manager

- **View queries in Cloudera Manager on the Impala** Queries **tab**
  - Options to filter queries and view query details

Clusters > Impala Queries



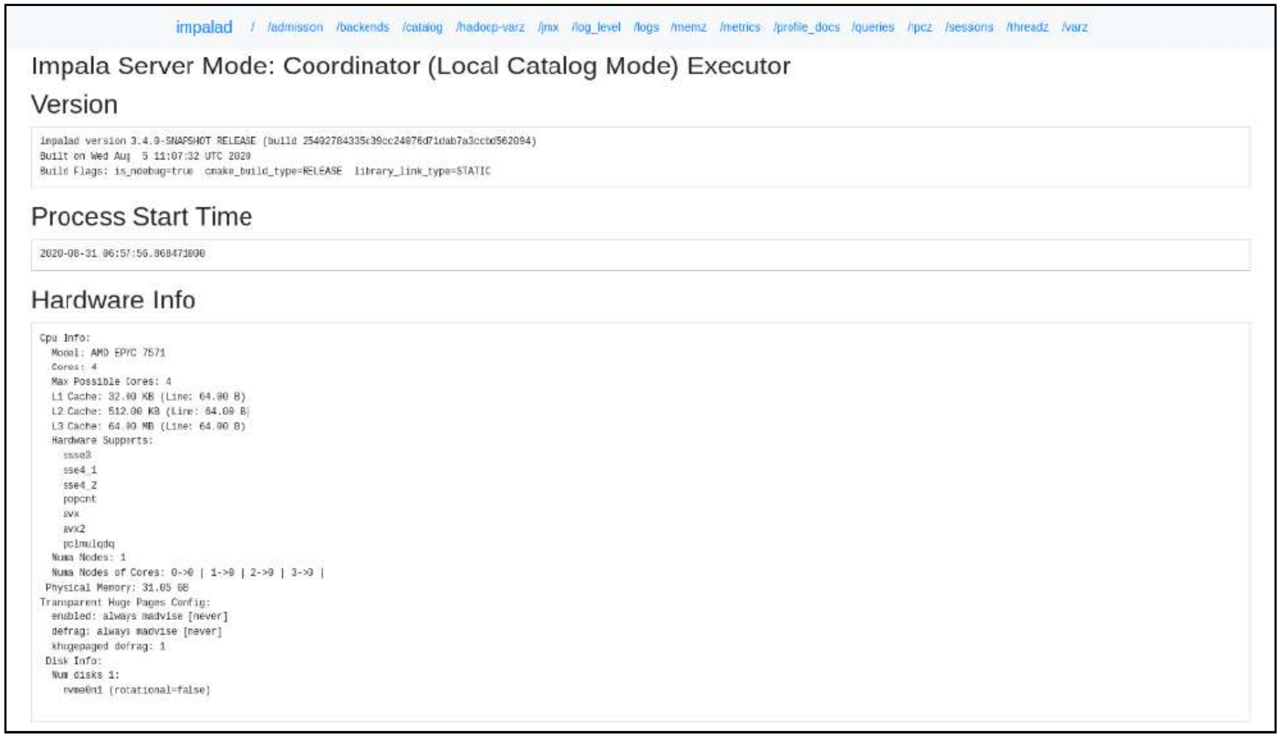
Impala Query Details



Impala Web UIs (1)

- **Each of the Impala daemons (impalad, statestored, and catalogd) includes a built-in web server**
  - Display diagnostic and status information

Impala Daemon Web UI: `http://`*`impalad-server`*`:25000`



Impala Web UIs (2)

Impala Catalog Server Web UI: `http://`*`catalog-server-host`*`:25020`

Impala StateStore Web UI: `http://`*`statestore-host`*`:25010`

Chapter Topics

Data Access and Discovery

- Apache Hive
- Apache Impala
- **Apache Impala Tuning**
- Hands-On Exercise: Install Impala and Hue
- Search Overview
- Hue Overview
- Managing and Configuring Hue
- Hue Authentication and Authorization
- CDSW Overview
- Essential Points
- Hands-On Exercise: Using Hue, Hive and Impala

What to Avoid with Impala

- **Joining lots of tables (more than 5)**
- **Dealing with complex views**
- **3NF schemas - the lesser number of joins and wider tables used corresponds to faster query execution times**
- **Batch Processing**
- **Anything that can create lots of small files**

Good Rules for Impala Query Creation

- Impala tables should have less than 2000 columns
- Skew is often an issue with joins – one customer might have 1000x more rows
- Codegen can sometimes add hours to a complex query and can be disabled if it shows to be a problem
- Partitions should optimally be around 2GB in size
- Should have less than 100K partitions, less than 20K is better

Impala Performance Best Practices

- **Choose the appropriate file format for the data:**
  - Typically, for large volumes of data (multiple gigabytes per table or partition), Parquet file format performs best
- **Avoid data ingestion processes that produce many small files**
- **Use smallest appropriate integer types for partition key columns**
- **Gather the statistics with COMPUTE STATS for performance-critical queries**
- **Hotspot analysis looking for Impala daemons spending greater amount of time processing data than its neighbors**

Chapter Topics

Data Access and Discovery

- Apache Hive
- Apache Impala
- Apache Impala Tuning
- **Hands-On Exercise: Install Impala and Hue**
- Search Overview
- Hue Overview
- Managing and Configuring Hue
- Hue Authentication and Authorization
- CDSW Overview
- Essential Points
- Hands-On Exercise: Using Hue, Hive and Impala

Hands-On Exercise: Install Impala and Hue

- **In this exercise, you will install Impala and HUE.**
- **Please refer to the Hands-On Exercise Manual for instructions**

Chapter Topics

Data Access and Discovery

- Apache Hive
- Apache Impala
- Apache Impala Tuning
- Hands-On Exercise: Install Impala and Hue
- **Search Overview**
- Hue Overview
- Managing and Configuring Hue
- Hue Authentication and Authorization
- CDSW Overview
- Essential Points
- Hands-On Exercise: Using Hue, Hive and Impala

Cloudera Search Overview

- Cloudera Search provides easy, natural language access to data stored in HDFS, HBase, or the cloud
- Cloudera Search is Apache Solr fully integrated in the Cloudera platform
- End users and other web services can use full-text queries
- Explore text, semi-structured, and structured data to filter and aggregate it without requiring SQL or programming skills

Cloudera Search Overview

- **Using Cloudera Search with the CDP infrastructure provides:**
  - Simplified infrastructure
  - Better production visibility and control
  - Quicker insights across various data types
  - Quicker problem resolution
  - Simplified interaction and platform access for more users and use cases beyond SQL
  - Scalability, flexibility, and reliability of search services on the same platform used to run other types of workloads on the same data
  - A unified security model across all processes with access to your data
  - Flexibility and scale in ingest and pre-processing options

The Need for Cloudera Search

- **There is significant growth in unstructured and semi-structured data**
  - Log files
  - Product reviews
  - Customer surveys
  - News releases and articles
  - Email and social media messages
  - Research reports and other documents
- **We need scalability, speed, and flexibility to keep up with this growth**
  - Relational databases can't handle this volume or variety of data
- **Decreasing storage costs make it *possible to store everything***
  - But *finding relevant data* is increasingly a problem

Cloudera Search Integrates Apache Solr with CDP

- **Apache Solr provides a high-performance search service**
  - Solr is a mature platform with widespread deployment
  - Standard Solr APIs and Web UI are available in Cloudera Search
- **Integration with CDP increases scalability and reliability**
  - The indexing and query processes can be distributed across nodes
- **Cloudera Search is 100% open source**
  - Released under the Apache Software License

Broad File Format Support

- **Cloudera Search is ideal for semi-structured and free-form text data**
  - This includes a variety of document types such as log files, email messages, reports, spreadsheets, presentations, and multimedia
- **Support for indexing data from many common formats, including**
  - Microsoft Office™ (Word, Excel, and PowerPoint)
  - Portable Document Format (PDF)
  - HTML and XML
  - UNIX mailbox format (mbox)
  - Plain text and Rich Text Format (RTF)
  - Hadoop file formats like SequenceFiles and Avro

- **Can also extract and index *metadata* from many image and audio formats**

Multilingual Support

- **You can index and query content in more than 30 languages**

"More Like This"

- **Aids in focusing results when searching on words with multiple meanings**

Showing results 1-5 out of 7,523 for term: apple

The Apple Macintosh Book
by Cary Lu (1984)
A wealth of information about the Macintosh family of computers... more like this

Wild Apple and Fruit Trees of Central Asia
by Jules Janick and Calvin Ross Sperling (2003)
The definitive source of information about Malus species found in... more like this

The Year the Big Apple Went Bust
by Fred Ferretti (1976)
Chronicles the 1975 fiscal crisis that nearly forced New York City... more like this

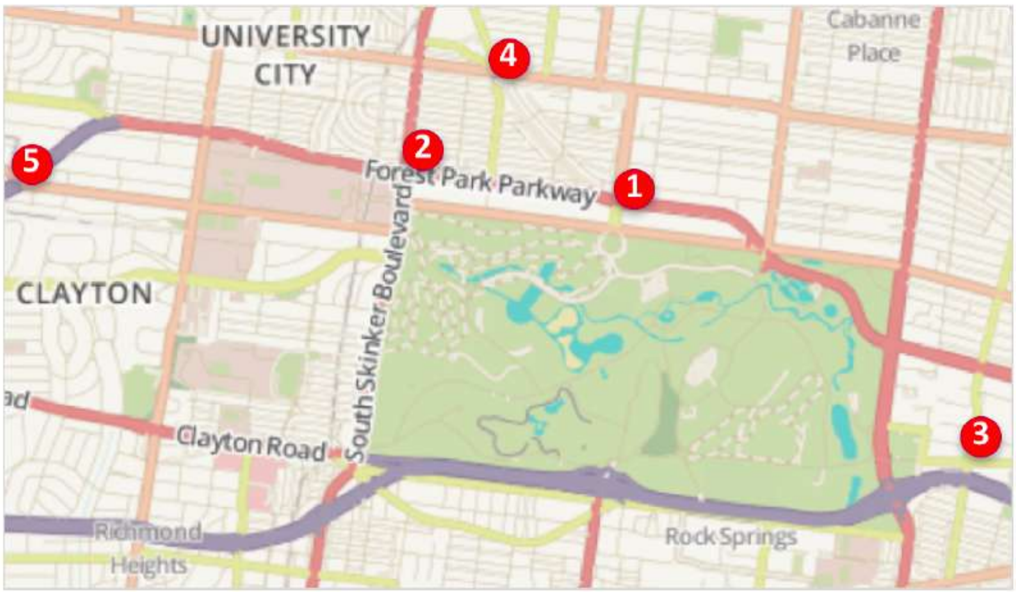
Apple of My Eye
by Patrick Redmond (2003)
When Susan and Ronnie first meet, the attraction is instant... more like this

They Were Strangers: A Family History
by Slovie Solomon Apple (1995)
Determined to survive at any cost, Clara endures untold hardships... more like this

Geospatial Search

- **Cloudera Search can use location data to filter and sort results**
  - Proximity is calculated based on longitude and latitude of each point

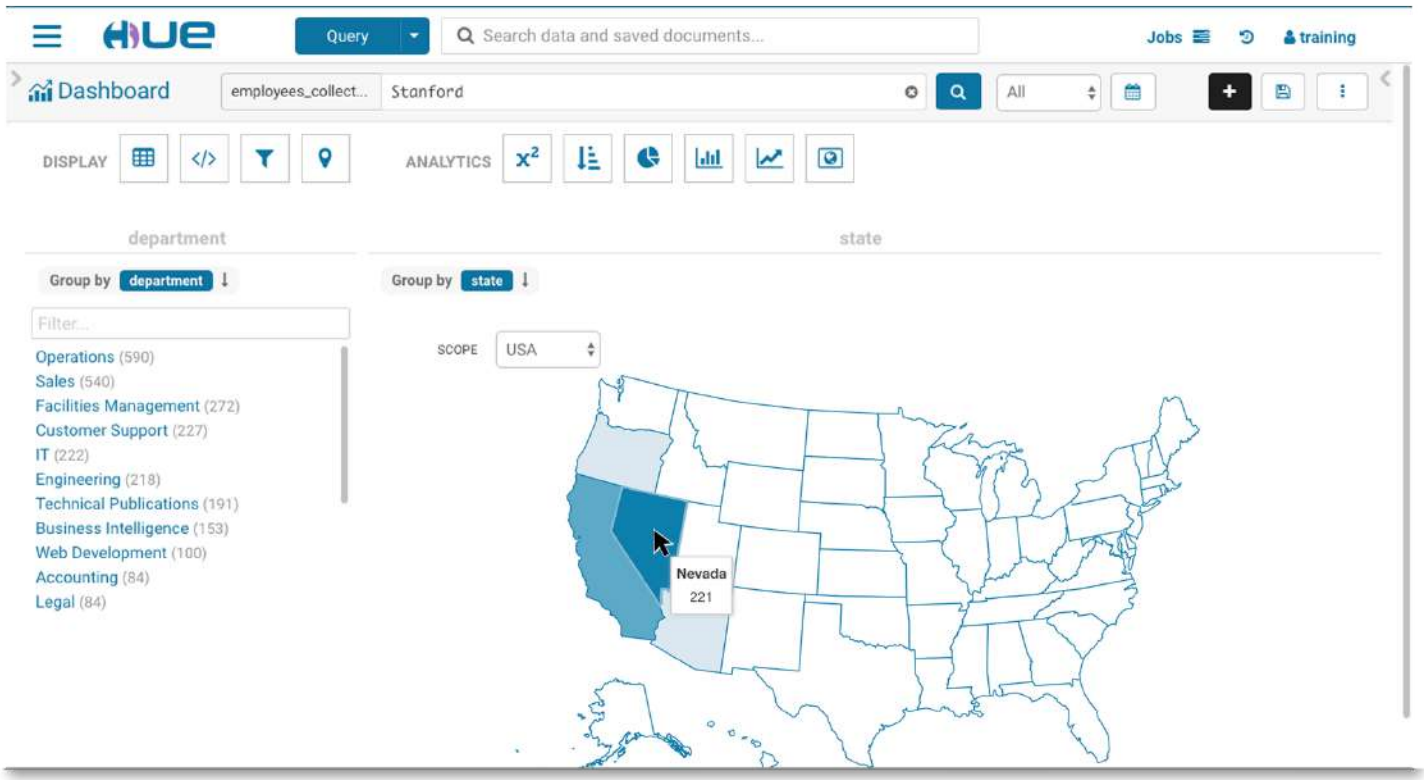
Showing all 5 results for Metrolink stations within 1 kilometer of Forest Park



1. **Forest Park Station**
   0.1 kilometers
2. **Skinker Station**
   0.2 kilometers
3. **Central West End Station**
   0.3 kilometers
4. **Delmar Station**
   0.3 kilometers
5. **Big Bend Station**
   0.9 kilometers

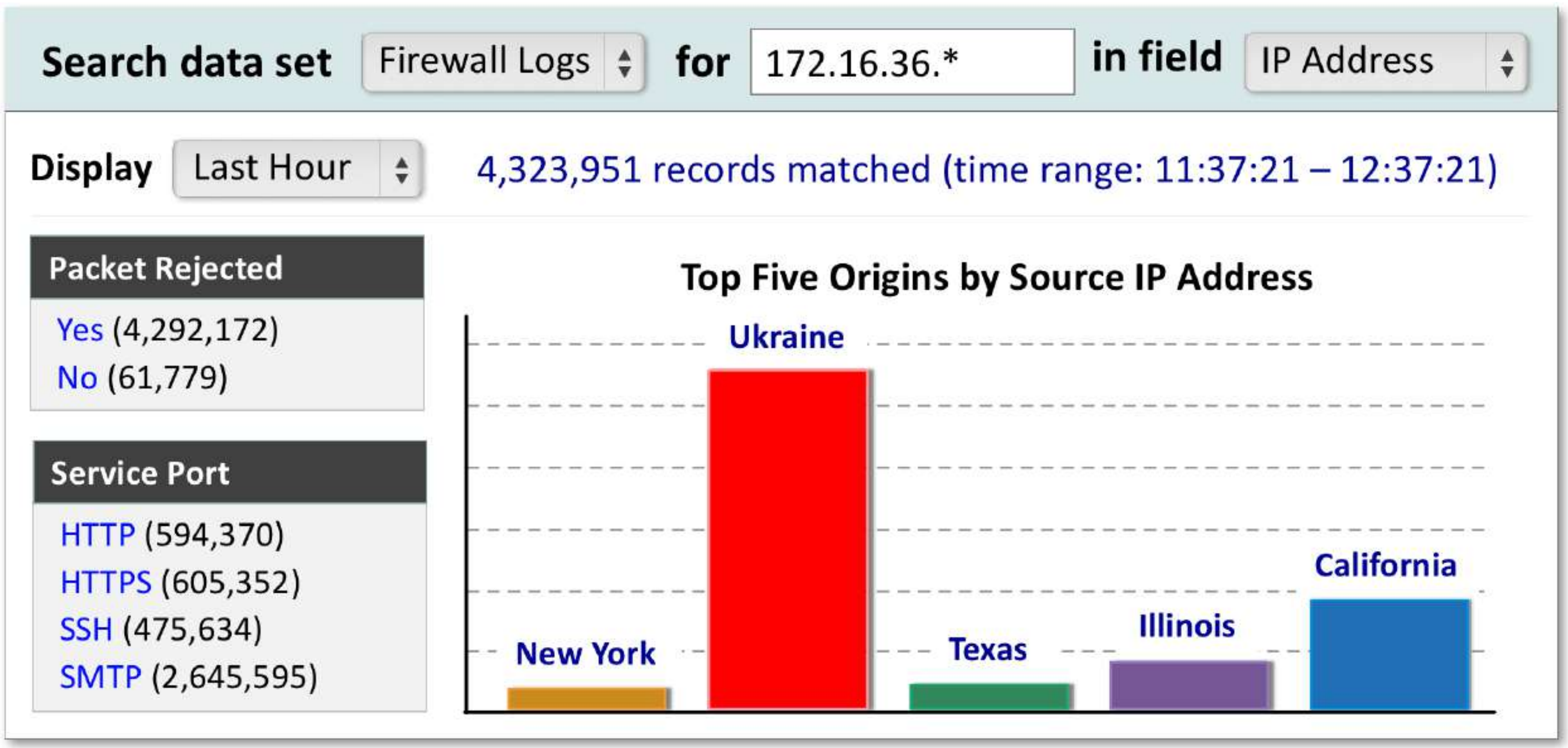
Hue: Search Dashboards

- **Hue has drag-and-drop support for building dashboards based on Search**



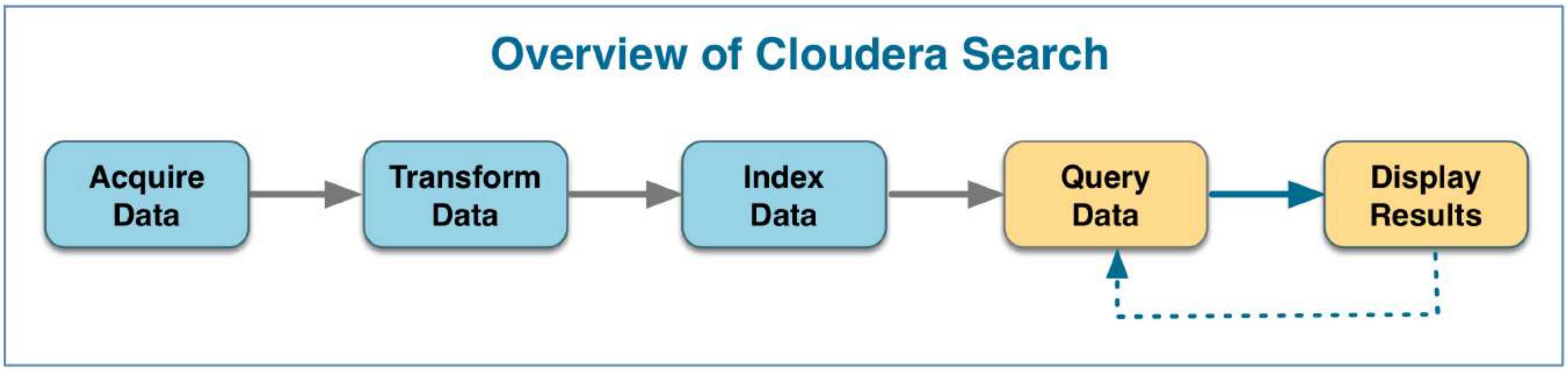
Threat Detection in Near-Real-Time

- **Looking at yesterday's log files allows us to react to history**
  - Yet emerging threats require us to react to what's happening right now
- **Search can help you identify important patterns in incoming data**



Indexing Data is a Prerequisite to Searching It

- **You must index data prior to querying that data with Cloudera Search**
- **Creating and populating an index requires specialized skills**
  - Somewhat similar to designing database tables
  - Frequently involves data extraction and transformation
- **Running basic queries on that data requires relatively little skill**
  - "Power users" who master the syntax can create very powerful queries



Deploying Cloudera Search

- When you deploy Cloudera Search, SolrCloud partitions your data set into multiple indexes and processes
- Search uses ZooKeeper to simplify management, which results in a cluster of coordinating Apache Solr servers
- The Add a Service wizard will automatically configure and initialize the Solr service
- See instructions for your specific version for details
- Cloudera has a *three-day course* on the details regarding the configuration and usage of Search

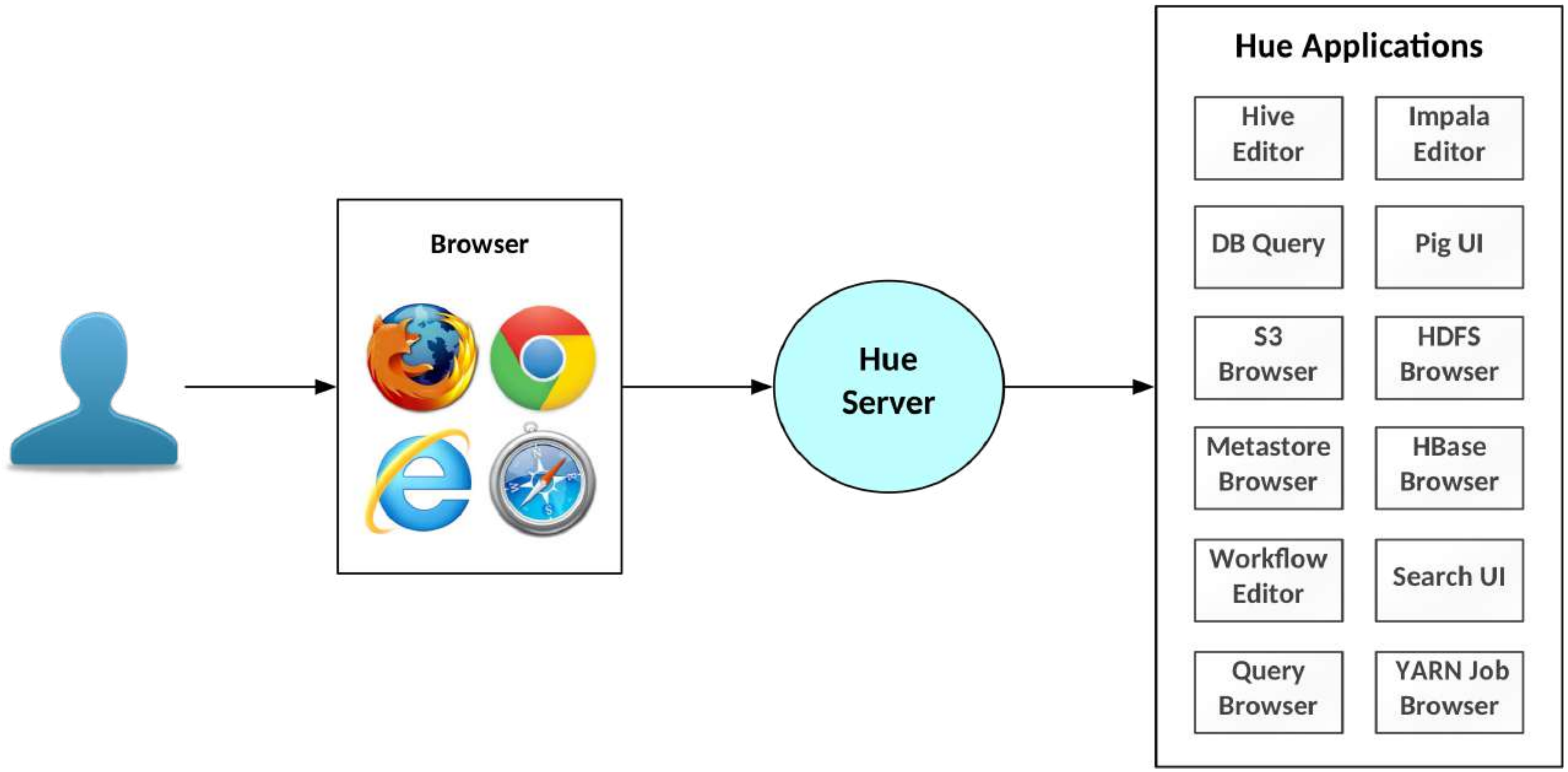
Chapter Topics

Data Access and Discovery

- Apache Hive
- Apache Impala
- Apache Impala Tuning
- Hands-On Exercise: Install Impala and Hue
- Search Overview
- **Hue Overview**
- Managing and Configuring Hue
- Hue Authentication and Authorization
- CDSW Overview
- Essential Points
- Hands-On Exercise: Using Hue, Hive and Impala

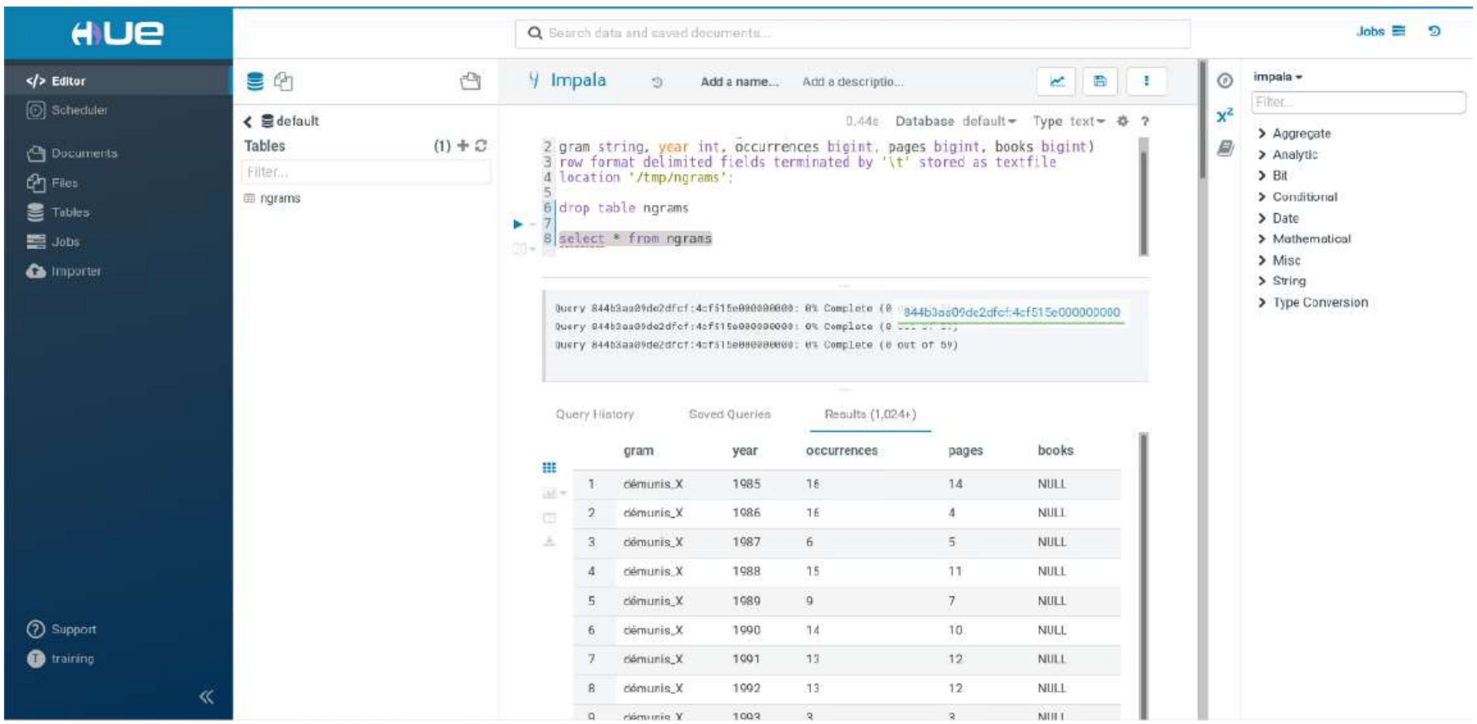
What Is Hue?

- **Hue provides a web interface for interacting with a Hadoop cluster**
  - Hue Applications run in the browser (no client-side installation)



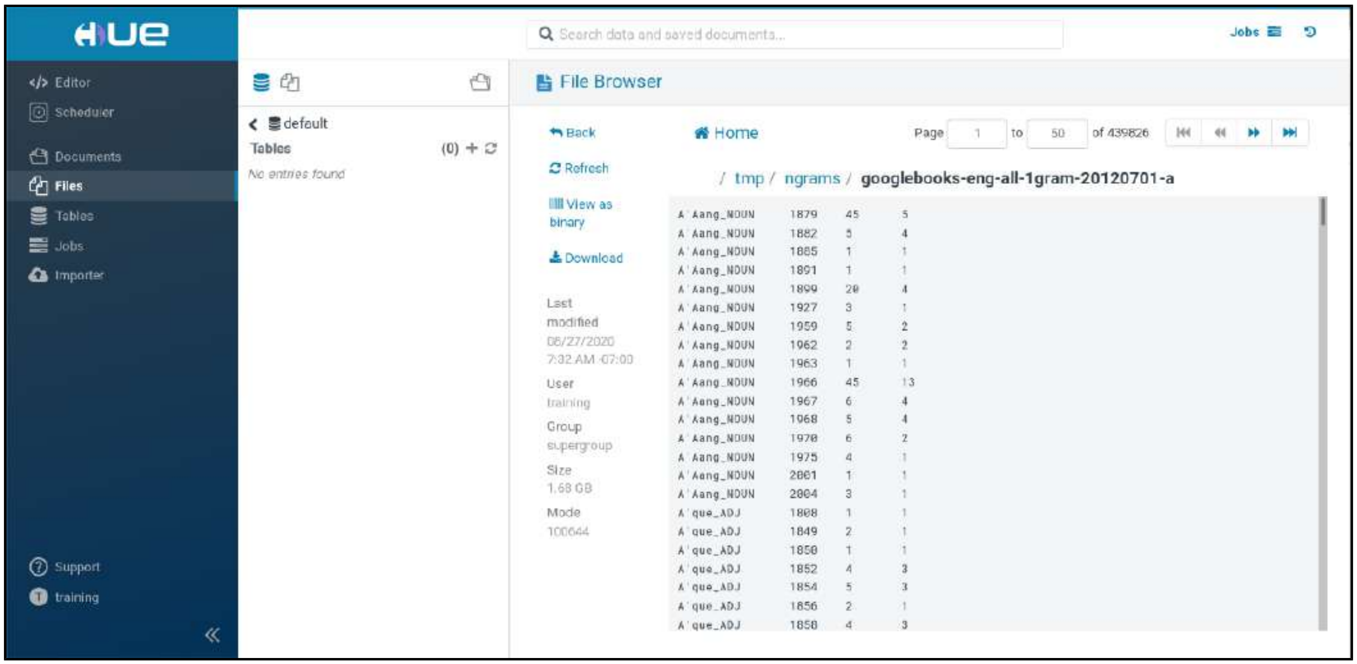
Key Hue Applications: Query Editor

- View tables, schema, and sample data
- Enter queries with assist and autocomplete
- Execute Impala or Hive queries and view and chart results



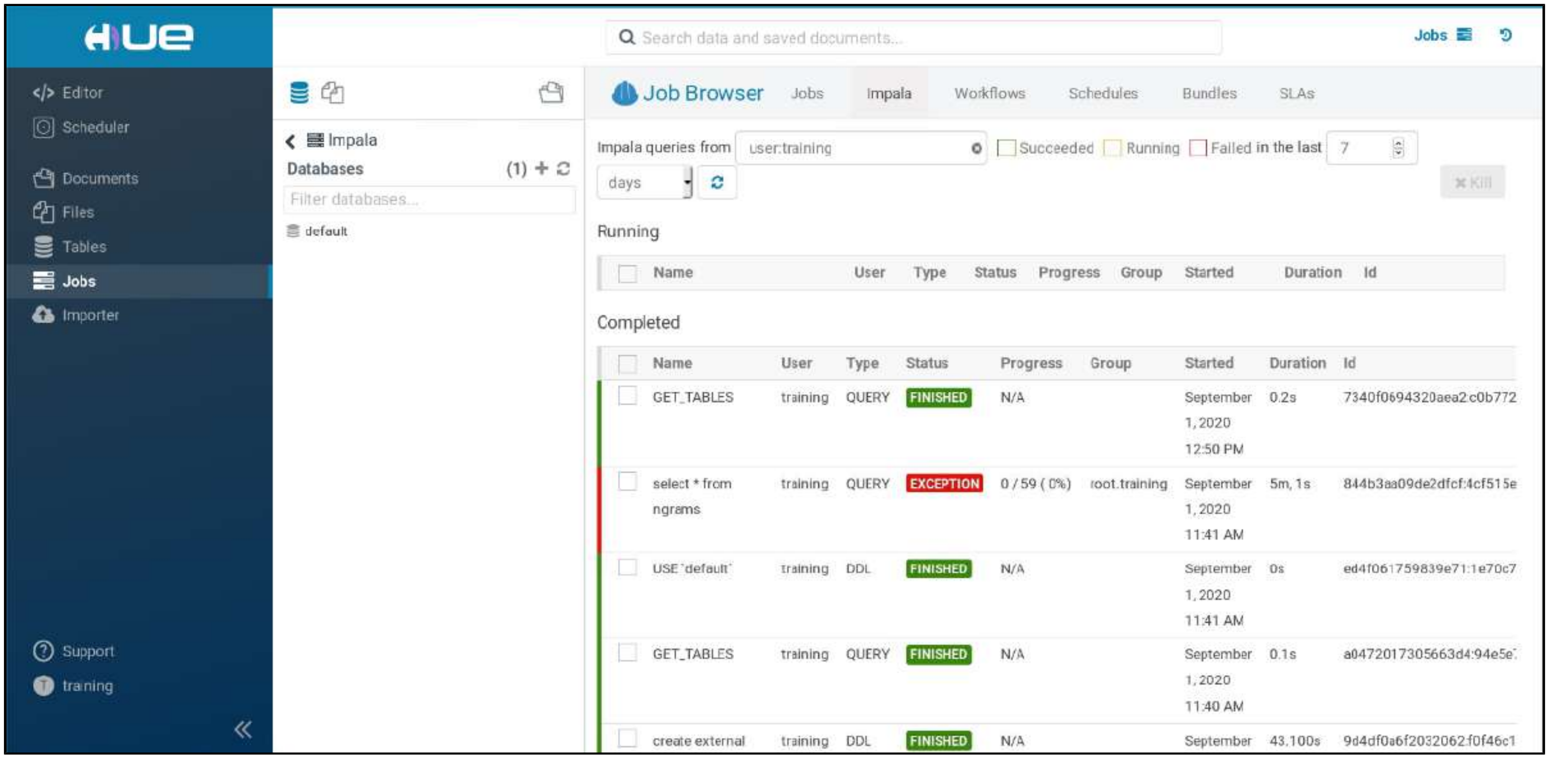
Key Hue Applications: File Browser

- **Browse directories and files in HDFS**
- **Upload, download, and manage files**
- **View details and file contents (for supported file formats)**



Key Hue Applications: Job Browser

- Monitor MapReduce and Spark YARN jobs, Impala queries, and Oozie workflows
- View job details and metrics



Chapter Topics

Data Access and Discovery

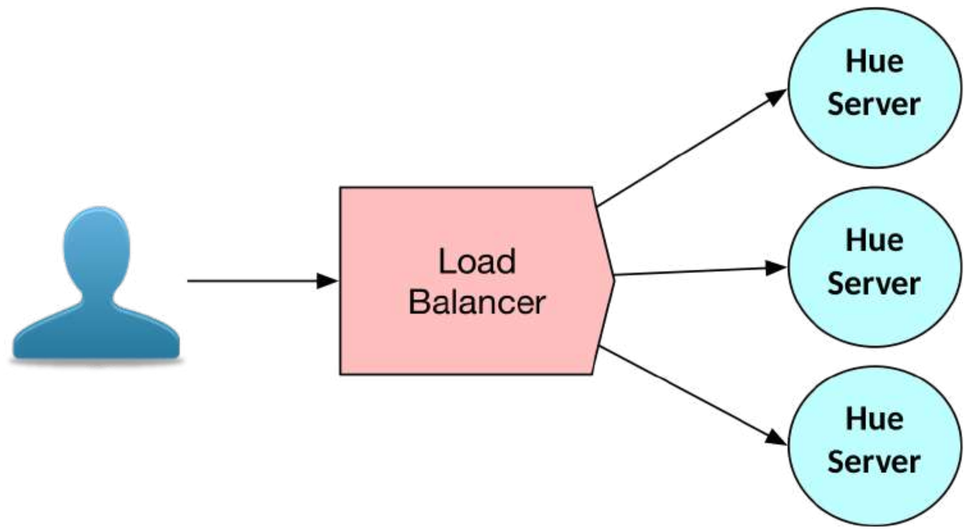
- Apache Hive
- Apache Impala
- Apache Impala Tuning
- Hands-On Exercise: Install Impala and Hue
- Search Overview
- Hue Overview
- **Managing and Configuring Hue**
- Hue Authentication and Authorization
- CDSW Overview
- Essential Points
- Hands-On Exercise: Using Hue, Hive and Impala

Installing and Accessing Hue

- **Installing Hue**
  - Before installing Hue, verify services are installed on the cluster
    - Required components for base Hue:
      - HDFS
    - Optional components to enable more Hue applications:
      - YARN, Oozie, Hive, Impala, HBase, Search, Spark, Sentry, Sqoop, ZooKeeper, HttpFS, WebHDFS, S3 Service
- **Access Hue from a browser**
  - Default Hue server port 8888
  - Default Load Balancer port 8889
  - Example: `http://`*`hue_server`*`:8888`

Hue Roles

- **Hue Server—provides user access to the UI**
- **Load Balancer—supports adding multiple Hue Servers for better performance**
  - Provides automatic failover



Hue Database

- **Hue stores user data and other information in the Hue database**
- **Hue is packaged with a lightweight embedded PostgreSQL database**
  - For proof-of-concept deployments
  - For production, use an external database
- **Supported production database systems**
  - MySQL
  - MariaDB
  - Oracle
  - PostgreSQL

Select Hue Applications—Requirements

Hue Application	Required Component and Configuration
Hive Query Editor Metastore Manager	■ HiveServer2 installed
Impala Query Editor	■ A shared Hive metastore ■ Impala service ■ Impala Gateway role on Hue server host
File Browser	■ HttpFS installed or WebHDFS enabled ■ HttpFS is required for HDFS HA deployments
Job Browser	■ YARN ■ Spark (optional)

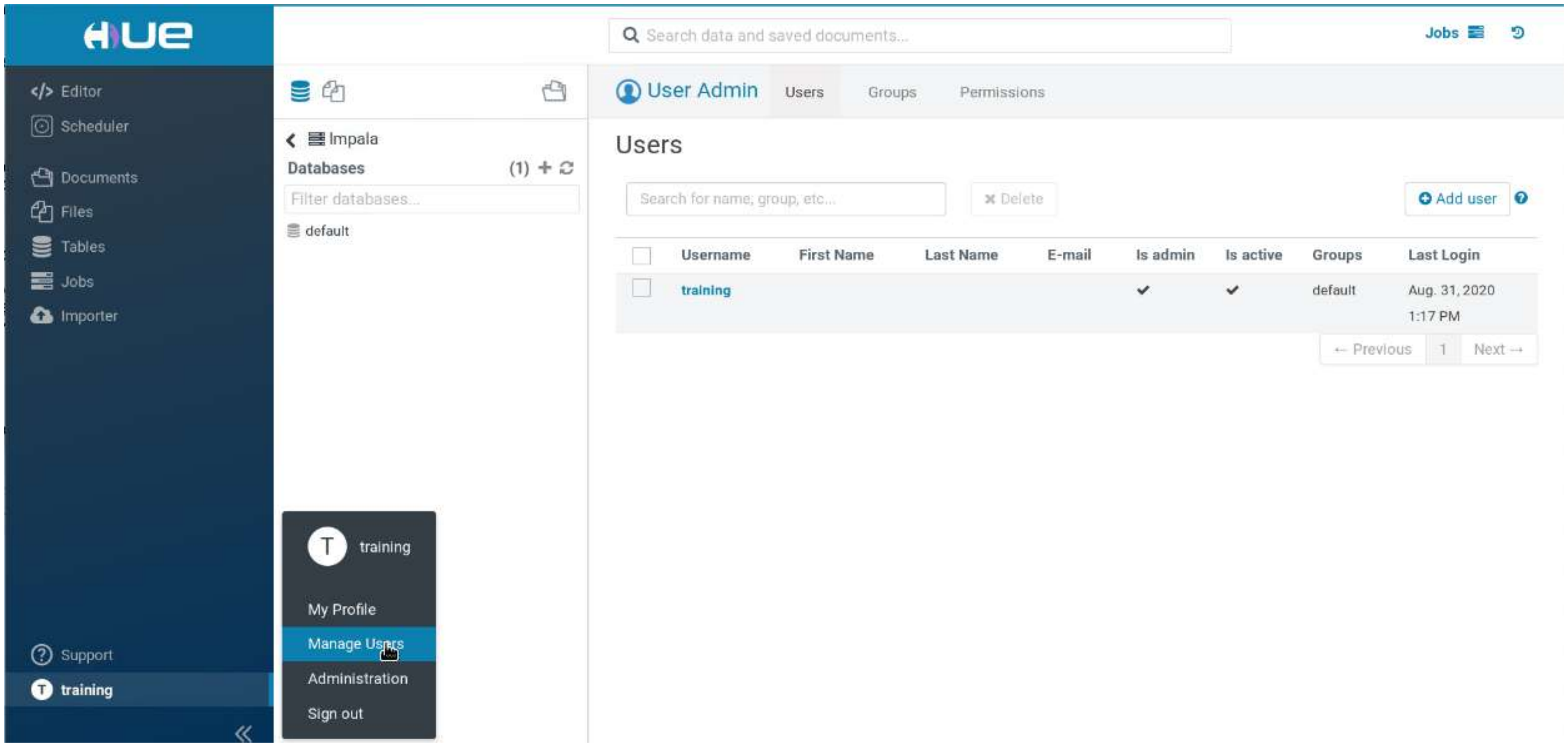
Chapter Topics

Data Access and Discovery

- Apache Hive
- Apache Impala
- Apache Impala Tuning
- Hands-On Exercise: Install Impala and Hue
- Search Overview
- Hue Overview
- Managing and Configuring Hue
- **Hue Authentication and Authorization**
- CDSW Overview
- Essential Points
- Hands-On Exercise: Using Hue, Hive and Impala

Managing Users in Hue

- **Use the Hue User Admin page to manage Hue users**

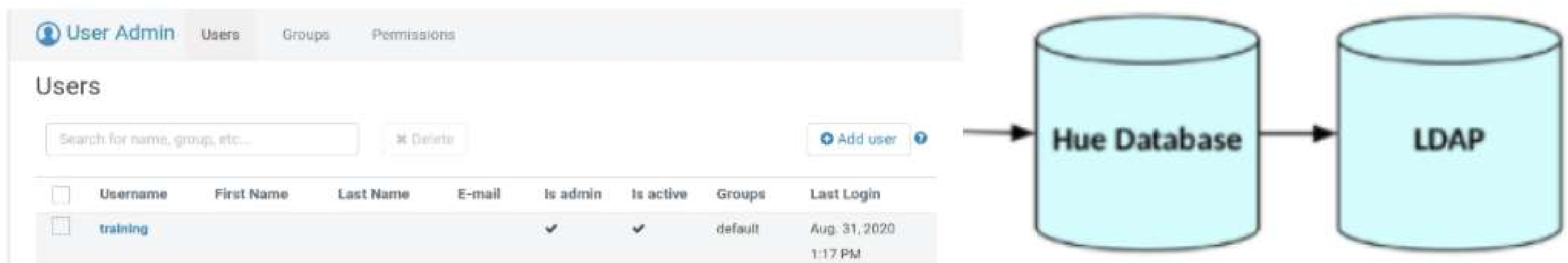


User Information

- **By default, all user information is stored in the Hue database**
  - Such as credentials and profile settings

Accessing Users and Groups From LDAP—Option 1

- **Administrator configures Hue to use an LDAP directory**
- **Administrator syncs users and groups from LDAP to the Hue database**
- **Hue authenticates users by accessing credentials from the Hue database**

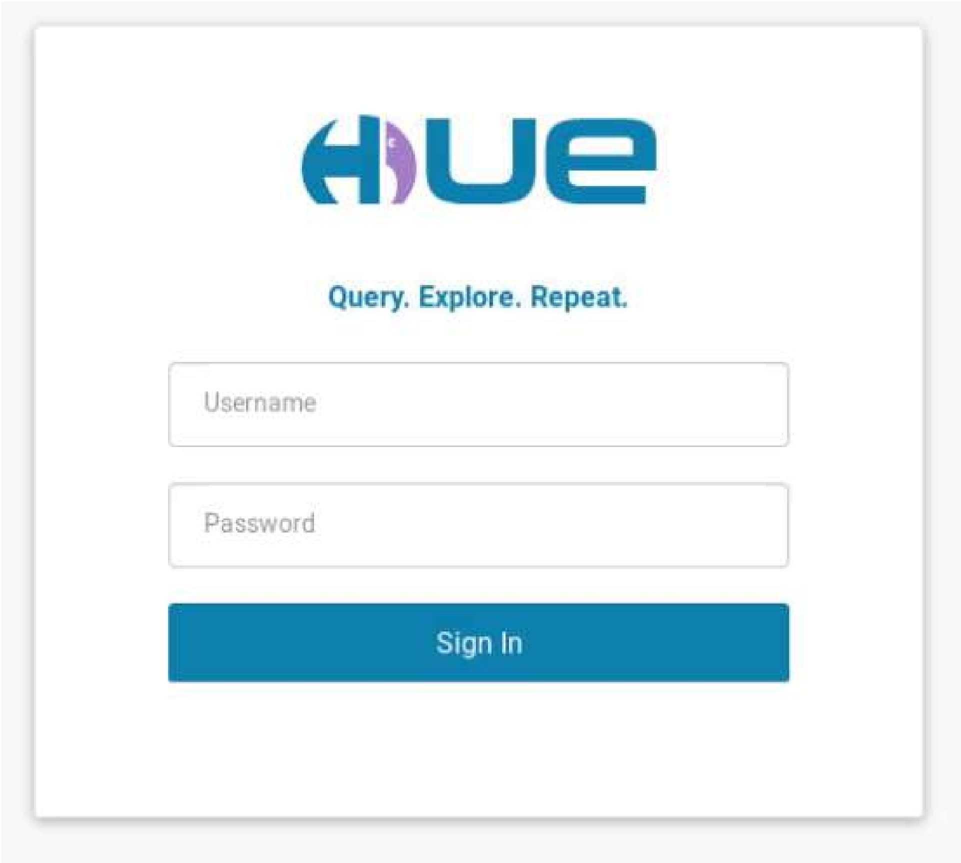


Accessing Users and Groups From LDAP—Option 2

- **Administrator configures Hue to access the LDAP directory**
- **Hue authenticates users by accessing credentials from LDAP**

First User Login

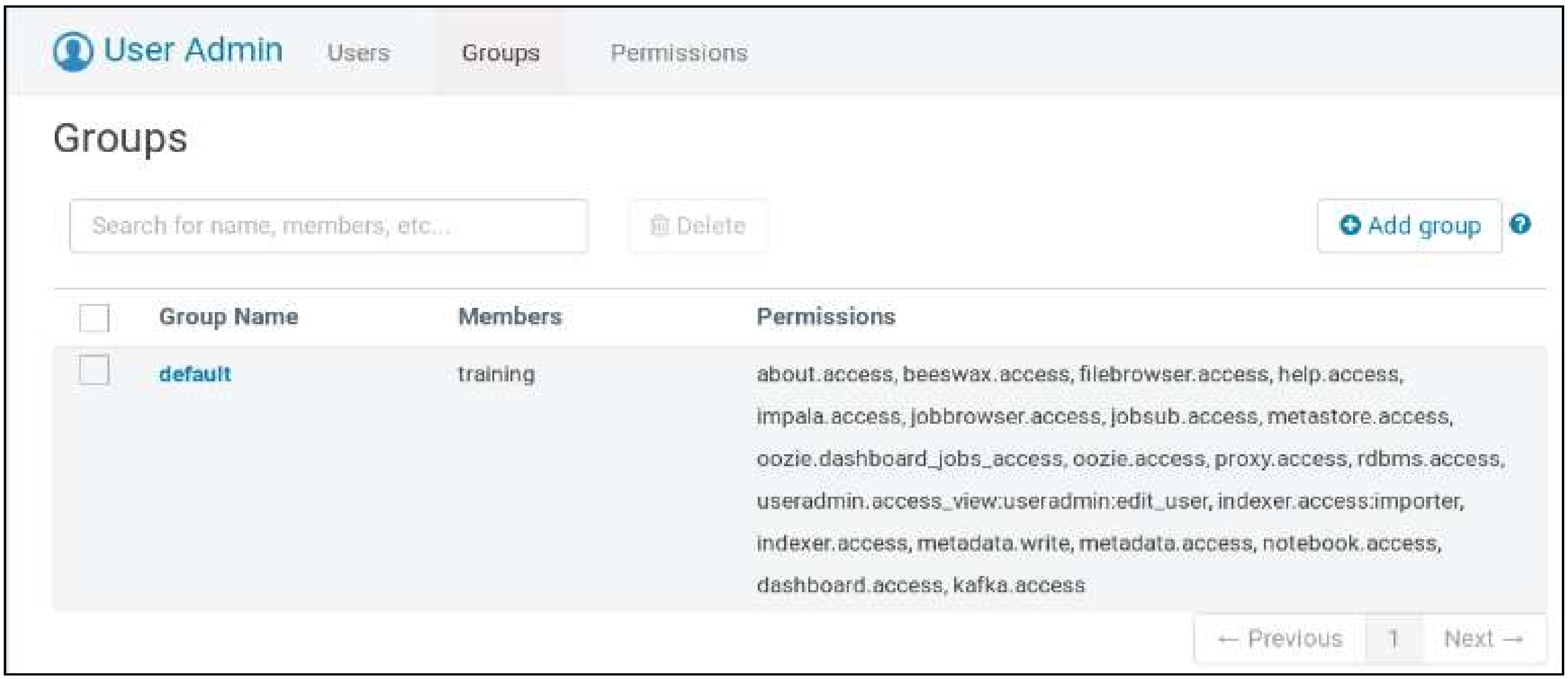
- The first user to log in to Hue receives superuser privileges automatically
- Superusers can be added and removed



Restricting Access to Hue Features

- **Use *groups* to manage user access**
- **Configure a group with a set of permissions**
- **Assign users to one or more groups**
- **The default group has access to every Hue application**

Manage Users > Groups tab



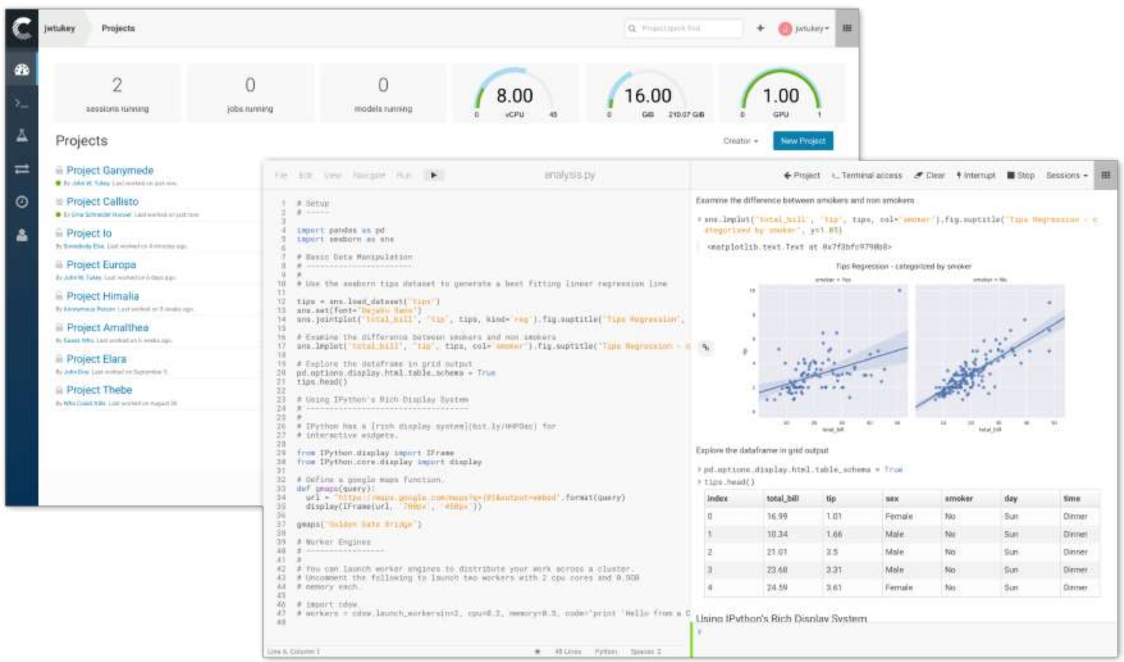
Chapter Topics

Data Access and Discovery

- Apache Hive
- Apache Impala
- Apache Impala Tuning
- Hands-On Exercise: Install Impala and Hue
- Search Overview
- Hue Overview
- Managing and Configuring Hue
- Hue Authentication and Authorization
- **CDSW Overview**
- Essential Points
- Hands-On Exercise: Using Hue, Hive and Impala

Cloudera Data Science Workbench (CDSW)

- **Enables fast, easy, and secure self-service data science for the enterprise**
  - Web browser-based interface
  - Direct access to a secure Cloudera cluster running Spark and other tools
  - Isolated environments running Python, R, and Scala
  - Teams, version control, collaboration, and sharing



How Cloudera Data Science Workbench Works

- **The CDSW host runs on a cluster gateway (edge) node**
  - Front end: Serves the CDSW web application
  - Back end: Hosts Docker containers running Python, R, or Scala
- **The Docker containers run user workloads**
  - Enables multitenancy with isolation and security
  - Each container can run different packages and versions
  - Each container provides a virtual gateway with secure access to the cluster
- **IT can easily add more CDSW worker hosts for greater numbers of users**
  - CDSW schedules containers across multiple hosts

How to Use Cloudera Data Science Workbench

1. **Log into the CDSW web application**
2. **Open an existing project or create a new one**
   - Create a new project by selecting a template, cloning a Git repository, forking an existing project, or uploading files
3. **Browse and edit scripts in the project**
4. **When ready to run code, launch a new session**
   - This starts a Docker container running a Python, R, or Scala
   - Run scripts in the session
   - Execute commands at the session prompt
   - View output in the session console
5. **When finished, stop the session**
   - CDSW saves the session output
   - CDSW saves all installed packages in the project

Installing Cloudera Data Science Workbench

- **Installation of CDSW can be a very complicated process**
- **See version specific instructions**
- **Use the following high-level steps to install Cloudera Data Science Workbench using Cloudera Manager:**
  - Secure your hosts, set up DNS subdomains, and configure Docker block devices
  - Configure Apache Spark 2
  - Configure JAVA_HOME
  - Download and Install the Cloudera Data Science Workbench CDSW
  - Install the Cloudera Data Science Workbench Parcel
  - Add the Cloudera Data Science Workbench Service

Troubleshooting CDSW

- **Check the current status of the application and run validation checks**
  - You can use the Status and Validate commands in Cloudera Manager
    - Status: Checks the current status of Cloudera Data Science Workbench
    - Validate: Runs common diagnostic checks to ensure all internal components are configured and running as expected
  - To run these commands, select Status or Validate from the Actions menu for CDSW on the Home Status page

CDSW Administration

- **Administrative tasks that can only be performed by a CDSW site administrator:**
  - Monitoring Site Activity
  - Configuring Quotas
  - CDSW in Cloudera Manager
  - Data Collection
  - Email with SMTP
  - License Keys
  - User Access to Features
- **For more information see the CDSW Administration Guide**

Video: Using Cloudera Data Science Workbench

- **Your instructor will show a video demonstrating the use of CDSW**
- **Click `here` for Video**

Chapter Topics

Data Access and Discovery

- Apache Hive
- Apache Impala
- Apache Impala Tuning
- Hands-On Exercise: Install Impala and Hue
- Search Overview
- Hue Overview
- Managing and Configuring Hue
- Hue Authentication and Authorization
- CDSW Overview
- **Essential Points**
- Hands-On Exercise: Using Hue, Hive and Impala

Essential Points (1)

- **Hive provides a SQL-like interface for running queries on data in HDFS or object storage**
- **Deploy the Hive Metastore Server (remote mode) to support Impala and JDBC access**
- **Impala also supports SQL queries, provides faster performance than Hive**
  - Runs Impala daemons on worker hosts
  - Like Hive, it can query data in HDFS, HBase, or object storage
  - Access using the `impala-shell` CLI or Hue
- **Hive and Impala use a common shared metastore**
  - Stores metadata such as column names and data types
  - Metastore resides in a relational database

Essential Points (2)

- **Cloudera Search provides easy, natural language access to data stored in HDFS, HBase, or the cloud**
- **Hue provides a web interface for interacting with a Hadoop cluster**
- **CDSW enables fast, easy, and secure self-service data science for the enterprise**
- **Hive and Impala use a common shared metastore**

Chapter Topics

Data Access and Discovery

- Apache Hive
- Apache Impala
- Apache Impala Tuning
- Hands-On Exercise: Install Impala and Hue
- Search Overview
- Hue Overview
- Managing and Configuring Hue
- Hue Authentication and Authorization
- CDSW Overview
- Essential Points
- **Hands-On Exercise: Using Hue, Hive and Impala**

Hands-On Exercise: Using Hue, Hive and Impala

- **In this exercise, you will test Hue applications, configure a limited-access user group for analysts, then define Hive tables in HDFS, run Hive on Spark, and use the Beeline and Hue interfaces to run Impala queries.**
- **Please refer to the Hands-On Exercise Manual for instructions**

Data Compute

Chapter 9

Course Chapters

- Introduction
- Cloudera Data Platform
- CDP Private Cloud Base Installation
- Cluster Configuration
- Data Storage
- Data Ingest
- Data Flow
- Data Access and Discovery
- **Data Compute**
- Managing Resources
- Planning Your Cluster
- Advanced Cluster Configuration
- Cluster Maintenance
- Cluster Monitoring
- Cluster Troubleshooting
- Security
- Private Cloud / Public Cloud
- Conclusion
- Appendix: Cloudera Manager API
- Appendix: Ozone Overview

Data Compute

After completing this chapter, you will be able to

- **Describe the role of distributed application frameworks in CDP**
- **Explain the purpose and basic architecture of YARN**
- **Explain how YARN handles failure**
- **Summarize how MapReduce runs on the cluster**
- **Manage YARN applications**
- **Describe the use of Tez for Hive queries**
- **Explain the use of ACID for Hive**
- **Describe Spark and how Spark applications run**
- **Summarize how to monitor Spark applications**

Chapter Topics

Data Compute

- **YARN Overview**
- Running Applications on YARN
- Viewing YARN Applications
- YARN Application Logs
- MapReduce Applications
- YARN Memory and CPU Settings
- Hands-On Exercise: Running YARN Applications
- Tez Overview
- ACID for Hive
- Spark Overview
- How Spark Applications Run on YARN
- Monitoring Spark Applications
- Essential Points
- Hands-On Exercise: Running Spark Applications

What is Apache Hadoop YARN?

- **Yet Another Resource Negotiator (YARN)**
- **A platform for managing resources for applications on a cluster**
- **Use resource management to:**
  - Guarantee completion in a reasonable time frame for critical workloads
  - Support reasonable cluster scheduling between groups of users
  - Prevent users from depriving other users access to the cluster

Frameworks for Distributed Applications

- **Distributed processing applications use frameworks to provide**
  - Batch processing
  - SQL queries
  - Search
  - Machine learning
  - Stream processing
- **The framework provides applications access to data in the cluster**
  - HDFS, HBase, Kudu, or object storage
- **Applications compete for resources**
  - YARN manages resources between applications

Major Frameworks on YARN

- **MapReduce**
  - The original Hadoop application framework
  - Proven, widely used
  - Sqoop and other tools use MapReduce to interact with HDFS
- **Spark**
  - A big data processing framework
  - Built-in modules for streaming, SQL, machine learning, and graph processing
  - Supports processing of streaming data
  - Faster than MapReduce for most workloads
- **Tez**
  - Provides a developer API and framework to write native YARN applications
  - Tez is extensible and embeddable
  - Has set the standard for true integration with YARN for interactive workloads
  - Hive embeds Tez so that it can translate complex SQL statements into interactive and highly optimized queries

YARN Benefits

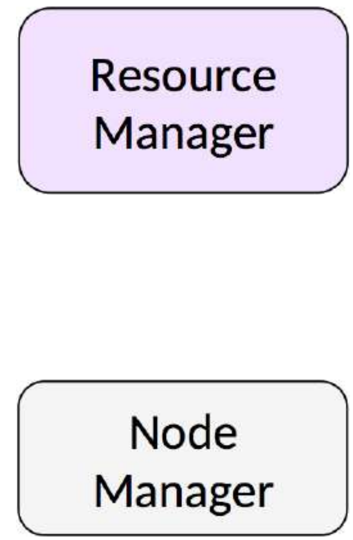
- **Diverse workloads can run on the same cluster**
- **Memory and CPU shared dynamically between applications**
- **Predictable performance**
  - Avoids “oversubscribing” hosts
    - Requesting more CPU or RAM than is available
  - Allows higher-priority workloads to take precedence
- **Full cluster utilization**

Chapter Topics

Data Compute

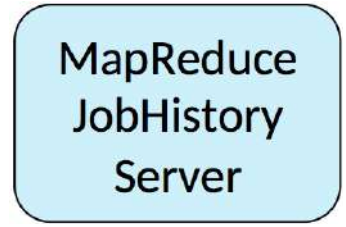
YARN Services (1)

- **ResourceManager**
  - Runs on a master host
  - High availability mode: two RMs per cluster
  - Classic mode: one RM per cluster
  - Schedules resource usage on worker hosts
- **NodeManager**
  - Many per cluster
  - Runs on worker hosts
  - Launches containers that run applications
  - Manages resources used by applications
  - Usually colocated with HDFS DataNodes



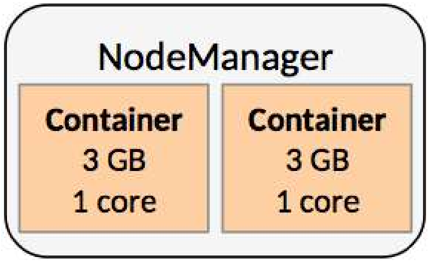
YARN Services (2)

- **MapReduce JobHistory Server**
  - One per cluster
  - Runs on a master host
  - Archives MapReduce jobs' metrics and metadata
  - Spark has a separate history server



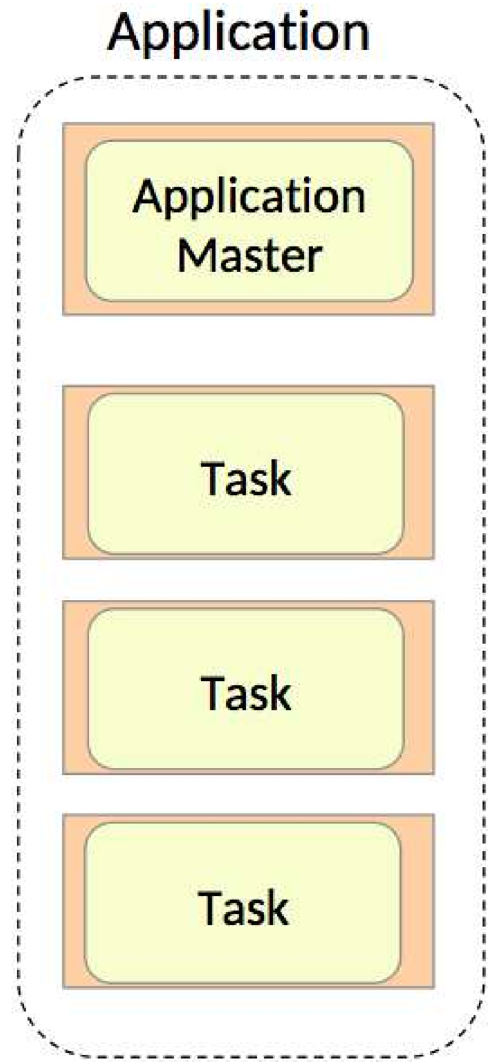
Applications on YARN (1)

- **Containers**
  - Allocated by ResourceManager's scheduler
  - Granted a specific amount of memory and CPU on a NodeManager host
  - Applications run in one or more containers
- **ApplicationMaster**
  - One per running application
  - Framework/application specific
  - Runs in a container
  - Requests more containers to run application tasks



Applications on YARN (2)

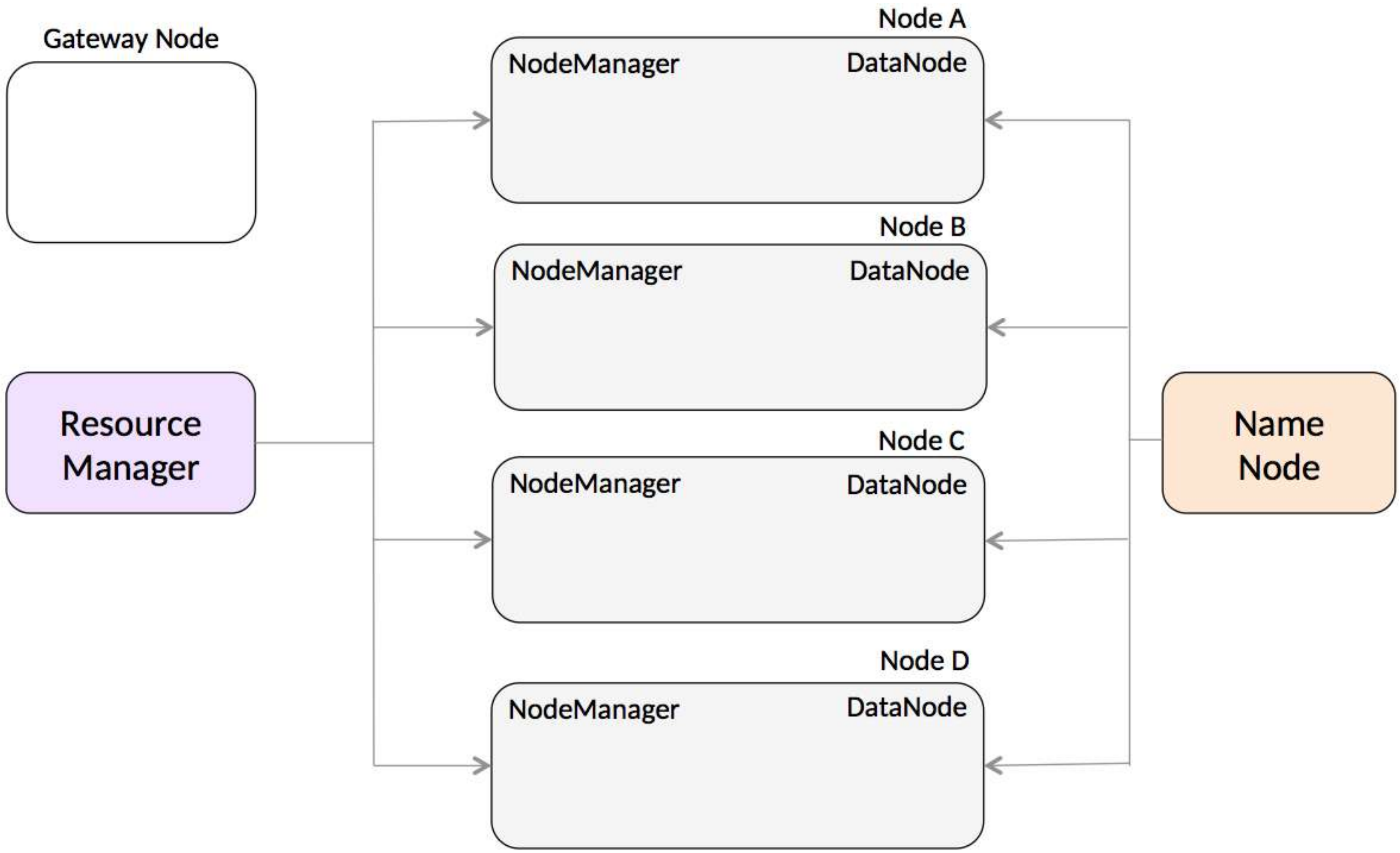
- **Each application consists of one or more containers**
  - The ApplicationMaster runs in one container
  - The application's distributed processes run in other containers
    - Processes run in parallel, managed by an ApplicationMaster
    - Processes called *executors* in Spark and *tasks* in MapReduce
- **Applications are typically submitted from a gateway host**



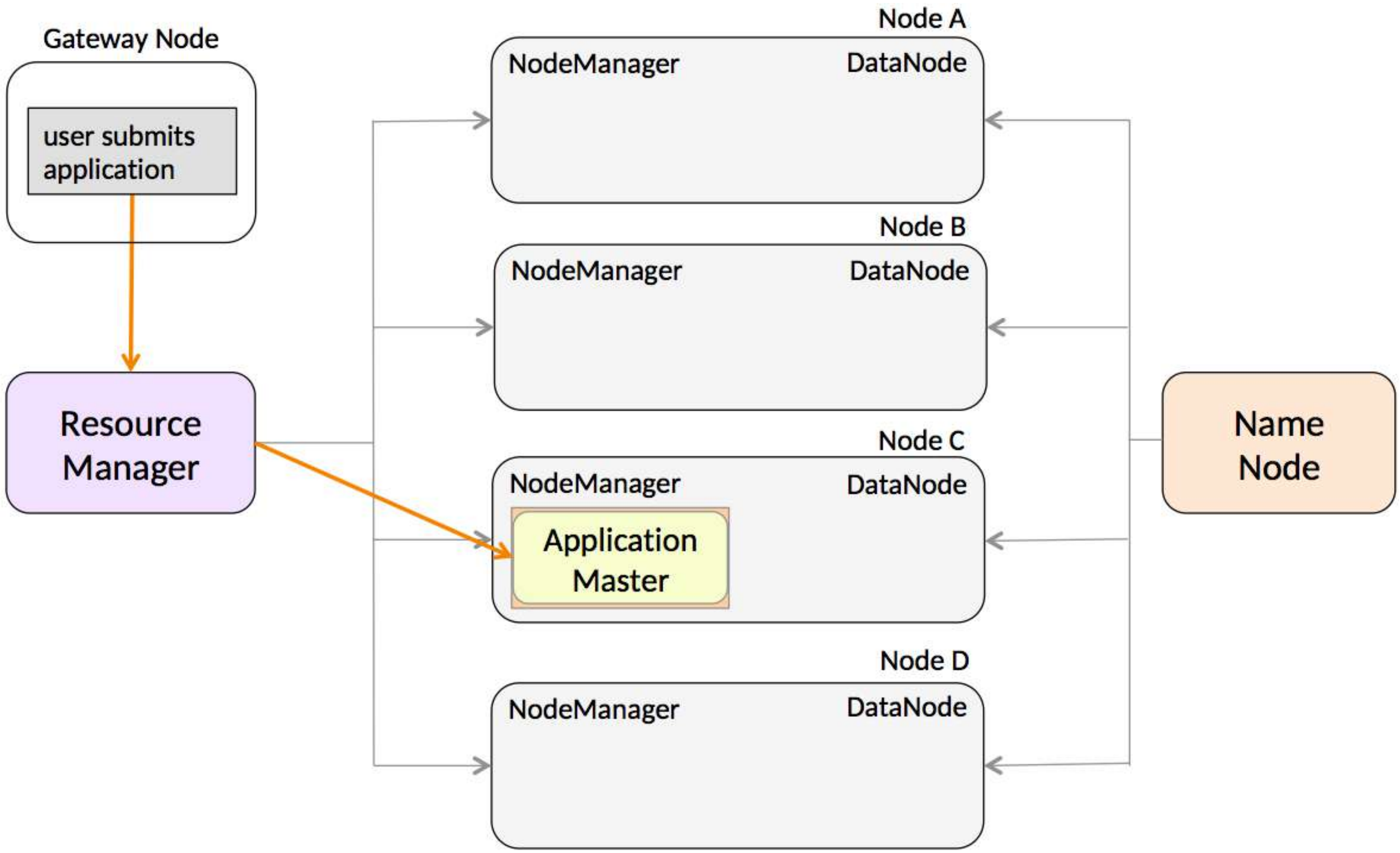
Cluster Resource Allocation

1. **ResourceManager allocates one container for the application's ApplicationMaster**
2. **The ApplicationMaster requests additional containers from the ResourceManager**
   - Requests include number of containers, preferred hosts or racks if any, required vcores and memory
3. **ResourceManager allocates containers**
   - Passes locations and container IDs to the Application Master
4. **The ApplicationMaster distributes tasks to run in container JVMs**
5. **NodeManagers start container JVMs and monitor resource consumption**

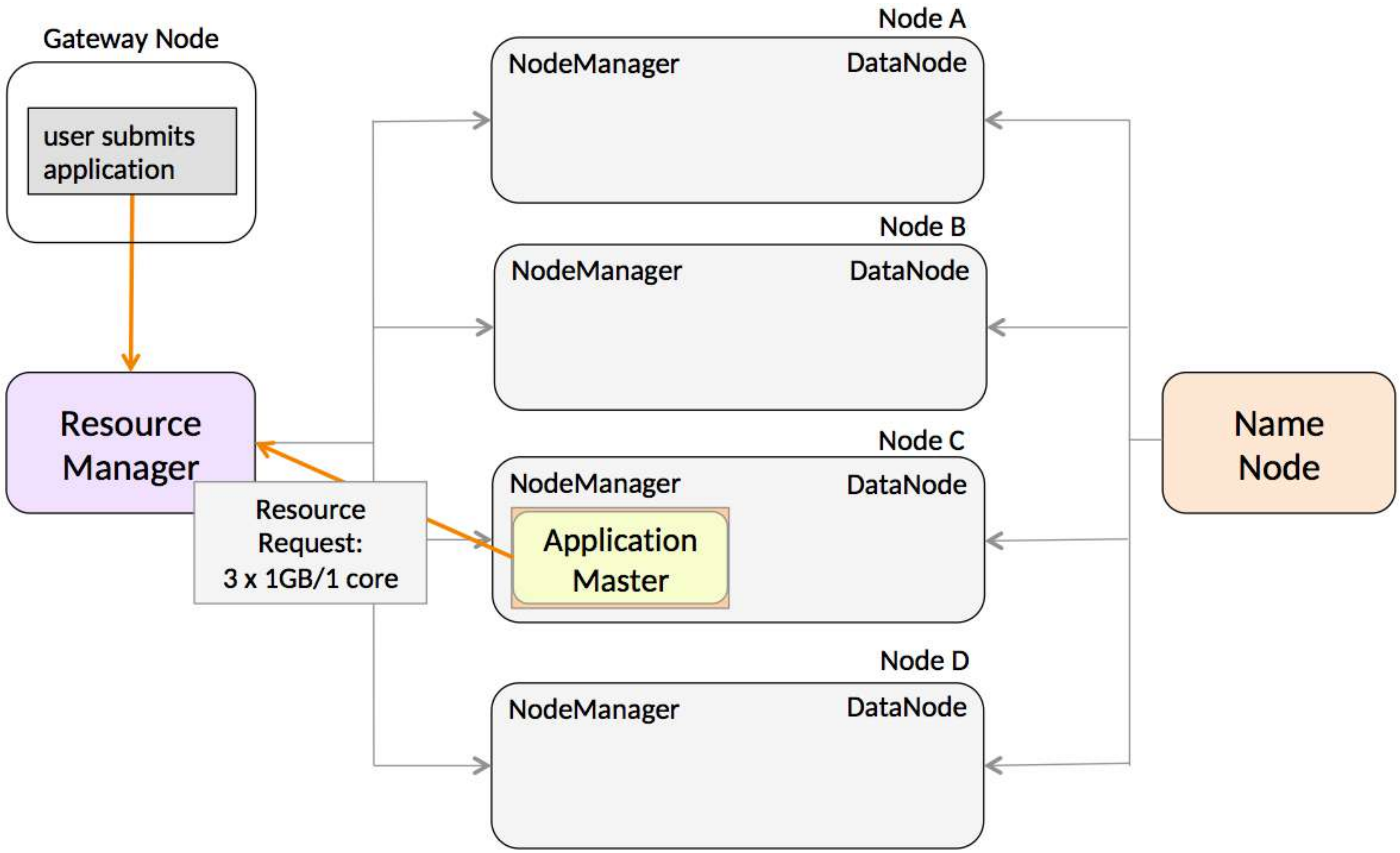
Running an Application on YARN (1)



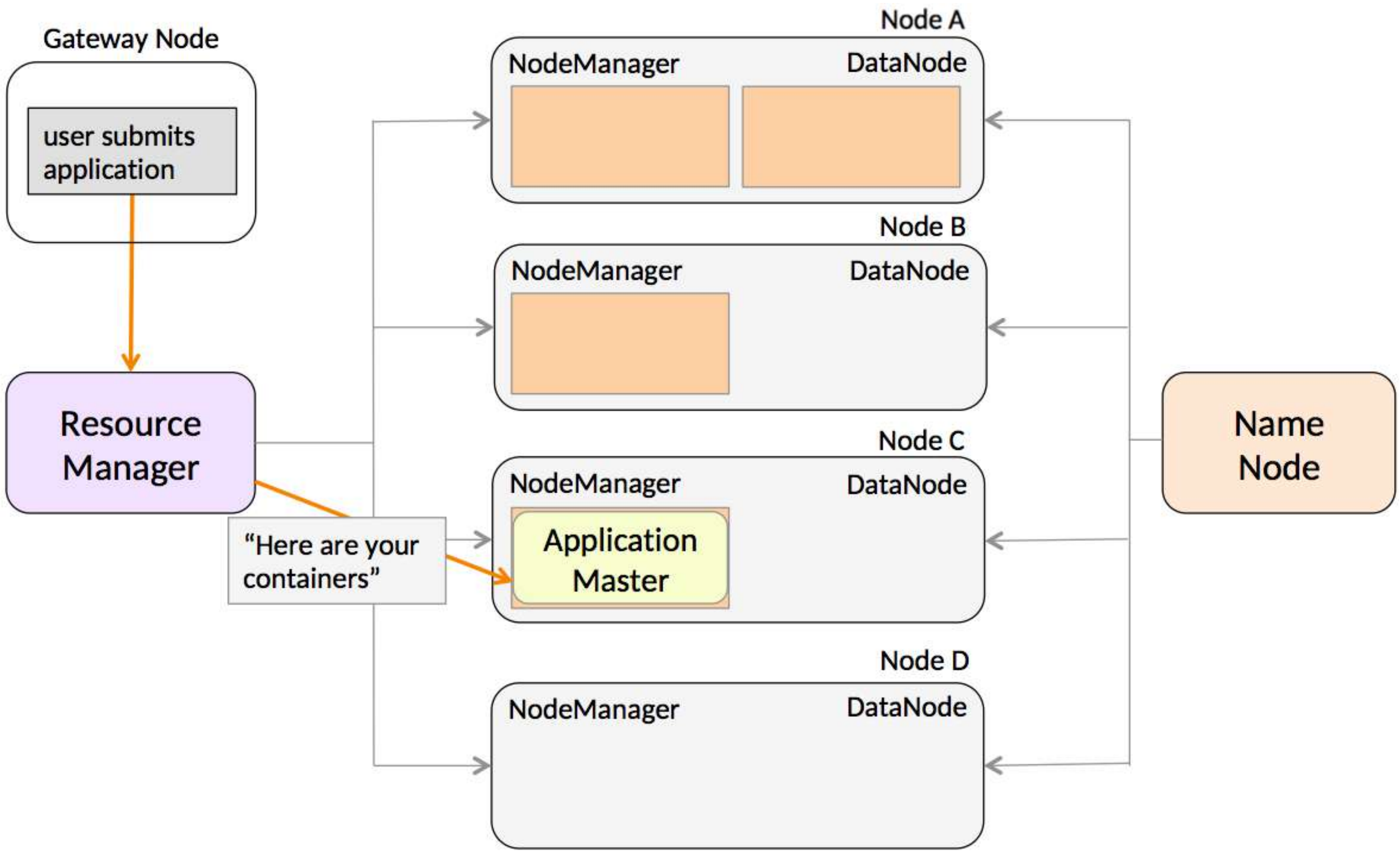
Running an Application on YARN (2)



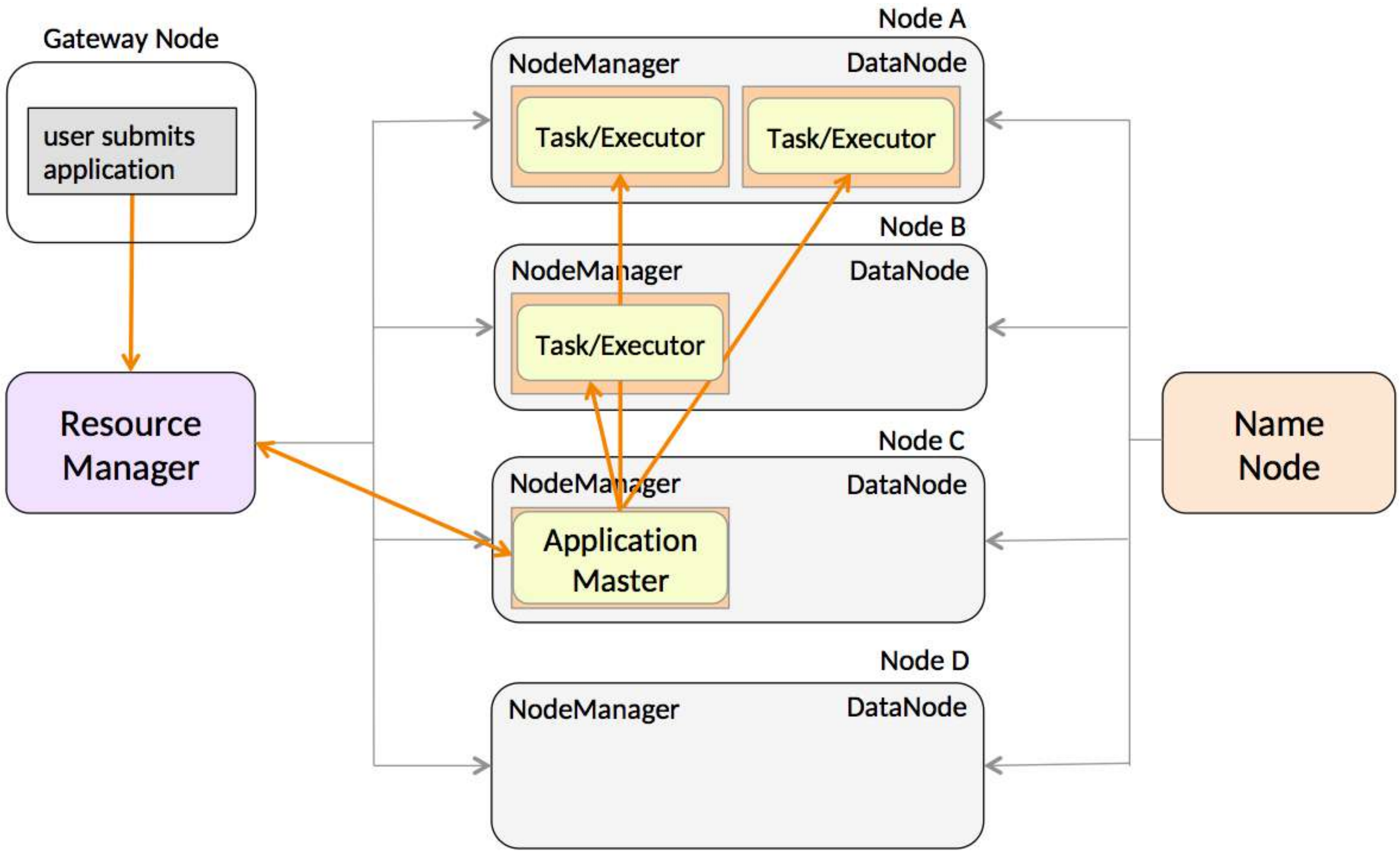
Running an Application on YARN (3)



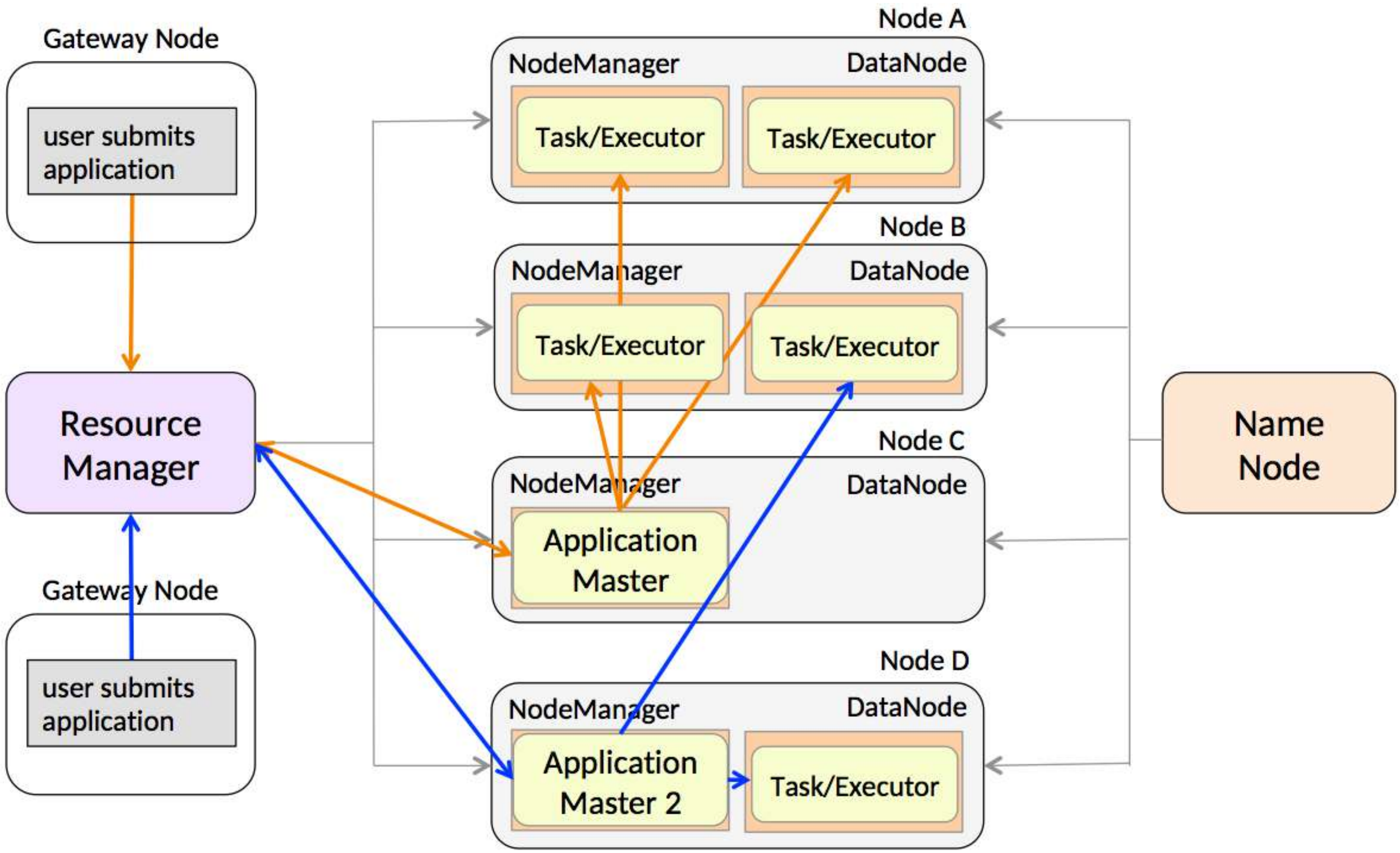
Running an Application on YARN (4)



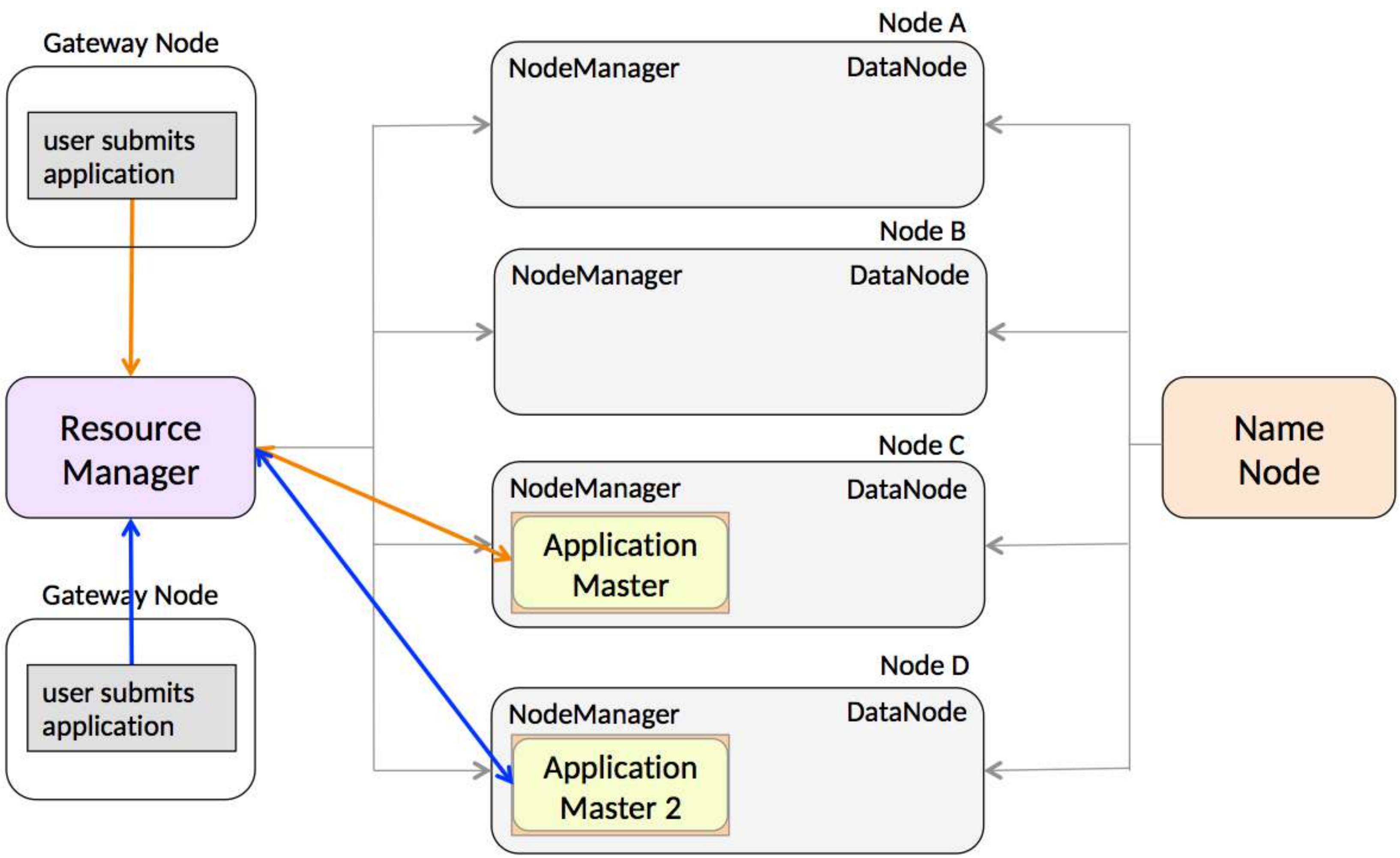
Running an Application on YARN (5)



Running an Application on YARN (6)



Running an Application on YARN (7)

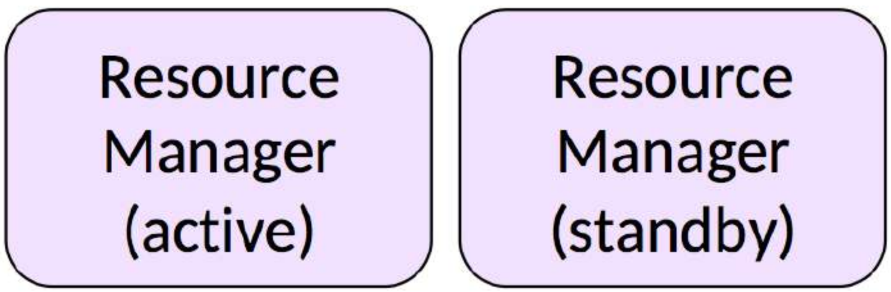


YARN Fault Tolerance (1)

Failure	Action Taken
ApplicationMaster stops sending heartbeats or YARN application fails	ResourceManager reattempts the whole application (default: 2 times)
Task exits with exceptions or stops responding	ApplicationMaster reattempts the task in a new container on a different host (default: 4 times)
Task fails too many times	Task aborted

YARN Fault Tolerance (2)

- **NodeManager**
  - If the NodeManager stops sending heartbeats to the ResourceManager
    - It is removed from list of active hosts
  - ApplicationMaster will treat tasks on the host as failed
  - If the ApplicationMaster host fails
    - Treated as a failed application
- **ResourceManager**
  - No applications or tasks can be launched while the ResourceManager is unavailable
  - Can be configured with high availability (HA)



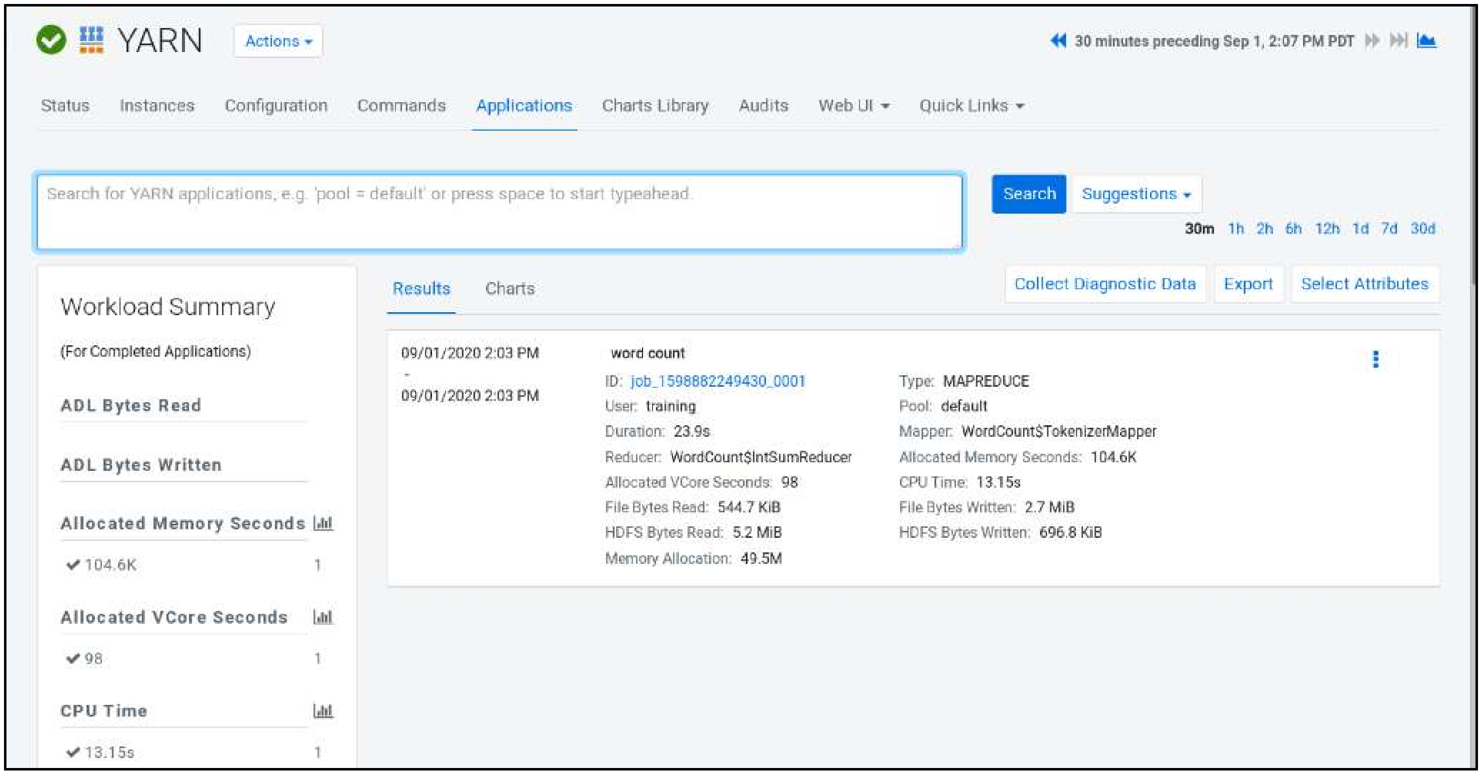
Chapter Topics

Data Compute

- YARN Overview
- Running Applications on YARN
- **Viewing YARN Applications**
- YARN Application Logs
- MapReduce Applications
- YARN Memory and CPU Settings
- Hands-On Exercise: Running YARN Applications
- Tez Overview
- ACID for Hive
- Spark Overview
- How Spark Applications Run on YARN
- Monitoring Spark Applications
- Essential Points
- Hands-On Exercise: Running Spark Applications

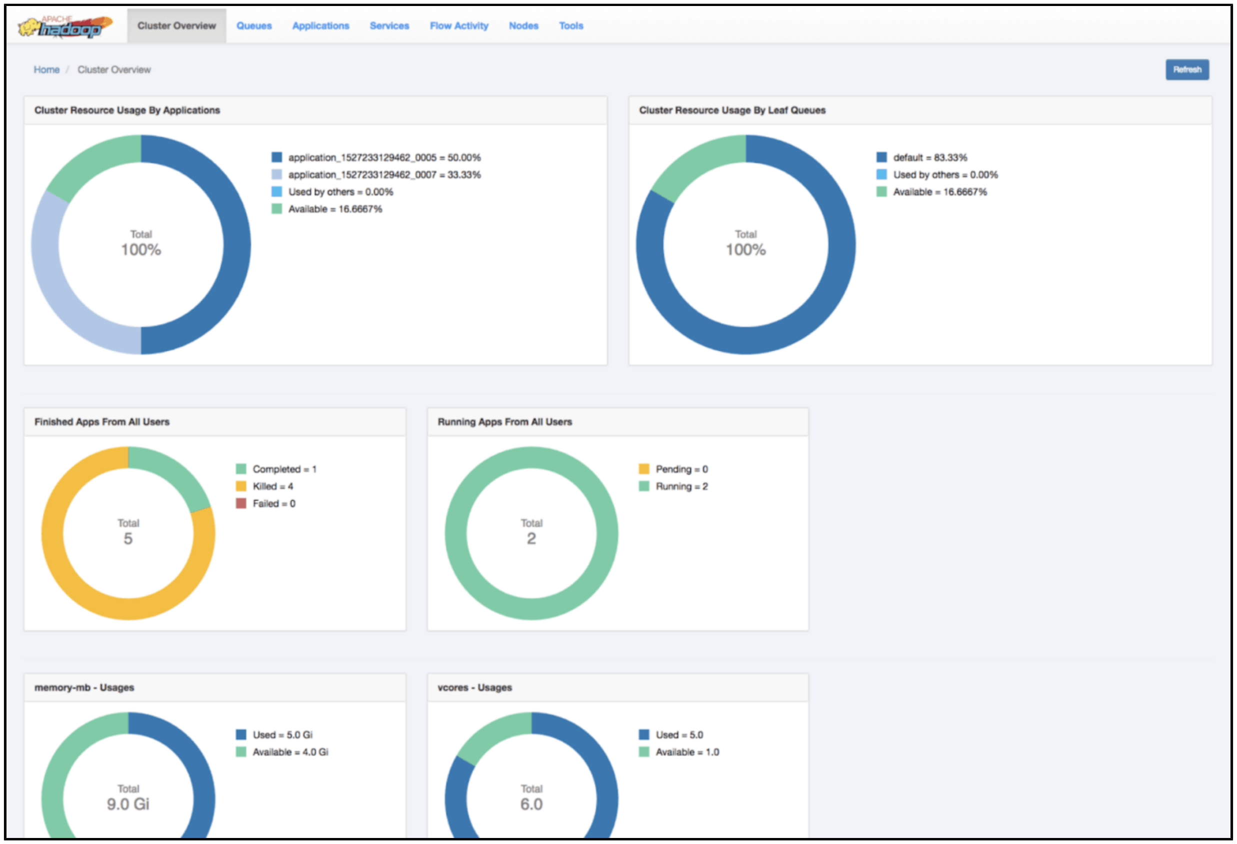
Cloudera Manager YARN Applications Page

- **Use the YARN Applications tab to view**
  - Details about individual running or completed jobs
  - Charts showing aggregated data by user, CPU usage, completion time, and so on



The ResourceManager Web UI (1)

- **ResourceManager UI**
  - Default URL `http://`*`yarn_rm_host`*`:8088`
    - Or use link from Cloudera Manager YARN service page



The ResourceManager Web UI (2)

- **The ResourceManager Web UI menu**
  - **Queues**
    - Displays details of YARN queues
  - **Applications**
    - List and filter applications
    - Link to Spark, Tez, or MapReduce ApplicationMaster UI (active jobs) or history server (completed jobs)
  - **Services**
    - Create new YARN services and view the list of existing services
  - **Nodes**
    - List NodeManagers and statuses
    - Link to details such as applications and containers on the node
  - **Tools**
    - View configuration, YARN logs, server details
    - Create standard services by providing their details or custom services by using JSON files containing definitions

MapReduce Job HistoryServer Web UI

- **The ResourceManager does not store details of completed jobs**
- **MapReduce HistoryServer web UI archives jobs metrics and metadata**
- **Role: JobHistoryServer (optional)**
- **Default URL: `http://`*`jhs_host`*`:19888`**
  - Or use links from Cloudera Manager YARN service page, ResourceManager UI, or the Hue Job Browser

YARN Command Line

- **Displaying YARN application details in the shell**
  - List all *running* applications

```
$ yarn application -list
```

    - Returns all running applications, including the application ID for each
  - List all applications, including completed applications

```
yarn application -list -appStates all
```

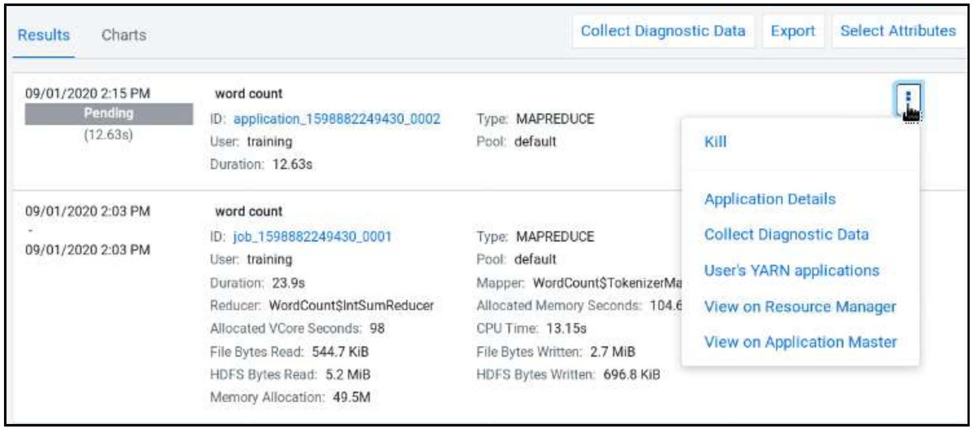
  - Display the status of an individual application
    - `yarn application -status` *`application_ID`*

YARN Application States and Types

- **Some logs and command results list YARN application states**
- **Operating states: `SUBMITTED`, `ACCEPTED`, `RUNNING`**
- **Initiating states: `NEW`, `NEW_SAVING`**
- **Completion states: `FINISHED`, `FAILED`, `KILLED`**
- **Some logs and command results list YARN application types: `MAPREDUCE`, `YARN`, `SPARK`, `TEZ`, `OTHER`**

Killing YARN Applications

- **To kill a running YARN application from Cloudera Manager**
  - Use the drop-down menu for the application in the YARN **Applications** tab



- **Use `yarn` to kill an application from the command line**
  - You *cannot* kill it using `Ctrl+C`
    - Only kills the client
    - The application is still running on the cluster

```
$ yarn application -kill application_ID
```

- **YARN applications can also be killed from the** Hue Job Browser

Chapter Topics

Data Compute

- YARN Overview
- Running Applications on YARN
- Viewing YARN Applications
- **YARN Application Logs**
- MapReduce Applications
- YARN Memory and CPU Settings
- Hands-On Exercise: Running YARN Applications
- Tez Overview
- ACID for Hive
- Spark Overview
- How Spark Applications Run on YARN
- Monitoring Spark Applications
- Essential Points
- Hands-On Exercise: Running Spark Applications

YARN Application Log Aggregation (1)

- **Application logs are distributed across the cluster**
  - Stored on NodeManager host local file systems
  - Makes debugging difficult
- **YARN aggregates logs**
  - Optional but recommended
    - Cloudera Manager enables by default
  - Logs are aggregated by application
  - Access the logs from Cloudera Manager, any HDFS client, or the YARN command line interface
- **Rolling log aggregation is now Supported**
  - Responsible for aggregating logs at set time intervals
  - Configurable by the user
  - Primarily used for long-running applications like Spark streaming jobs

YARN Application Log Aggregation (2)

- **For clusters with a large number of YARN aggregated logs:**
  - It can be helpful to combine them into Hadoop archives in order to reduce the number of small files
  - Hadoop Archives (HAR) can pack a number of small files into large files so that the original files can be accessed transparently (without expanding the files)
  - The stress on the NameNode reduced as well
  - Run the hadoop archive command by specifying the archive name to create, the parent directory relative to the archive location, the source files to archive, and the destination archive location

```
$ hadoop archive -archiveName name -p
```

YARN Application Log Aggregation (2)

- **Container log files moved to HDFS when application completes**
  - Default NodeManager local directory: `/yarn/container-logs`
    - While application is running
  - Default HDFS directory: `/tmp/logs/`*`user-name`*`/logs`
    - After application completes
- **Application log retention window**
  - Apache default: indefinitely
  - Cloudera Manager default: seven days

Accessing YARN Application Logs from the Command Line

- **Command line utility: `yarn`**
  - Combine with utilities such as `grep` for easier log analysis
- **Find the application ID with `yarn application -list` then show logs by application ID**

```
$ yarn application -list -appStates FINISHED | grep 'word count'
Total number of applications (application-types: [] and states: [FINISHED]):2
Application-Id                  Application-Name  Application-Type  User     Queue
application_1543841889521_0002  word count        MAPREDUCE
 training…
$ yarn logs --applicationId application_1543841889521_0002
Container container_1543841889521_0002_01_000007 on worker-1.example.com_8041
 LogType:stderr
Log Upload Time:Thu Mar 14 09:44:14 -0800 2018
…
LogLength:3191
Log Contents:
2019-03-14 09:44:03,863 INFO [main] org.apache.hadoop.metrics2.impl.MetricsConfig:…
2019-03-14 09:44:03,938 INFO [main] org.apache.hadoop.mapred.YarnChild: Executing…
```

Chapter Topics

Data Compute

- YARN Overview
- Running Applications on YARN
- Viewing YARN Applications
- YARN Application Logs
- **MapReduce Applications**
- YARN Memory and CPU Settings
- Hands-On Exercise: Running YARN Applications
- Tez Overview
- ACID for Hive
- Spark Overview
- How Spark Applications Run on YARN
- Monitoring Spark Applications
- Essential Points
- Hands-On Exercise: Running Spark Applications

What Is Apache Hadoop MapReduce?

- **Framework for developing distributed applications**
- **Hadoop MapReduce is part of the YARN service**
- **Based on the map-reduce programming paradigm**
  - Record-oriented data processing
- **Applications execute in two phases**
  - *Map*—input records are processed one at a time
    - Mapper output sorted and saved to local disk
  - *Reduce*—aggregates mapped data
    - Reducers read mapper output
- **Data is shuffled between NodeManager hosts running map and reduce tasks**
- **MapReduce applications are usually written in Java**

MapReduce Application Terminology

- **Job**
  - A mapper, a reducer, and a list of inputs to process
- **Task**
  - An individual unit of work
  - A job consists one or more tasks
    - Map task or reduce task
  - Each task runs in a container on a worker host
- **Client**
  - The program that submits the job to the ResourceManager

How MapReduce Applications Run on YARN

- **One container is created for *each task* in a job**
  - Each container has a JVM that runs its task
  - Containers are deleted after tasks completes
- **ApplicationMasters request containers on hosts close to input data blocks in HDFS**
  - This feature is called *data locality*
- **Scheduler assigns tasks to requested hosts when possible**
  - When resource availability permits
  - If host is not available, YARN will prefer the same rack
- **Start applications using the `hadoop jar` command**

```
$ hadoop jar application-file.jar class-name arguments…
```

Chapter Topics

Data Compute

- YARN Overview
- Running Applications on YARN
- Viewing YARN Applications
- YARN Application Logs
- MapReduce Applications
- **YARN Memory and CPU Settings**
- Hands-On Exercise: Running YARN Applications
- Tez Overview
- ACID for Hive
- Spark Overview
- How Spark Applications Run on YARN
- Monitoring Spark Applications
- Essential Points
- Hands-On Exercise: Running Spark Applications

YARN Resource Configuration

- **Set configuration properties for**
  - Worker host resources—how much memory and CPU cores are available on each host?
    - Cluster resources = sum of all worker resources
  - Container resource limits—how much memory and how many cores should each application be allowed?
    - ResourceManager Scheduler allocates resources for containers
    - Each container runs one application task
  - MapReduce applications—how much resources do applications request?
    - Scheduler provides containers subject to limits

Worker Host Resources

`yarn.nodemanager.resource.memory-mb`
Set in YARN / NodeManager Group / Resource Management

- Amount of RAM available on this host for YARN-managed tasks
- Recommendation: the amount of RAM on the host minus the amount needed for non-YARN-managed work (including memory needed by the DataNode daemon)
- Used by the NodeManagers

`yarn.nodemanager.resource.cpu-vcores`
Set in YARN / NodeManager Group / Resource Management

- Number of vcores available on this host for YARN-managed tasks
- Recommendation: the number of physical cores on the host minus 1
- Used by the NodeManagers

Per-Container Resources

`yarn.scheduler.minimum-allocation-mb`
`yarn.scheduler.maximum-allocation-mb`
`yarn.scheduler.minimum-allocation-vcores`
`yarn.scheduler.maximum-allocation-vcores`

Set in YARN / ResourceManager Group / Resource Management

- Minimum and maximum memory and vcores to allocate per container
- Cloudera Manager Defaults: Minimum=1GB/1 vcores, Maximum=NodeManager memory/vcores
- Minimum memory recommendation: increase up to 4GB if needed
- Minimum cores recommendation: keep the 1 vcore default
- Maximums should never exceed NodeManager resources
- Used by the ResourceManager

`yarn.scheduler.increment-allocation-mb`
`yarn.scheduler.increment-allocation-vcores`

Set in YARN / ResourceManager Group / Resource Management

- Requests are rounded up to nearest multiple of increment
- Cloudera Manager Defaults: 512MB and 1 vcore

MapReduce Applications: Memory Request for Containers

`mapreduce.map.memory.mb` and **`mapreduce.reduce.memory.mb`**
Set in YARN / Gateway Group / Resource Management

- Amount of memory to allocate for Map or Reduce tasks
- Cloudera Manager Default: 1GB
- Recommendation: Increase **`mapreduce.map.memory.mb`** up to 2GB, depending on your developers' requirements. Set **`mapreduce.reduce.memory.mb`** to twice the mapper value.
- Used by clients and NodeManagers

`yarn.app.mapreduce.am.resource.mb`
Set in YARN / Gateway Group / Resource Management

- Amount of memory to allocate for the ApplicationMaster
- Cloudera Manager Default: 1GB
- Recommendation: 1GB, but you can increase it if jobs contain many concurrent tasks
- Used by clients and NodeManagers

MapReduce Applications: Container JVM Heap Size

`yarn.app.mapreduce.am.command-opts`
Set in YARN / Gateway Group

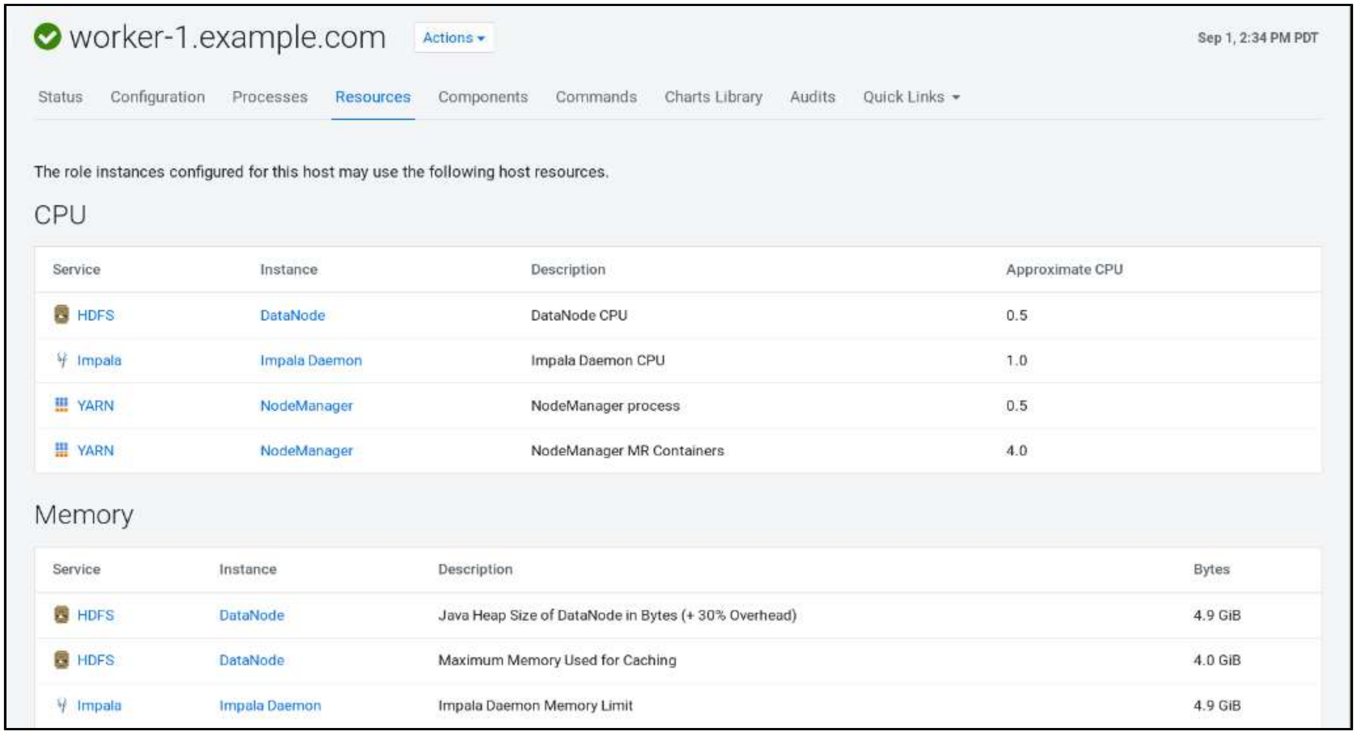
- Memory MapReduce applications request for ApplicationMaster containers
- Default is 1GB of heap space
- Used when ApplicationMasters are launched

`mapreduce.map.java.opts`
`mapreduce.reduce.java.opts`
Set in YARN / Gateway Group

- Java options passed to mappers and reducers
- Default is 85% of the max container allocation
- Recommendation: 1GB to 4GB, depending on the requirements from your developers
- Used when mappers and reducers are launched

Cluster Resource Utilization

- **Use Cloudera Manager to view resource usage on hosts**



YARN Tuning Recommendations

- **Inventory the vcores, memory, and disks available on each worker node**
- **Calculate the resources needed for other processes**
  - Reserve 4GB to 8GB of memory for the OS
  - Reserve resources for any non-Hadoop applications
  - Reserve resources for other any Hadoop components
    - HDFS caching (if configured), NodeManager, DataNode
    - Impalad, HBase RegionServer, Solr, and so on.
- **Grant the resources not used by the above to your YARN containers**
- **Configure the YARN scheduler and application framework settings**
  - Based on the worker node profile determined above
  - Determine the number of containers needed to best support YARN applications based on the type of workload
  - Monitor usage and tune estimated values to find optimal settings

The YARN Tuning Guide

- **Cloudera provides a useful guidance tool for YARN and MapReduce tuning**
  - Interactive MS Excel spreadsheet
  - `http://www.cloudera.com/content/www/en-us/documentation/other/shared/yarn-tuning-guide.xlsx`ƒ
- **Recommends settings based on worker host resources**

Chapter Topics

Data Compute

- YARN Overview
- Running Applications on YARN
- Viewing YARN Applications
- YARN Application Logs
- MapReduce Applications
- YARN Memory and CPU Settings
- **Hands-On Exercise: Running YARN Applications**
- Tez Overview
- ACID for Hive
- Spark Overview
- How Spark Applications Run on YARN
- Monitoring Spark Applications
- Essential Points
- Hands-On Exercise: Running Spark Applications

Hands-On Exercise: Running YARN Applications

- In this exercise, you will run a MapReduce job and examine the results in HDFS and in the YARN application logs
- Please refer to the Hands-On Exercise Manual for instructions

Chapter Topics

Data Compute

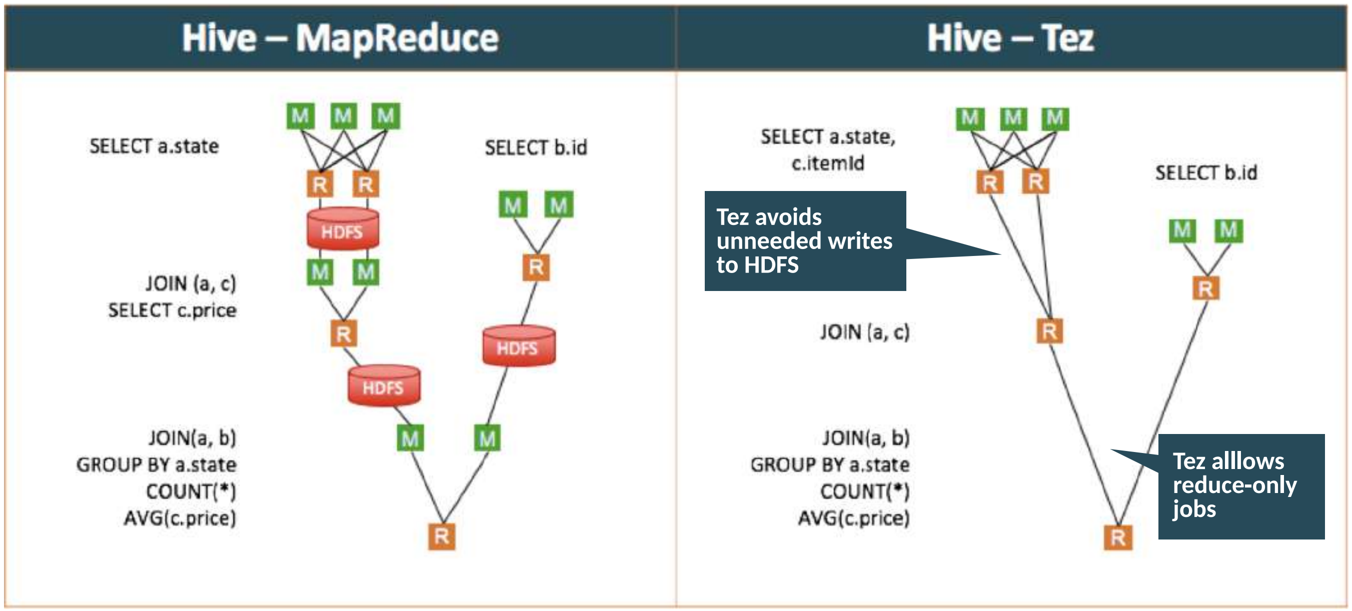
- YARN Overview
- Running Applications on YARN
- Viewing YARN Applications
- YARN Application Logs
- MapReduce Applications
- YARN Memory and CPU Settings
- Hands-On Exercise: Running YARN Applications
- **Tez Overview**
- ACID for Hive
- Spark Overview
- How Spark Applications Run on YARN
- Monitoring Spark Applications
- Essential Points
- Hands-On Exercise: Running Spark Applications

What is Tez?

- **A Framework for YARN-based, Data Processing Applications In Hadoop**
- **An extensible framework for building high performance batch and interactive applications**
- **Tez improves the MapReduce paradigm by dramatically improving its speed, while maintaining the ability to scale to petabytes of data**
- **Used by a growing number of third party data access applications developed for the Hadoop ecosystem**
- **In Cloudera Data Platform (CDP), Tez is usually used only by Hive**
- **Apache Tez provides a developer API and framework to write native YARN applications**
- **Hive embeds Tez so that it can translate complex SQL statements into highly optimized, purpose-built data processing graphs that strike the right balance between performance, throughput, and scalability**

Hive on Tez Architecture

```
SELECT a.state, COUNT(*), AVG(c.price)
FROM a
  JOIN b ON (a.id = b.id)
  JOIN c ON (a.itemid = c.itemid)
GROUP BY a.state
```



Hive on Tez

- Cloudera Runtime services include Hive on Tez and Hive Metastore
- In CDP, the Hive Metastore (HMS) is an independent service called Hive
- MapReduce and Spark are no longer supported for Hive
- Hive on Tez performs the HiveServer (HS2) role in a Cloudera cluster
- Install the Hive service and Hive on Tez in CDP. The HiveServer role is installed automatically during this process

Hive Upgrade Considerations to CDP

- **To assess the hardware and resource allocations for your cluster:**
  - CDP does not support Hive on Spark. Scripts that enable Hive on Spark do not work
  - Remove set hive.execution.engine=spark from your scripts
  - The CDP upgrade process tries to preserve your Hive configuration property overrides, however it does not perserve all overrides
  - If you upgrade from CDH and want to run an ETL job, set configuration properties to allow placement on the Yarn queue manager
  - Remove the hive user superuser group membership
  - Hive on Tez cannot run some queries on tables stored in encryption zones under certain conditions
  - CDP Private Cloud Base stores Hive data on HDFS by default, but CDP Public Cloud stores Hive data on S3 by default

Chapter Topics

Data Compute

- YARN Overview
- Running Applications on YARN
- Viewing YARN Applications
- YARN Application Logs
- MapReduce Applications
- YARN Memory and CPU Settings
- Hands-On Exercise: Running YARN Applications
- Tez Overview
- **ACID for Hive**
- Spark Overview
- How Spark Applications Run on YARN
- Monitoring Spark Applications
- Essential Points
- Hands-On Exercise: Running Spark Applications

ACID Operations

- **Stands for atomicity, consistency, isolation, and durability**
- **Perform ACID v2 transactions at the row level without configuration**
- **A Hive operation is atomic**
  - The operation succeeds completely or fails; it does not result in partial data
- **A Hive operation is also consistent**
  - Results are visible to the application in every subsequent operation
- **Hive operations are isolated**
  - Operations are isolated and do not cause unexpected side effects for others
- **Finally, a Hive operation is durable**
  - A completed operation is preserved in the event of a failure
- **Hive supports single-statement ACID transactions that can span multiple partitions**
- **A Hive client can read from a partition at the same time another client adds rows to the partition**

Administrator Duties for ACID

- **As Administrator, you can:**
  - Configure partitions for transactions
  - View transactions
  - View transaction locks

View Transactions

- **As Administrator, you can view a list of open and aborted transactions**
- **Enter a query to view transactions: `SHOW TRANSACTIONS`**
- **The following information appears in the output:**
  - Transaction ID
  - Transaction state
  - Hive user who initiated the transaction
  - Host machine or virtual machine where transaction was initiated

View Transaction Locks

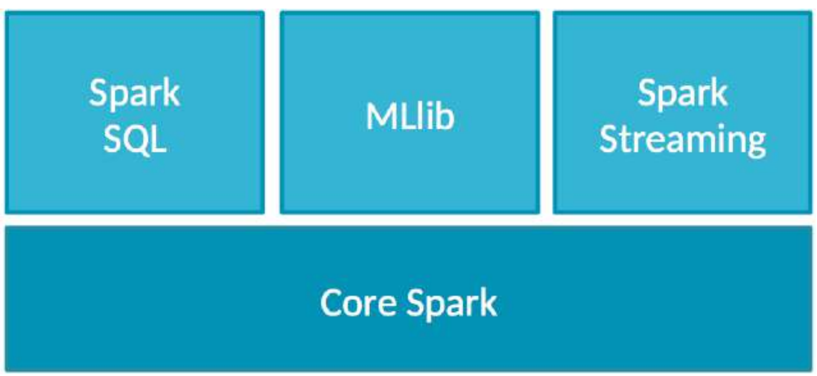
- **Hive transactions, enabled by default, disables Zookeeper locking**
- **DbLockManager stores and manages all transaction lock information in the Hive Metastore**
- **Heartbeats are sent regularly from lock holders to prevent stale locks and transactions**
- **To view transaction locks:**
  - Before you begin, check that transactions are enabled (the default)
  - Enter a Hive query to check table locks:
    - `SHOW LOCKS mytable EXTENDED;`
  - Check partition locks:
    - `SHOW LOCKS mytable PARTITION(ds='2018-05-01', hr='12') EXTENDED;`
  - Check schema locks:
    - `SHOW LOCKS SCHEMA mydatabase;`

Chapter Topics

Data Compute

What is Apache Spark?

- **Apache Spark is a fast, distributed, general-purpose engine for large-scale data processing**
  - The Spark application framework provides analytic, batch, and streaming capabilities
- **Spark provides a stack of libraries built on core Spark**
  - Core Spark provides the fundamental Spark abstraction: Resilient Distributed Datasets (RDDs)
  - Spark SQL works with structured data
  - MLlib supports scalable machine learning
  - Spark Streaming applications process data in real time



Spark Key Concepts

- **RDDs are the fundamental unit of data in core Spark**
  - Provide the low-level foundation of all Spark processes
  - Very flexible but complicated to work with directly
- **DataFrames and Datasets**
  - The primary mechanisms for working with data in Spark applications
    - Higher level than RDDs
  - Represent data as a table
  - Data can be queried using SQL-like operations such as `select`, `where`, `join`, and `groupBy` aggregation
- **CDP Supports Apache Spark:**
  - Apache Livy for local and remote access to Spark through the Livy REST API
  - Apache Zeppelin for browser-based notebook access to Spark
  - Spark LLAP connector is not supported
- **Spark 3 is available as a separate parcel**

Installing Spark

- **Spark roles**
  - **Spark Gateway**—installs configuration files and libraries required for Spark applications
    - Typically assigned to gateway hosts
  - **Spark History Server**—runs HistoryServer daemon to track completed Spark applications
    - Typically assigned to a master host

Chapter Topics

Data Compute

- YARN Overview
- Running Applications on YARN
- Viewing YARN Applications
- YARN Application Logs
- MapReduce Applications
- YARN Memory and CPU Settings
- Hands-On Exercise: Running YARN Applications
- Tez Overview
- ACID for Hive
- Spark Overview
- **How Spark Applications Run on YARN**
- Monitoring Spark Applications
- Essential Points
- Hands-On Exercise: Running Spark Applications

Spark Application Components

- **A Spark application consists of**
  - A client
    - Submits the application to run on the cluster
  - A driver
    - Manages applications tasks
  - Multiple executors
    - JVMs that run Spark tasks

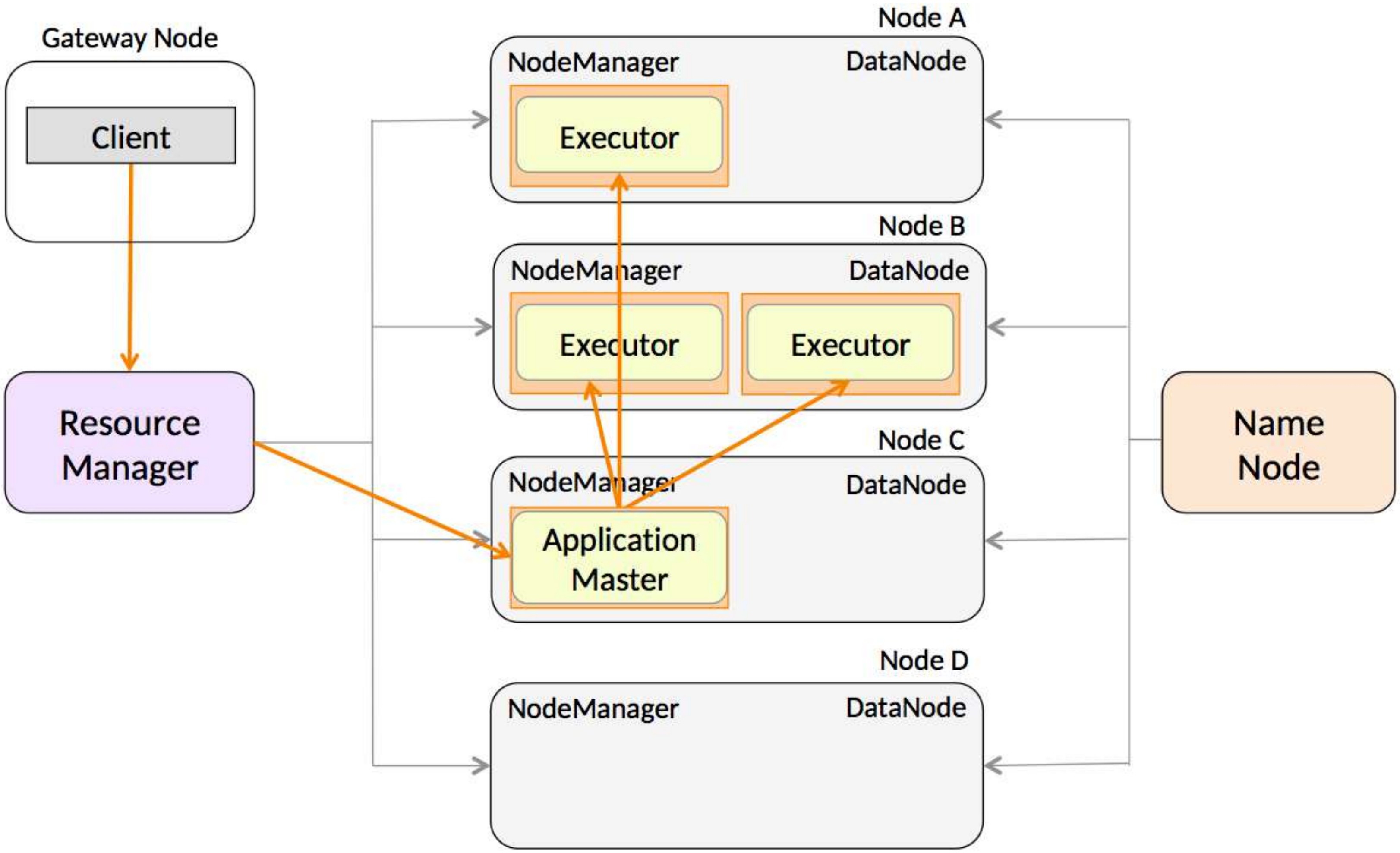
Types of Spark Applications

- **Spark shell**
  - Interactive Spark application for learning and exploring data
  - Start with `pyspark` (Python) or `spark-shell` (Scala)
- **Spark standalone applications**
  - Runs as a batch operation on the cluster
  - Python, Scala, or Java
  - Start with `spark-submit`

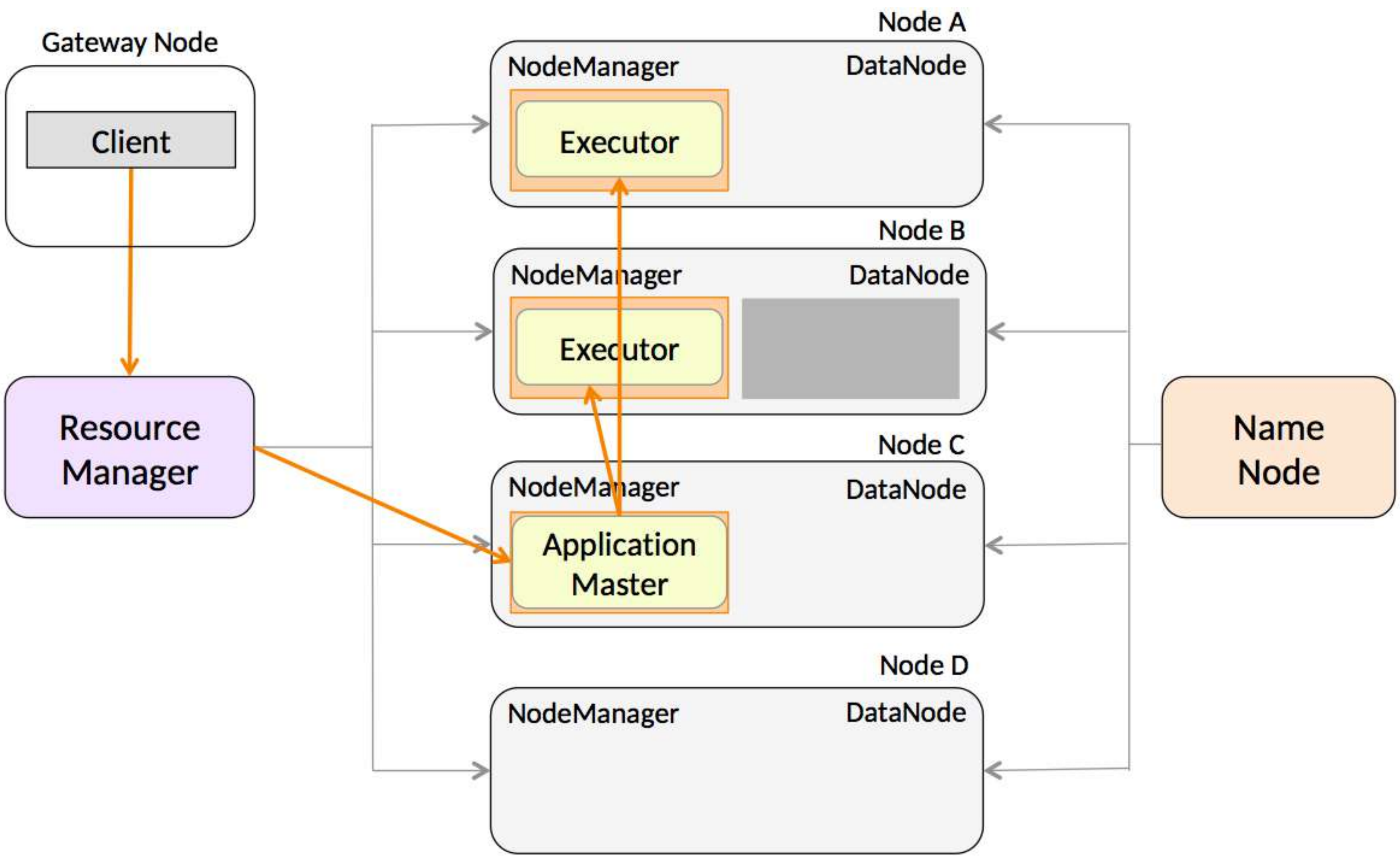
Spark Applications on YARN

- **Clients run in a JVM on a gateway host**
- **Executors run in JVMs in containers on worker hosts**
- **Each application has an Application Master**
  - Runs in a container JVM on the cluster

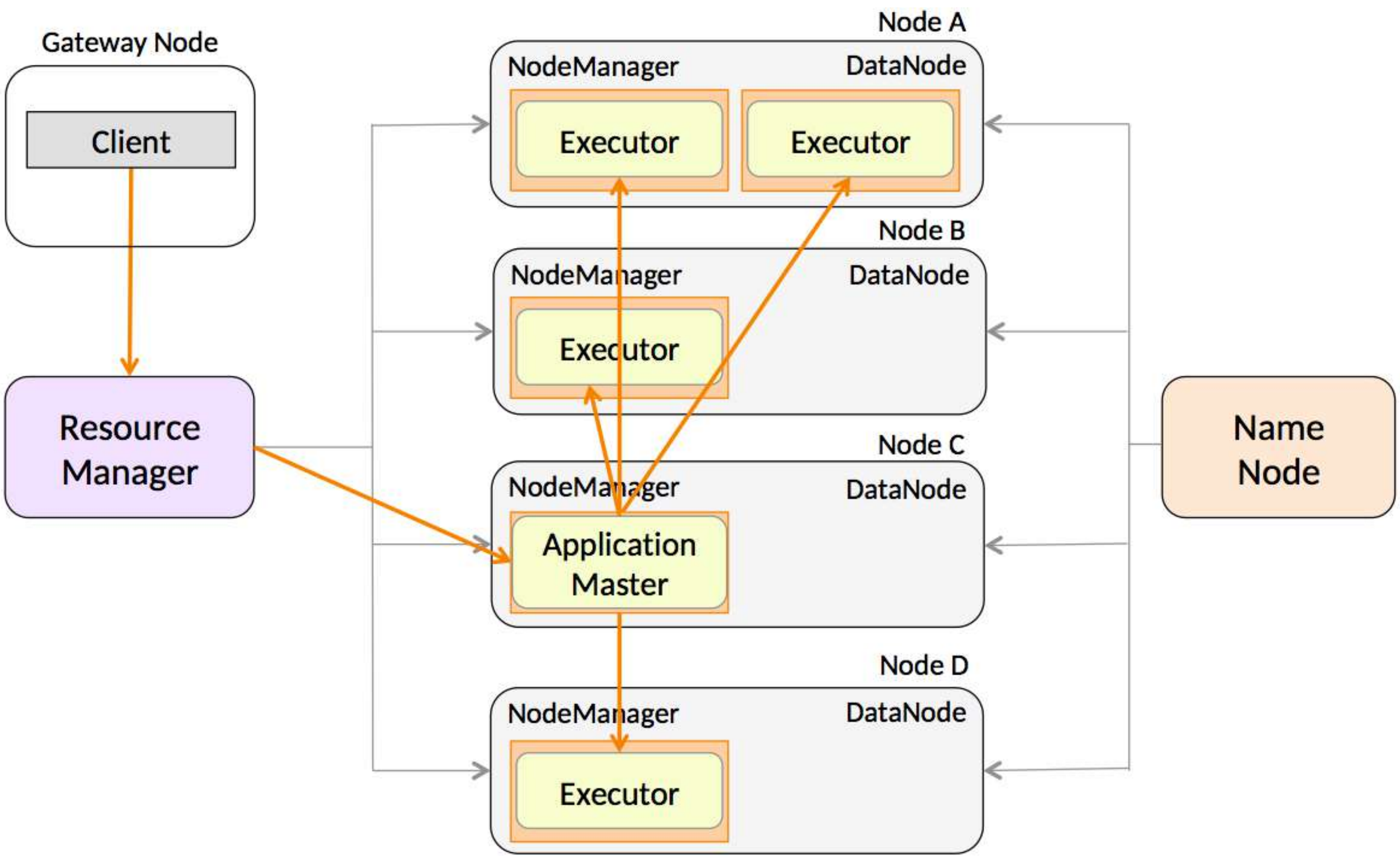
How Spark Applications Run on YARN (1)



How Spark Applications Run on YARN (2)



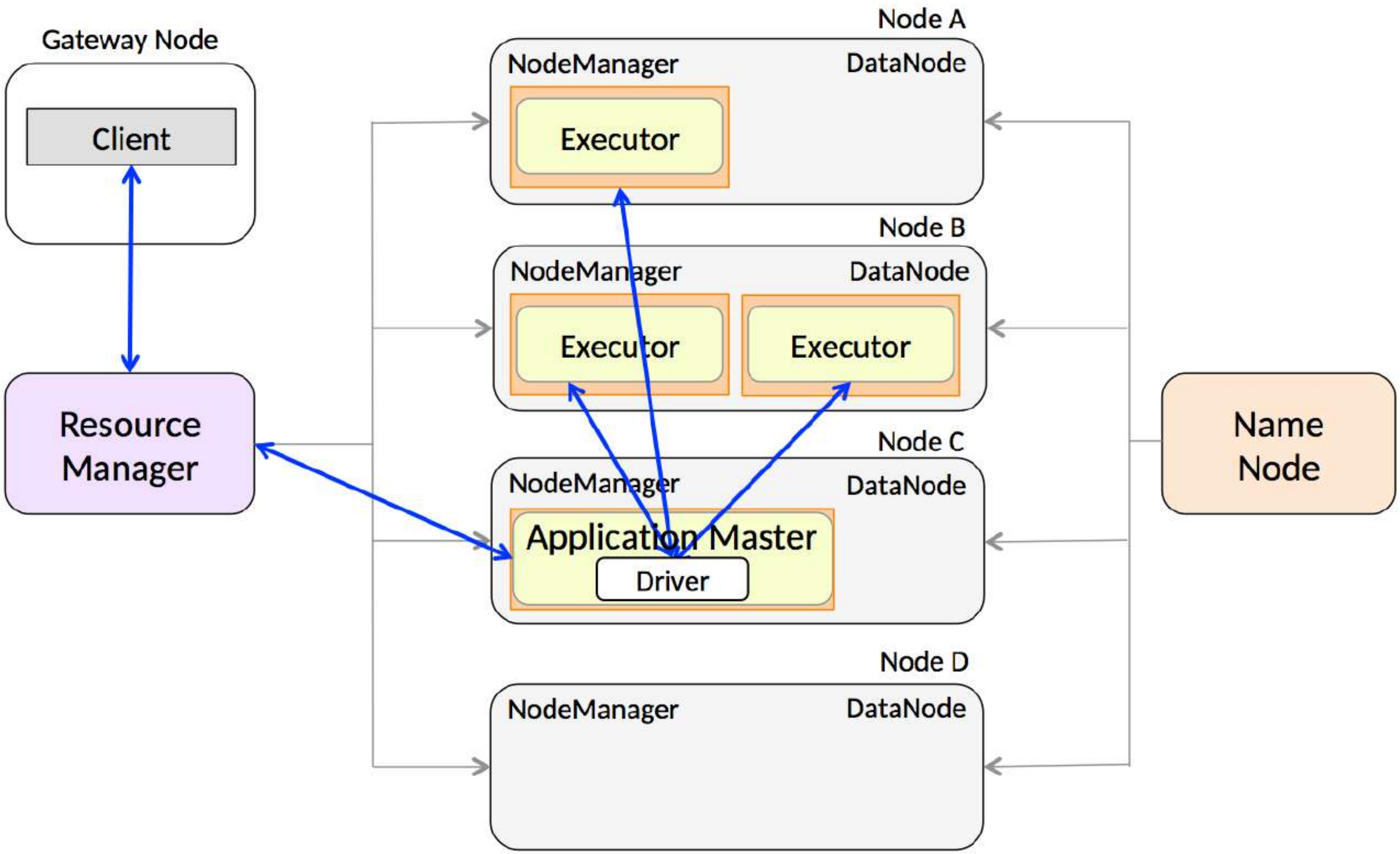
How Spark Applications Run on YARN (3)



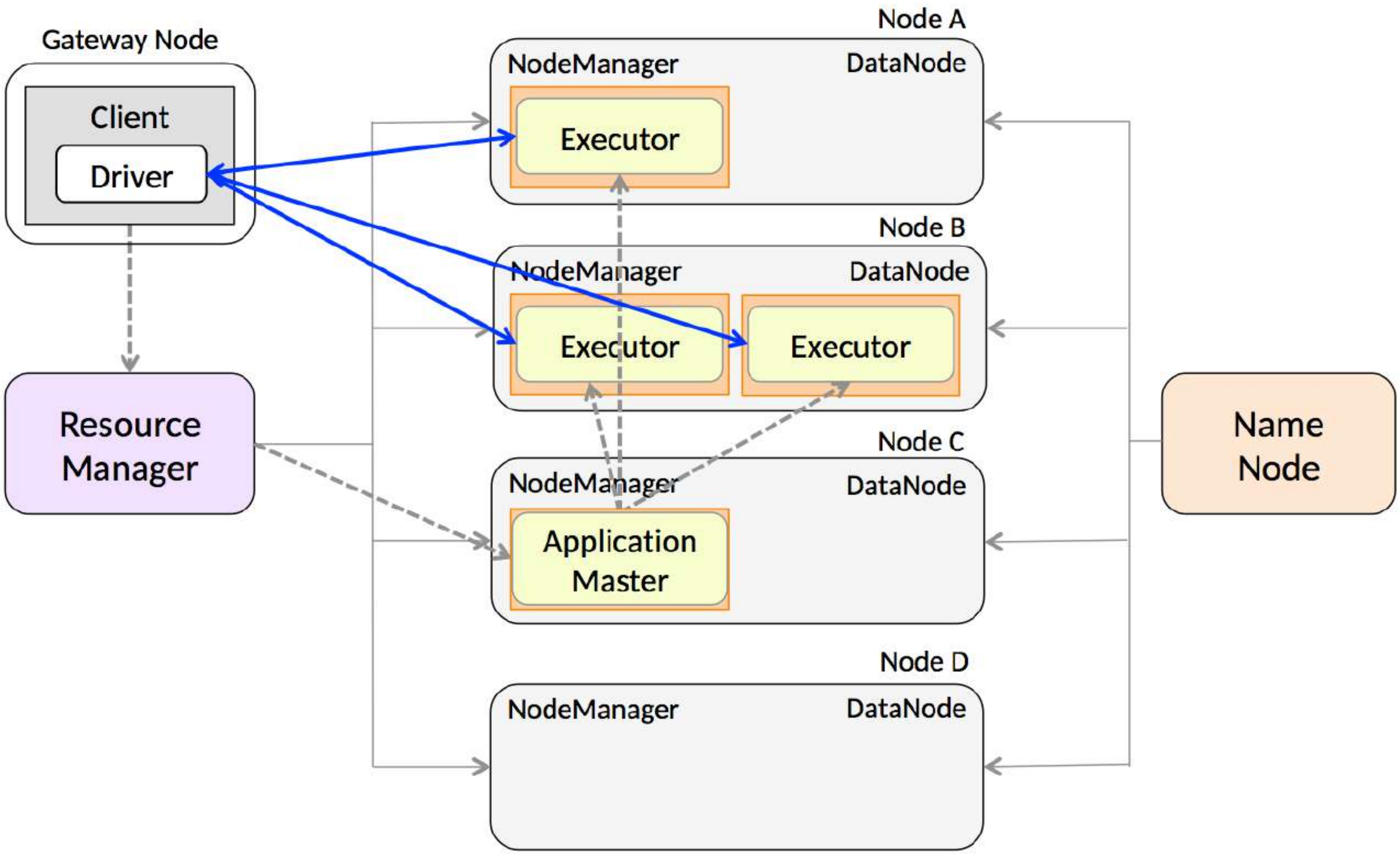
Application Deployment Modes

- **Cluster Deployment Mode**
  - Most common mode for standalone applications
  - Spark driver runs in the ApplicationMaster on the cluster
    - Minimal CPU, memory, or bandwidth requirements for client host
  - If the client disconnects the application will continue to run
  - More secure—all communication happens between nodes within the cluster
- **Client Deployment Mode (default)**
  - Spark driver runs in the client
    - Client node may require additional CPU, memory, or bandwidth
  - Required to run Spark Shell
  - Less secure—driver needs to communicate directly with cluster worker hosts
- **Configure `spark_deploy_mode` to change default**

Cluster Deployment Mode



Client Deployment Mode

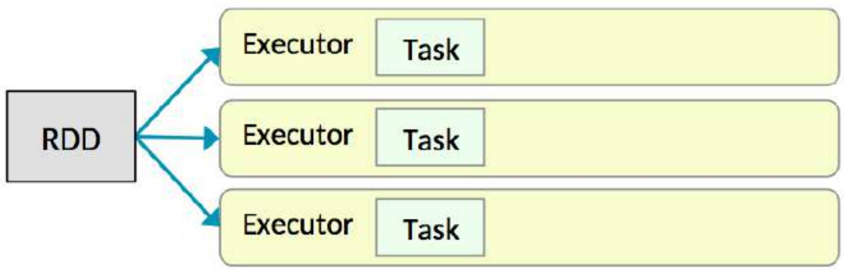


Spark Distributed Processing (1)

- **Spark partitions the data in an RDD, DataFrame, or Dataset**
- **Partitions are copied to different executors**

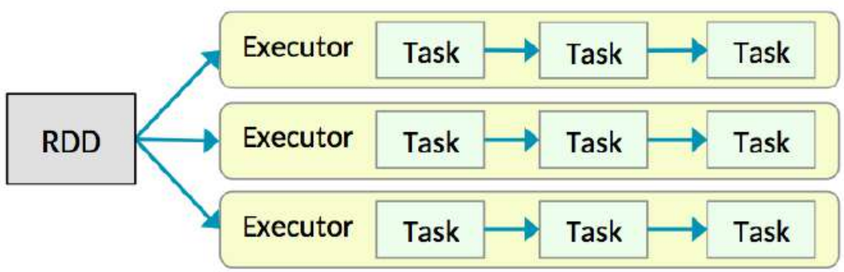
Spark Distributed Processing (2)

- **Executors run tasks that operate on the data partition distributed to that executor**
- **Multiple tasks on the same RDD, DataFrame, or Dataset can run in parallel on cluster**



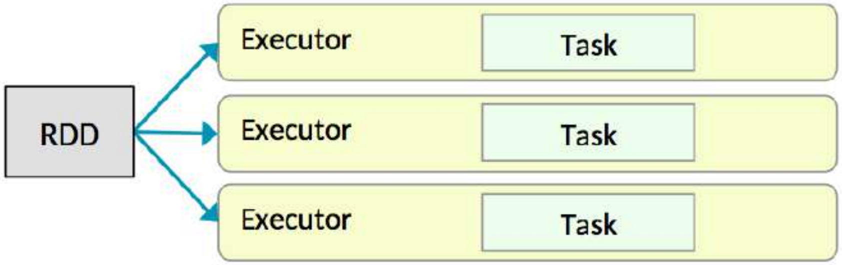
Spark Distributed Processing (3)

- **A series of tasks that all operate on the same data partition run within the same executor**



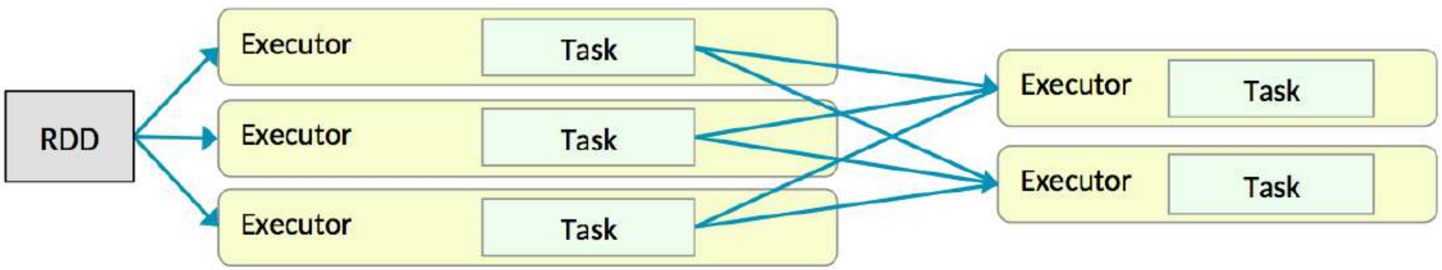
Spark Distributed Processing (4)

- **Spark consolidates tasks that operate on the same partition into a single task**
  - This is called *pipelining*



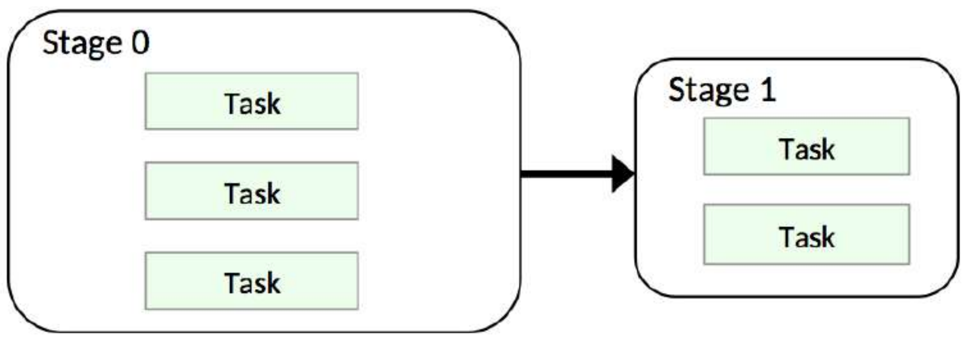
Spark Distributed Processing (5)

- **Some operations work on multiple partitions, such as joins and aggregations**
- **Data partitions are *shuffled* between executors**



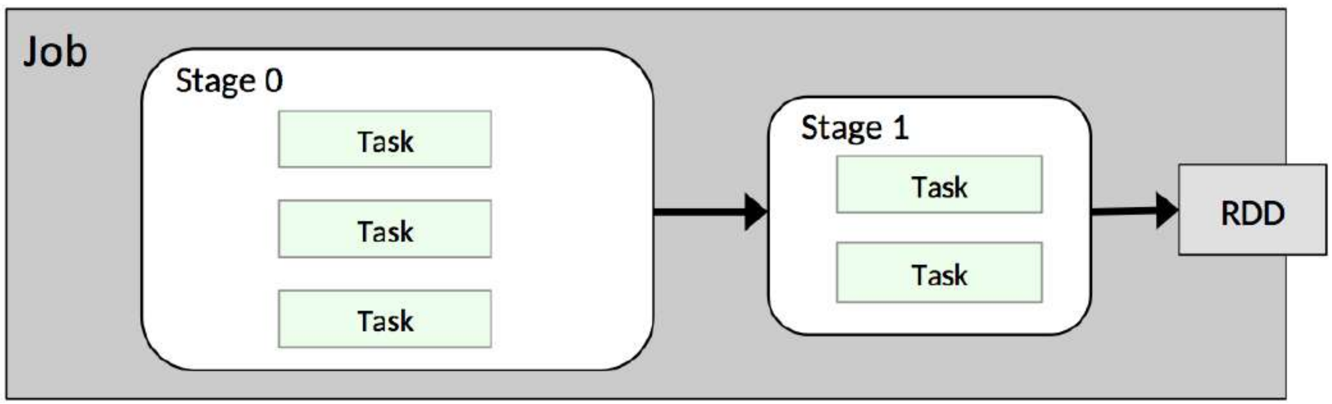
Spark Distributed Processing (6)

- **Tasks that operate in parallel on the same dataset are called a *stage***
- **Dependent stages must wait until prior stages complete**



Spark Distributed Processing (7)

- **Stages that work together to produce a single output dataset are called a *job***
- **One application can run any number of jobs**



Chapter Topics

Data Compute

Spark Web UIs

- **Running applications: Application UI**
  - Served by driver through ResourceManager proxy
- **Completed applications: History Server UI**
  - Default URL: `http://history-server:18089`
- **Access UIs through CM Spark service page or YARN ResourceManager UI**
- **Spark History Server log directory**
  - Default: `/var/log/spark`

Chapter Topics

Data Compute

Essential Points (1)

- **YARN manages resources and applications on Hadoop clusters**
  - YARN ResourceManager schedules resources and manages the application lifecycle
  - YARN NodeManagers launch containers for application tasks
- **YARN can run applications based on computational frameworks such as MapReduce and Spark**
- **YARN applications consist of an ApplicationMaster and one or more containers on NodeManager worker hosts**
  - The ApplicationMaster manages individual tasks running in container JVMs
- **Monitor YARN applications using**
  - Cloudera Manager's YARN **Applications** tab
  - ResourceManager Web UI
  - The (MapReduce) Job History Server Web UI
- **YARN aggregates logs from multiple application tasks**

Essential Points (2)

- **Tez is an extensible framework that runs under YARN**
- **In Cloudera Data Platform, Tez is usually used only by Hive**
- **Hive embeds Tez so that it can translate complex SQL statements**
- **You can perform ACID v2 transactions at the row level without any configuration**
- **Apache Spark is a fast, distributed, general-purpose engine for large-scale data processing**
- **Spark provides a stack of libraries built on core Spark**
- **To Monitor Spark applications:**
  - Running applications: Application UI
  - Completed applications: History Server UI
  - Access UIs through CM Spark 2 service page or YARN ResourceManager UI

Chapter Topics

Data Compute

- YARN Overview
- Running Applications on YARN
- Viewing YARN Applications
- YARN Application Logs
- MapReduce Applications
- YARN Memory and CPU Settings
- Hands-On Exercise: Running YARN Applications
- Tez Overview
- ACID for Hive
- Spark Overview
- How Spark Applications Run on YARN
- Monitoring Spark Applications
- Essential Points
- **Hands-On Exercise: Running Spark Applications**

Hands-On Exercise: Running Spark Applications

- **In this exercise, you will run a Spark job and examine the results in HDFS and in the YARN application logs**
- **Please refer to the Hands-On Exercise Manual for instructions**

Managing Resources

Chapter 10

Course Chapters

- Introduction
- Cloudera Data Platform
- CDP Private Cloud Base Installation
- Cluster Configuration
- Data Storage
- Data Ingest
- Data Flow
- Data Access and Discovery
- Data Compute
- **Managing Resources**
- Planning Your Cluster
- Advanced Cluster Configuration
- Cluster Maintenance
- Cluster Monitoring
- Cluster Troubleshooting
- Security
- Private Cloud / Public Cloud
- Conclusion
- Appendix: Cloudera Manager API
- Appendix: Ozone Overview

Managing Resources

After completing this chapter, you will be able to

- **Summarize the purpose and operation of the Capacity Scheduler**
- **Configure and manage YARN queues**
- **Control access to YARN queues**
- **Queue Manager**
- **Impala Query Scheduling**

Chapter Topics

Managing Resources

- **Managing Resources Overview**
- Node Labels
- Configuring cgroups
- The Capacity Scheduler
- Managing Queues
- Impala Query Scheduling
- Essential Points
- Hands-On Exercise: Using The Capacity Scheduler

Managing Resources (1)

- **Hadoop applications compete for cluster resources**
- **Objectives of cluster resource management as mentioned in previous chapter**
  - Guarantee the completion of critical workloads in reasonable timeframe
  - Coordinate cluster resource usage between competing groups of users
  - Prevent users from depriving other users access to cluster resources

Managing Resources (2)

- **You can manage resources for the applications**
  - Partitioning the cluster into subclusters using node labels
    - So that jobs run on nodes with specific characteristics
    - Use Node labels to run YARN applications on nodes with specified node label
  - Limiting CPU Usage through Linux Control Groups (cgroups)
    - Enable CPU Scheduling to enable cgroups to limit CPU usage
    - If you are not using CPU Scheduling, do not enable cgroups
  - Allocating resources through scheduling
    - Allocate CPU, and memory among users and groups
    - Configure using *dynamic resource pools* and other settings

Chapter Topics

Managing Resources

- Managing Resources Overview
- **Node Labels**
- Configuring cgroups
- The Capacity Scheduler
- Managing Queues
- Impala Query Scheduling
- Essential Points
- Hands-On Exercise: Using The Capacity Scheduler

Node Labels

- **You can use Node labels to run YARN applications on cluster nodes that have one of the following node labels:**
  - *exclusive*: Access is restricted to applications running in queues associated with the node label
  - *sharable*: If idle capacity is available on the labeled node, resources are shared with all applications in the cluster
- **You can use Node labels to partition a cluster into sub-clusters so that jobs run on nodes with specific characteristics**
- **For example, you can use node labels to run memory-intensive jobs only on nodes with a larger amount of RAM**

Node Label Configuration

- **To configure node labels:**
  - Make configuration changes for the YARN ResourceManager by adding directories and setting an advanced configuration snippet
  - To add the node labels "x" as exclusive, and "y" as shareable (non-exclusive) use the commands:

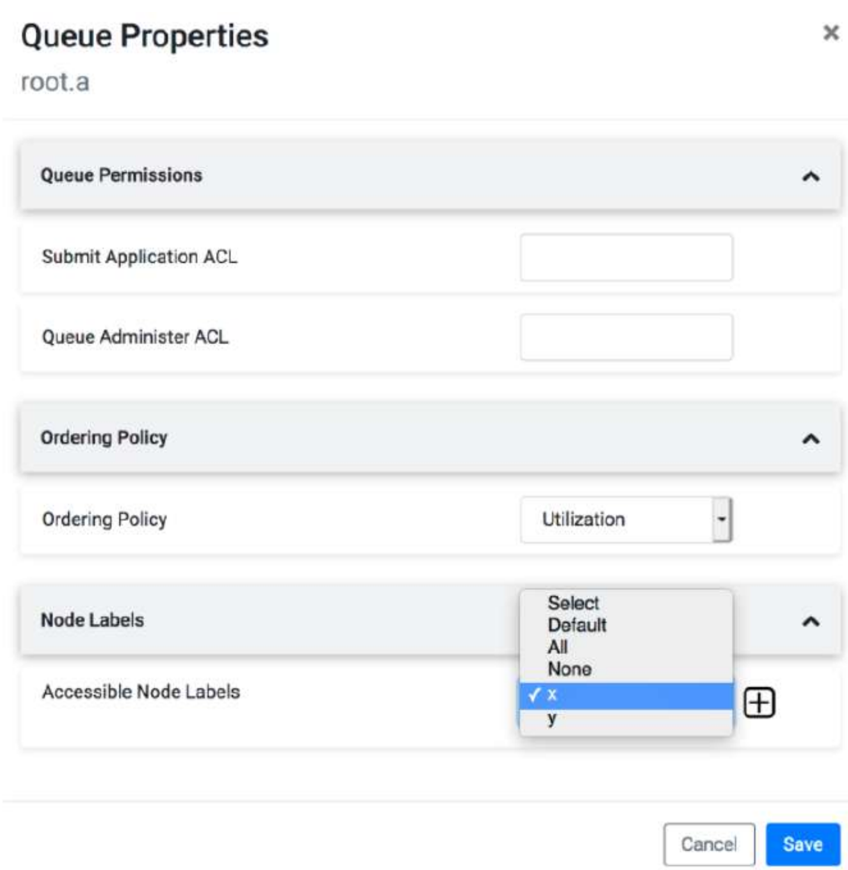
```
$ yarn rmadmin -addToClusterNodeLabels \
 "x(exclusive=true),y(exclusive=false)";
```

- **To list all node labels in the cluster:**

```
$ yarn cluster --list-node-labels
```

Node Label Usage

- **Node lables can be designated when defining a YARN queue**



Chapter Topics

Managing Resources

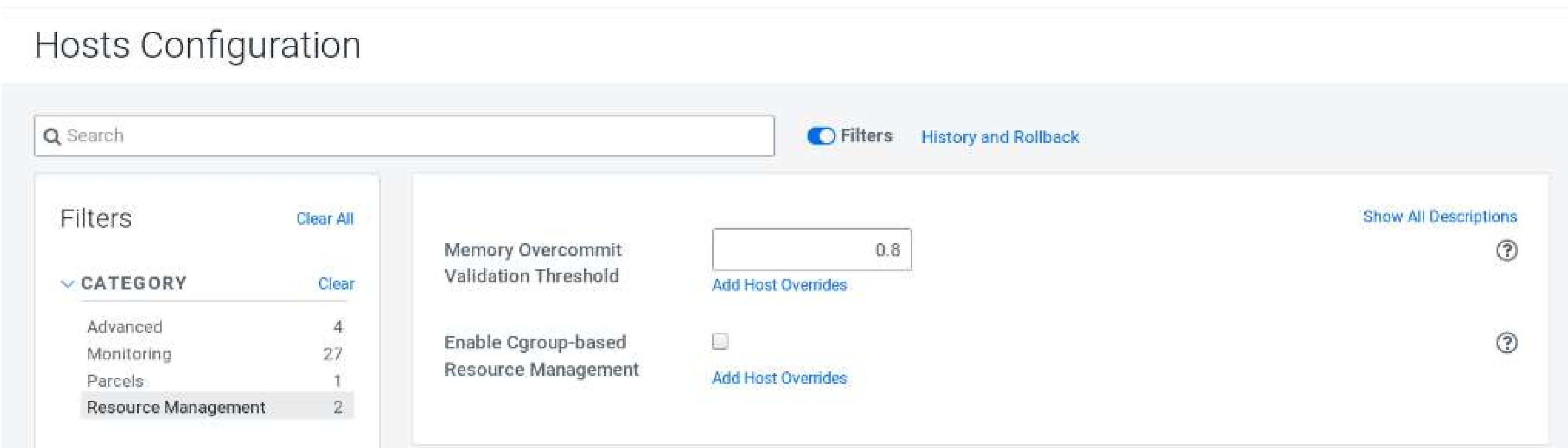
- Managing Resources Overview
- Node Labels
- **Configuring cgroups**
- The Capacity Scheduler
- Managing Queues
- Impala Query Scheduling
- Essential Points
- Hands-On Exercise: Using The Capacity Scheduler

Linux Control Groups (cgroups)

- **Linux control groups (cgroups) are a kernel feature for setting restrictions on Linux processes**
  - Useful for isolating computation frameworks from one another
- **RHEL and CentOS 6+ support cgroups**
  - See the Cloudera documentation for cgroup support on other Linux distributions
- **Configure cgroups to ensure that one service cannot overuse cluster CPU resources**
  - cgroups are not enabled by default on CDP
  - cgroups require that the CDP cluster be Kerberos enabled
- **Use cgroups when YARN and non-YARN services will share cluster resources**
  - Example
    - MapReduce and/or Spark running on YARN
    - Impala, HBase, HDFS, or Search services on the same cluster

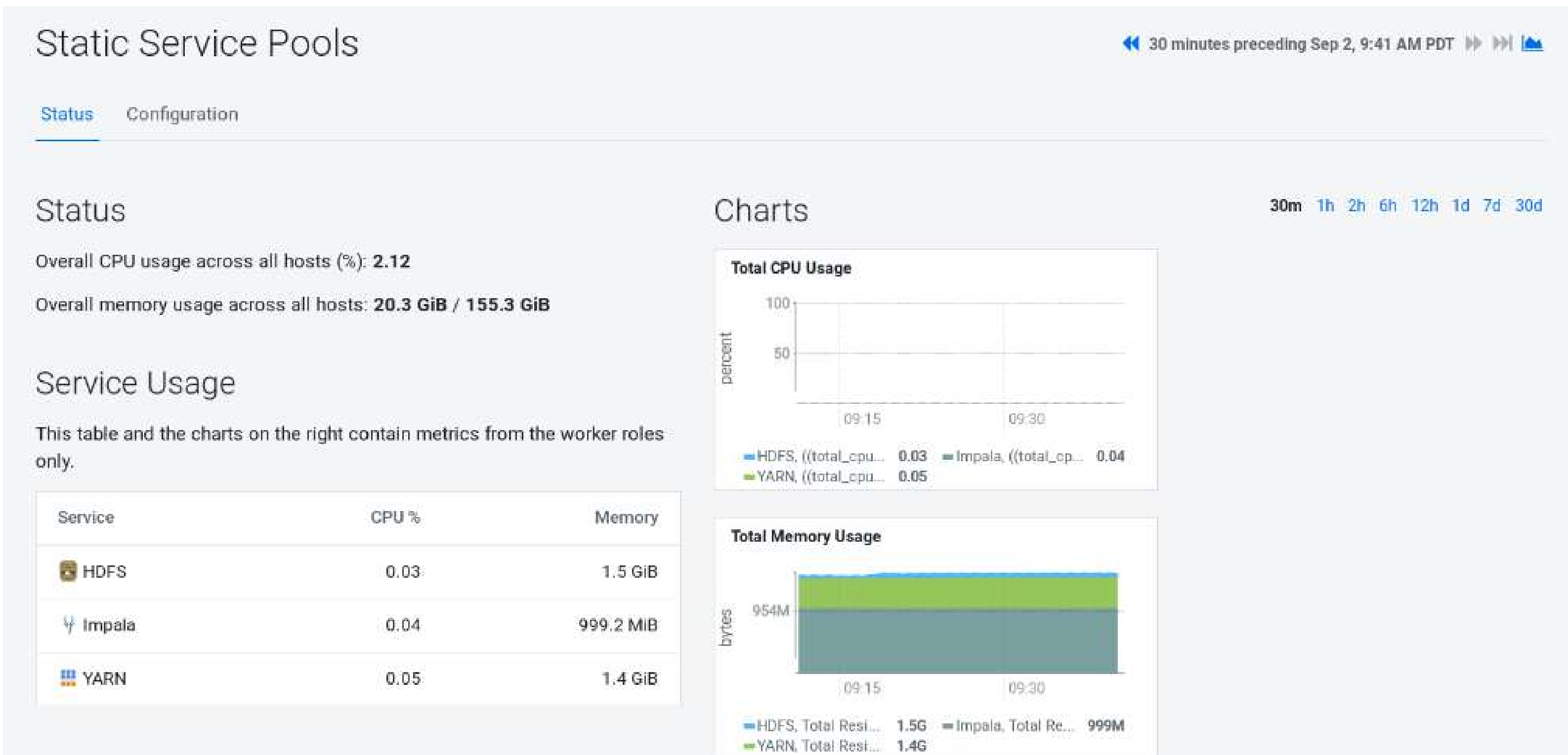
Enabling Resource Management with Cgroups

- **To enable Linux Control Groups (cgroups) using Cloudera Manager:**
  - Click Hosts, Host Configuration
  - Click Category, Resource Management
  - Select the Enable Cgroup-based Resource Management parameter
  - To enable Cgroups only on specific hosts, click the Add Host Overrides link
  - Restart all roles on the host(s)



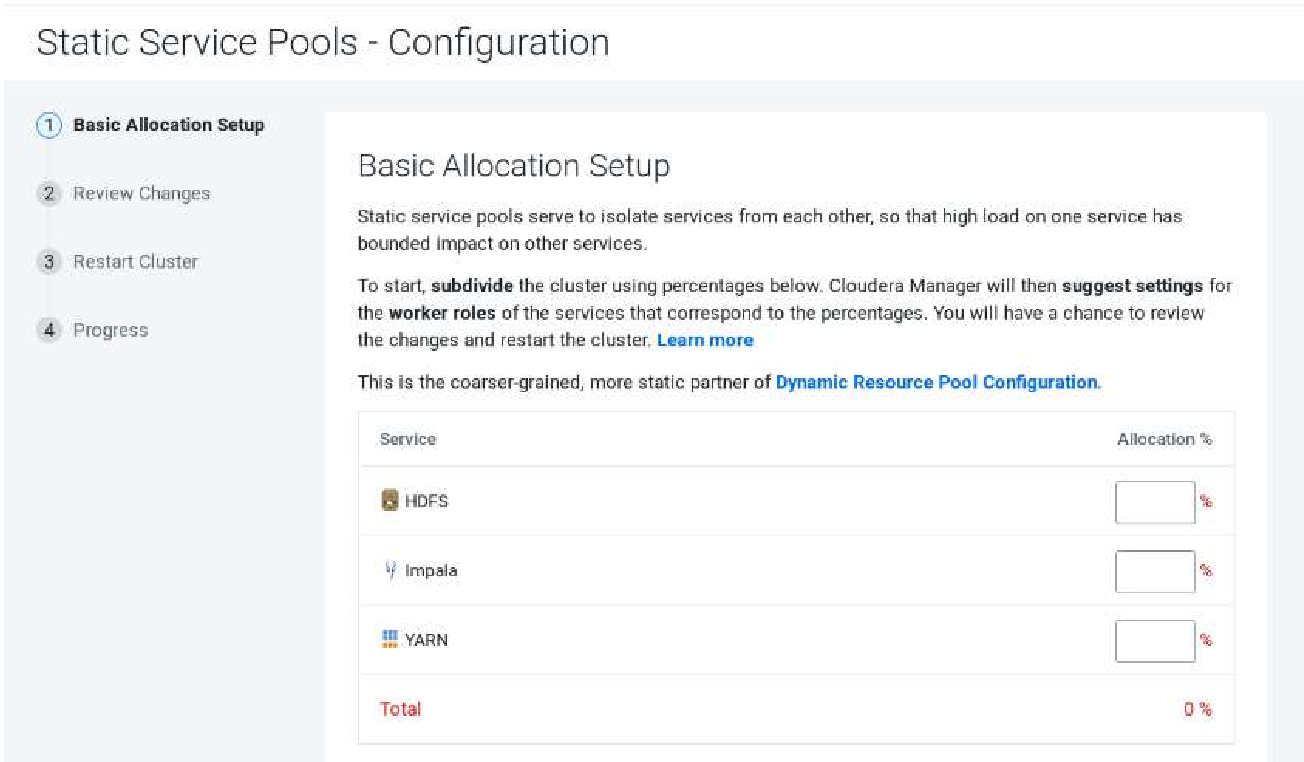
Static Service Pools - Status

- **Static service pools isolate the services in your cluster from one another, so that load on one service has a bounded impact on other services**
- **Select Clusters, Cluster name, Static Service Pools**



Static Service Pools - Configuration

- **Select the Configuration tab**



Configure CPU Scheduling

- **You can configure CPU scheduling on your cluster to allocate the best possible nodes having the required CPU resources for application containers**
  - YARN service, configuration tab
  - Set Resource Calculator Class to the org.apache.hadoop.yarn.util.resource.DominantResourceCalculator option
  - Set yarn.nodemanager.resource.cpu-vcores to the number of vcores to match the number of physical CPU cores on the NodeManager
- **Enable cgroups along with CPU scheduling to activate strict enforcement**
- **With cgroups strict enforcement activated, each CPU process receives only the resources it requests**
- **Without cgroups activated, the DRF scheduler attempts to balance the load, but unpredictable behavior may occur**

Chapter Topics

Managing Resources

- Managing Resources Overview
- Node Labels
- Configuring cgroups
- **The Capacity Scheduler**
- Managing Queues
- Impala Query Scheduling
- Essential Points
- Hands-On Exercise: Using The Capacity Scheduler

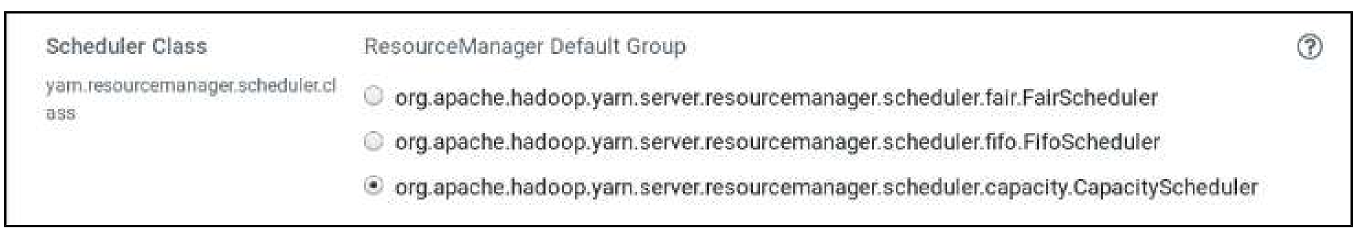
The YARN Scheduler

- **The YARN ResourceManager's scheduling component assigns resources to YARN applications**
  - The scheduler decides *where* and *when* containers will be allocated to applications
  - Based on requirements of each application
  - Containers get specific resources
    - Memory, CPU
- **Administrators define a scheduling policy that best fits requirements**
  - For example, a policy that gives priority to SLA-driven * applications
- **The scheduling policy establishes rules for resource sharing**
  - Rules are well-defined
  - Form the basis for application start and completion expectations

* Service Level Agreement

The Capacity Scheduler

- **Manage your cluster capacity using the Capacity Scheduler in YARN**
- **To allocate available resources**
  - If you have only one type of resource use the `DefaultResourceCalculator`
  - If you have multiple resource types, use the `DominantResourceCalculator` (default)
- **To enable the Capacity Scheduler**
  - In Cloudera Manager, go to the Configuration page for the YARN service
  - Set the Scheduler Class property to org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.CapacityScheduler (default)



Capacity Scheduler Queues

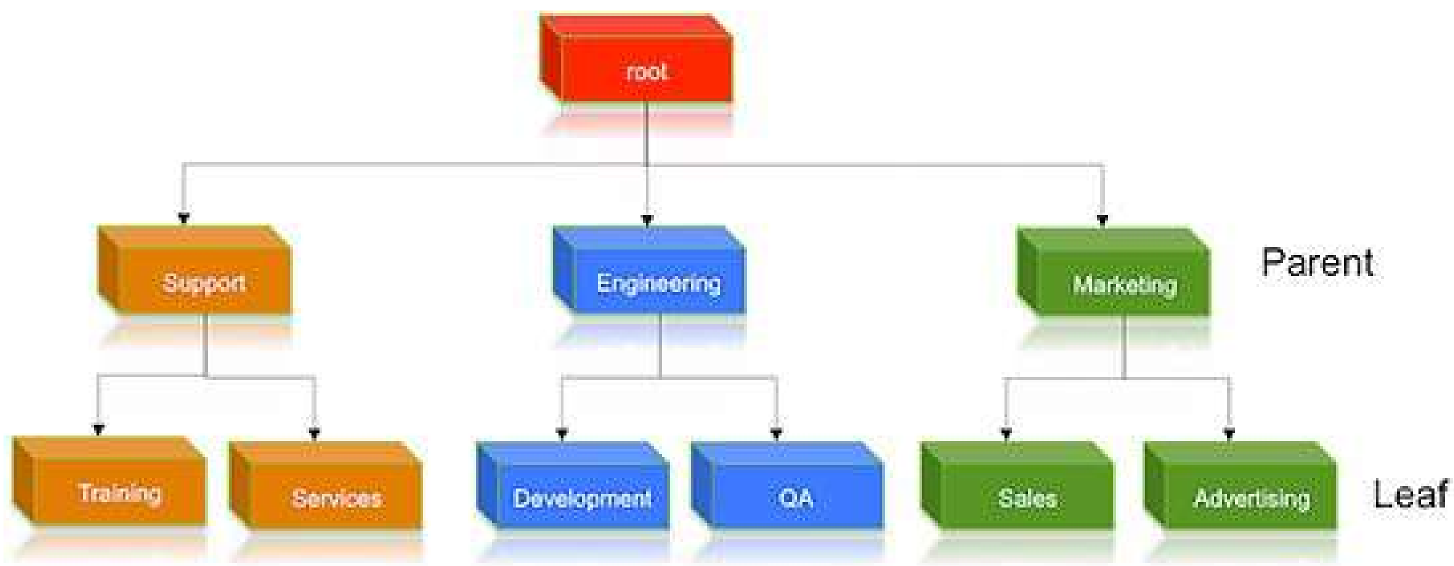
- **The fundamental unit of scheduling in YARN is the *queue***
- **The capacity specifies the percentage of cluster resources available for items submitted to the queue**
- **Each queue in the Capacity Scheduler has the following properties:**
  - A short queue name and path
  - A list of associated child-queues
  - The guaranteed capacity of the queue
  - The maximum capacity of the queue
  - A list of active users and their corresponding resource allocation limits
  - The state of the queue
  - Access control lists (ACLs) governing access to the queue

Hierarchical Queue Characteristics (1)

- **Consider the various characteristics of the Capacity Scheduler hierarchical queues before setting them up**
- **There are two types of queues: *parent* queues and *leaf* queues**
  - Parent queues enable the management of resources across organizations
  - Parent queues can contain more parent queues or leaf queues
  - Leaf queues are the queues that live under a parent queue and accept applications
  - Leaf queues do not have any child queues
  - There is a top-level parent root queue that does not belong to any organization and represents the cluster

Hierarchical Queue Characteristics (2)

- **Using parent and leaf queues, administrators can specify capacity allocations**
- **Every parent queue applies its capacity constraints to all of its child queues**
- **Leaf queues hold the list of active applications and schedules resources in a FIFO manner, while adhering to capacity limits specified for individual users**



Resource Distribution Workflow

- **During scheduling, queues are sorted in the order of their current used capacity**
- **Available resources are distributed to queues that are most under-served**
- **With respect to capacities, the resource scheduling has the following workflow:**
  - The more under-served a queue is, the higher the priority it receives during resource allocation
  - Once it is decided to give a parent queue the currently available free resources, scheduling is done recursively to determine which child queue gets to use the resources
  - Further scheduling happens inside each leaf queue to allocate resources to applications in a FIFO order
  - Capacity configured but not utilized by any queue due to lack of demand is assigned to queues that are in need of resources

CDH Fair Scheduler to CDP Capacity Scheduler

- **What we gain by switching to Capacity Scheduler:**
  - Scheduling throughput improvements
    - Look at several nodes at one time
    - Fine-grained locks
    - Multiple allocation threads
    - 5-10x throughput gains
- **Node partitioning and labeling**
- **The fs2cs convertor tool:**
  - To invoke the tool, you need to use the yarn fs2cs command with various command-line arguments
  - The tool generates two files as output: a capacity-scheduler.xml and a yarn-site.xml

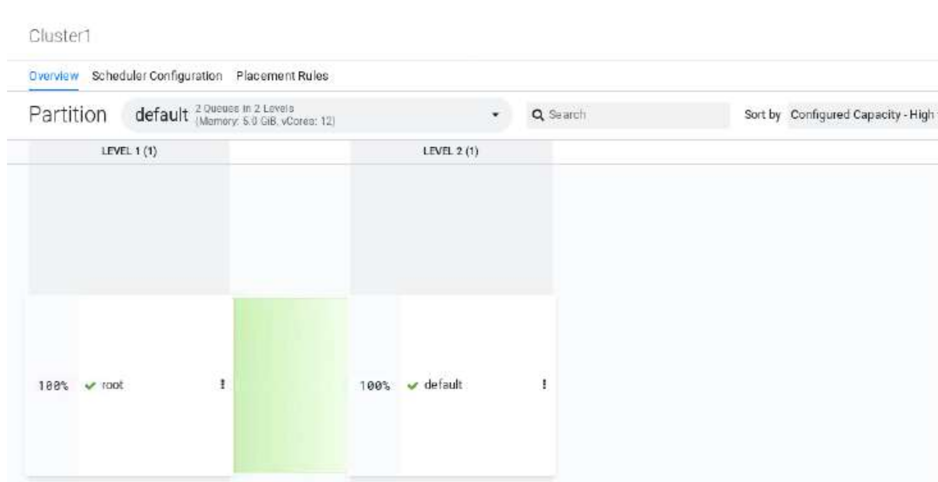
Chapter Topics

Managing Resources

- Managing Resources Overview
- Node Labels
- Configuring cgroups
- The Capacity Scheduler
- **Managing Queues**
- Impala Query Scheduling
- Essential Points
- Hands-On Exercise: Using The Capacity Scheduler

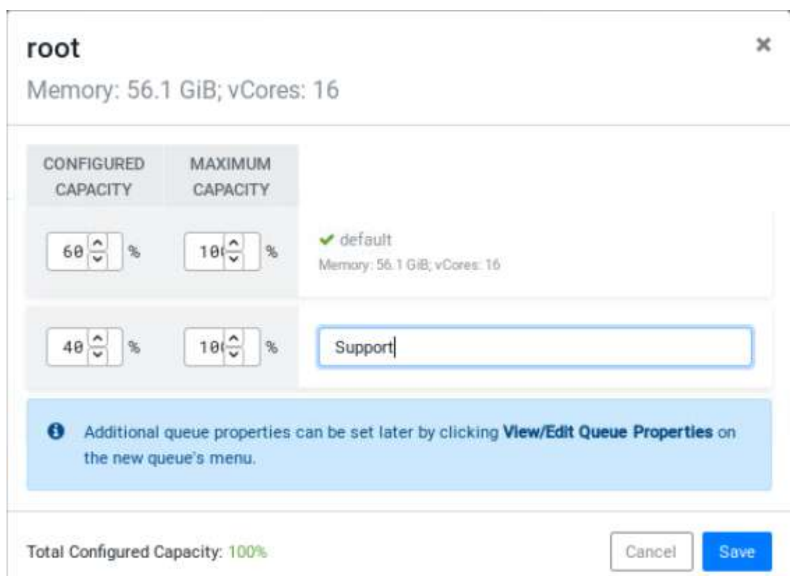
Manage Queues

- **YARN Queue Manager is the queue management GUI for YARN Capacity Scheduler**
  - Add the YARN Queue Manager service to the cluster
  - Enable YARN Queue Manager for the YARN service
  - Capacity Scheduler has a predefined queue called root
  - Manage your cluster capacity using queues
  - Configure queues to own a fraction of the capacity of each cluster



Adding Queues

- **In the YARN Queue Manager:**
  - Click on the three vertical dots on the root and select **Add Child Queue**
  - Enter the name of the queue, Configured Capacity, and Maximum Capacity values for the queue
  - To start a queue: select **Start Queue**
  - To stop a queue: select **Stop Queue**
  - Administrators can stop queues at run-time, so that while current applications run to completion, no new applications are accepted

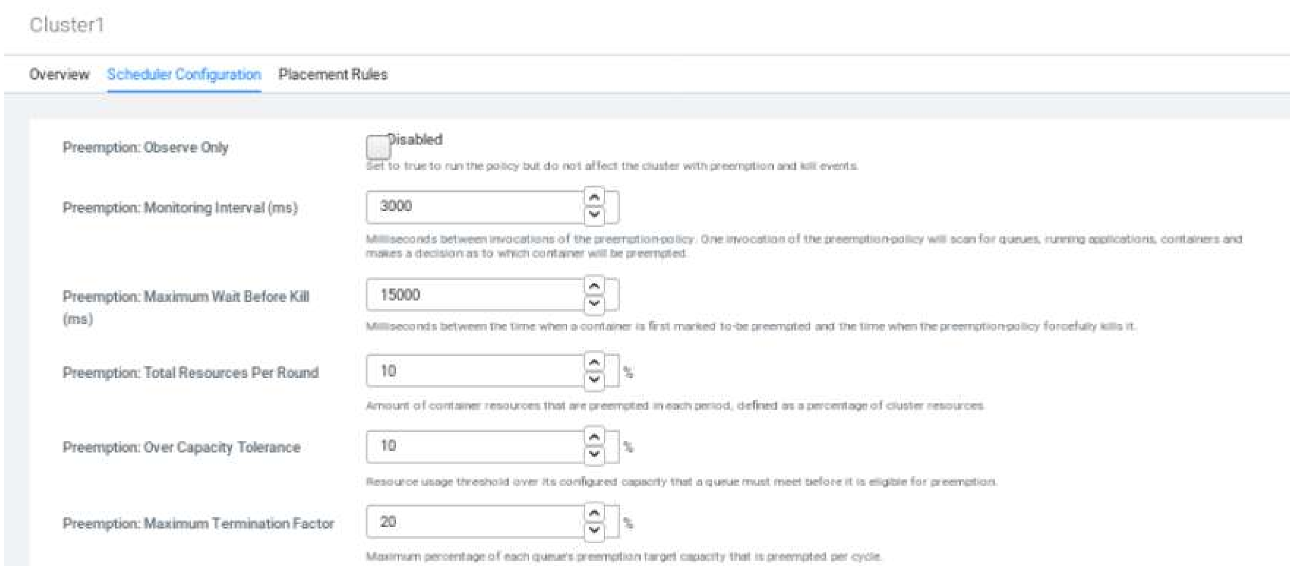


Queue Properties

- **Set user limits within a queue**
  - Set a minimum percentage of resources allocated to each leaf queue user
- **Set Maximum Application limit for a specific queue**
  - Configurable limit on the total number of concurrently active (both running and pending) applications at any one time
  - Set Application-Master resource-limit for percentage of cluster resources allocated to AM (default is 10%)
- **Set access to queues using ACLs**
- **Preemption allows higher-priority applications (those assigned to queues with a guaranteed level of cluster resources) to preempt lower-priority applications**
- **If the preemption policy is disabled in the scheduler configurations, you cannot enable preemption for a specific queue**

Configuring Preemption - Cluster-wide (1)

- **Capacity Scheduler Preemption allows higher-priority applications to preempt lower-priority applications.**
- **A scenario can occur in which a queue has a guaranteed level of cluster resources, but must wait to run applications because other queues are utilizing all of the available resources**
- **If preemption is enabled, applications of higher priority do not have to wait because applications of lower priority have taken up the available capacity**

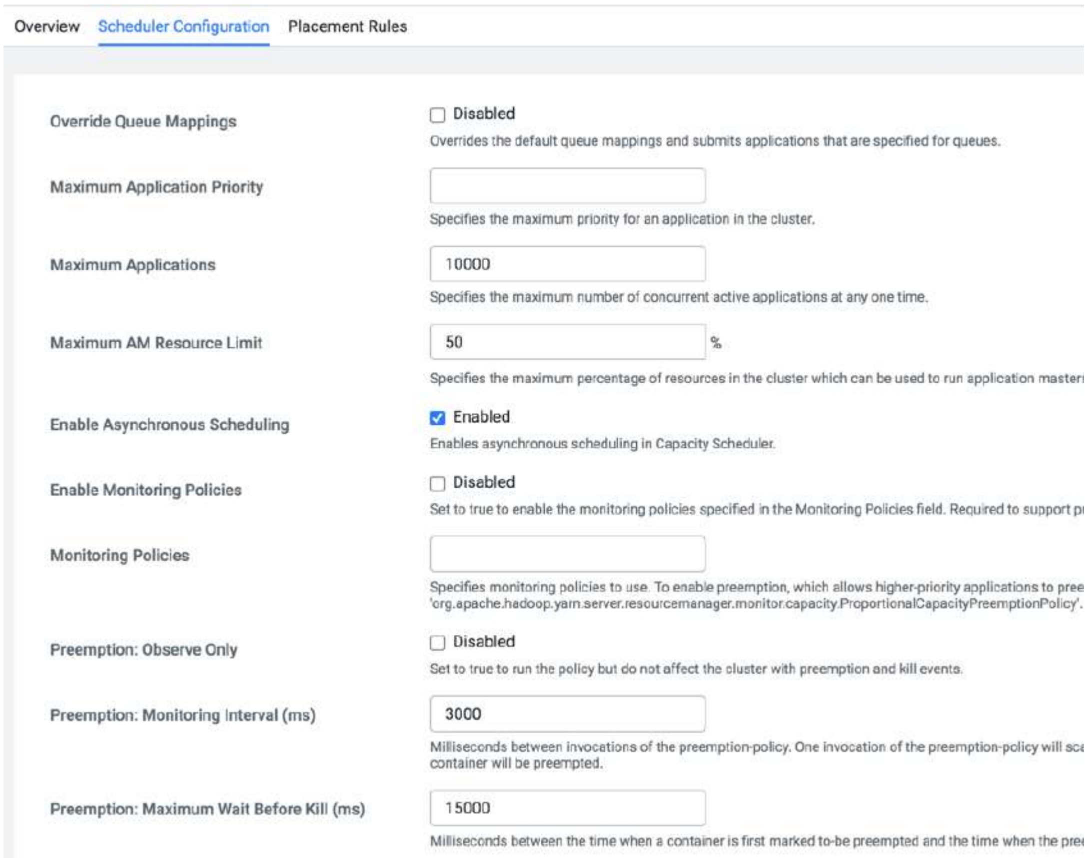


Configuring Preemption - Cluster-wide (2)

- **YARN Queue Manager, Scheduler Configuration tab**
- **Preemption Policies:**
  - Preemption: Observe Only - Runs the policy, but will not affect the cluster with preemption and kill events
  - Preemption: Monitoring Interval (ms) - The time between invocations of this policy
  - Preemption: Maximum Wait Before Kill (ms) - The time given to gracefully release containers before killing applications
  - Preemption: Total Resources Per Round - Use this value to restrict the pace at which containers are reclaimed from the cluster
  - Preemption: Over Capacity Tolerance - The default value is 0.1, which means that the Resource Manager starts preemption for a queue only when it goes 10% above its guaranteed capacity
  - Preemption: Maximum Termination Factor - The maximum percentage of each queue's preemption target capacity that is preempted per cycle

Global Level Scheduler Properties

- **Define the behaviour of all the queues**
- **Parent and children queues inherit the properties**

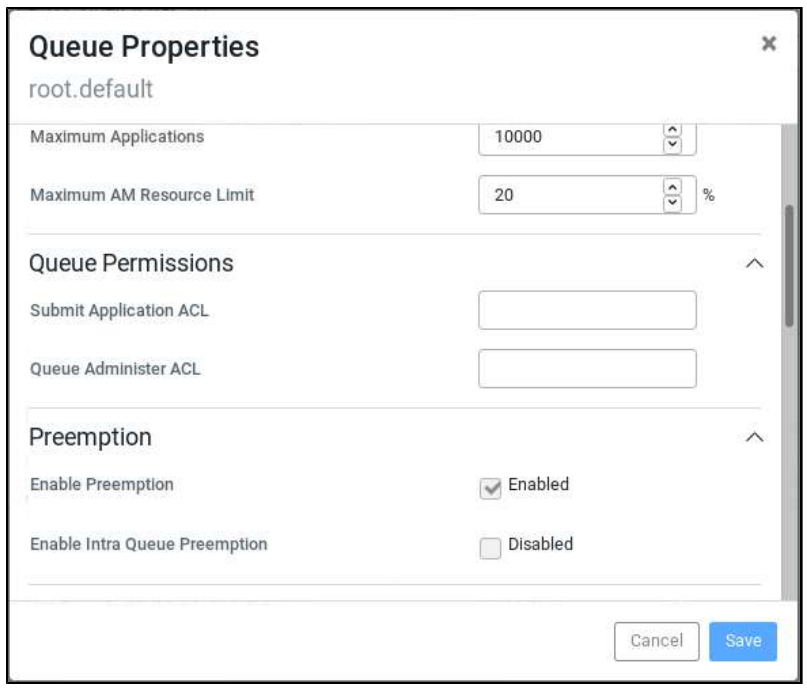


Control Access to Queues Using ACLs (1)

- Application submission can really only happen at the leaf queue level
- Restriction set on a parent queue will be applied to all of its descendant queues
- ACLs are configured by granting queue access to users and groups
- For administration of queues, set the Queue Administer ACL value

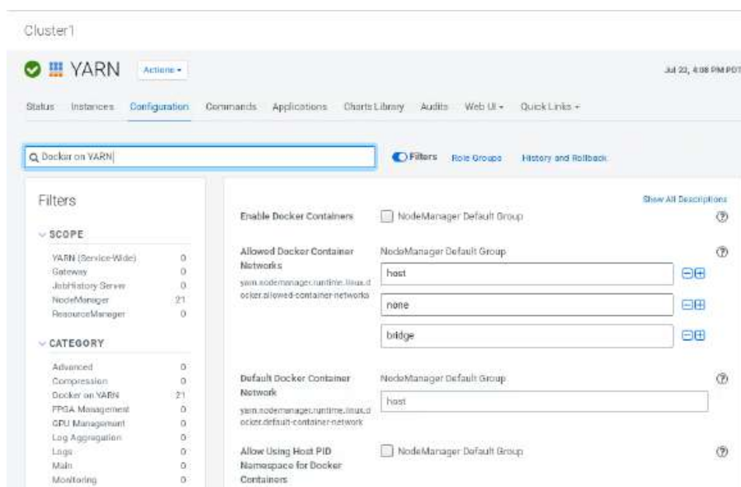
Control Access to Queues Using ACLs (2)

- Set the Submit Application ACL parameter to a comma-separated list of users, followed by a space, followed by a comma-separated list of groups
- Example: user1,user2 group1,group2
- Can also be set to "*" (asterisk) to allow access to all users and groups
- Can be set to " " (space character) to block access to all users and groups



Configuring YARN Docker Containers Support

- **You can configure YARN to run Docker containers**
  - Provides isolation and enables you to run multiple versions of the same applications
  - Using Docker containers introduces a new layer of virtualization, thus creates overhead compared to regular containers
  - YARN expects that Docker is already installed on all NodeManager hosts
  - After Docker is installed, edit the Docker's daemon configuration (daemon.json) file to add the recommended configuration
  - Use Cloudera Manager to configure YARN for managing Docker containers



Chapter Topics

Managing Resources

- Managing Resources Overview
- Node Labels
- Configuring cgroups
- The Capacity Scheduler
- Managing Queues
- **Impala Query Scheduling**
- Essential Points
- Hands-On Exercise: Using The Capacity Scheduler

Managing Resources in Impala

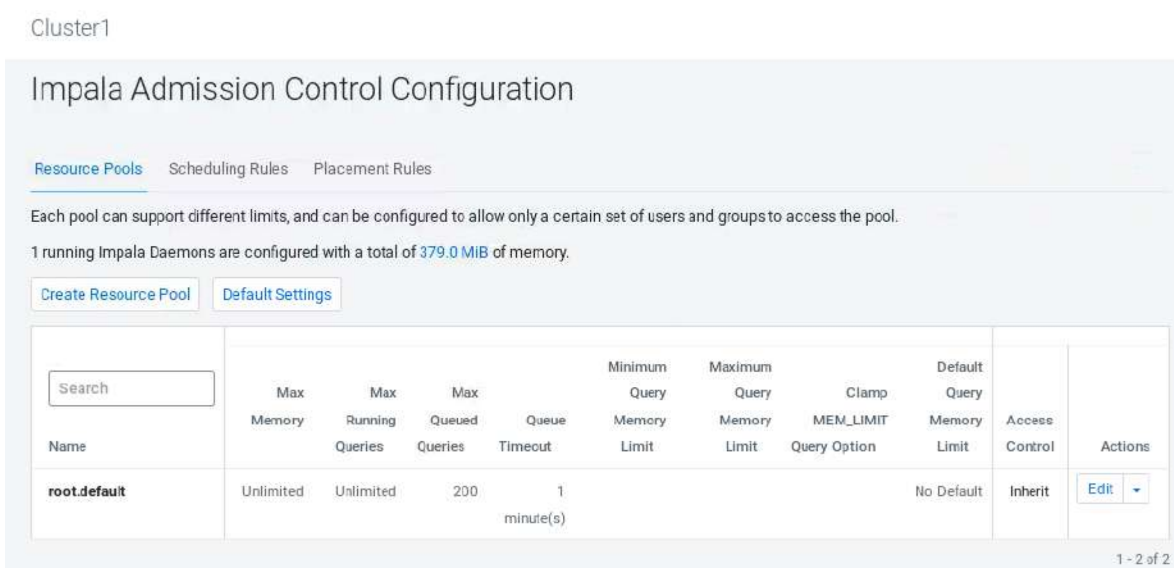
- **A typical deployment uses the following resource management features:**
  - Static service pools
    - Use the static service pools to allocate dedicated resources for Impala to manage and prioritize workloads on clusters
    - To configure: Select Static Service Pools from the Clusters menu
  - Admission control
    - Within the constraints of the static service pool, you can further subdivide Impala's resources using dynamic resource pools and admission control
    - Select both the Enable Impala Admission Control and the Enable Dynamic Resource Pools
- **Admission control and dynamic resource pools are enabled by default**
- **However, until you configure the settings for the dynamic resource pools, the admission control feature is effectively not enabled**

Impala Admission Control

- **Impala Admission Control provides a simple and robust way to manage per-pool resource utilization**
  - Enable this feature if the cluster is underutilized at some times and overutilized at others
- **Imposes resource scheduling on concurrent SQL queries**
  - Sets max number of concurrent Impala queries and memory used by queries in a given pool
  - Helpful for avoiding "out of memory" issues on busy clusters
- **Allows you to limit the number of queued (waiting) queries**
  - Prevents query coordinators from being overloaded

Configuring Dynamic Resource Pools

- **There is always a resource pool designated as root.default**
- **From the Impala Admission Control Configuration page**
  - Select Create Resource Pool button to add a new pool
  - Select Default Setting to set the default pool (which already exists) which is a catch all for ad-hoc queries
- **The default pool (which already exists) is a catch all for ad-hoc queries**



Recommendations for Tuning Admission Control

- **Goal is to increase throughput: queries finished per minute**
- **Secondary goal is to improve your bounded metrics - CPU, Memory, I/O**
- **Recommended steps (Perform one at a time):**
  1. Upgrade Impala: newer versions will improve performance
  2. Limit the total # of concurrently executing Impala queries system wide
  3. Limit the memory that an Impala query can use before swapping
  4. Protect Impala from Yarn contention with Static Service Pools
  5. Implement Admission Control Queues for different Impala User Groups
     - Limit concurrent queries for each queue/pool
     - Implement memory limits on the queue/pools and Running Queries
  6. Put a disk limit on runaway queries specifying the maximum amount of disk storage, in bytes, that any Impala query can consume on any host

Chapter Topics

Managing Resources

- Managing Resources Overview
- Node Labels
- Configuring cgroups
- The Capacity Scheduler
- Managing Queues
- Impala Query Scheduling
- **Essential Points**
- Hands-On Exercise: Using The Capacity Scheduler

Essential Points

- **cgroups allow you to partition cluster resources between different Hadoop frameworks such as MapReduce and Impala**
  - Configurable by defining static service pools
- **The Capacity Scheduler is the default scheduler**
  - Allow resources to be controlled proportionally
  - Ensure that a cluster is used efficiently
- **Configure Admission Control and Dynamic Resource Pools to allocate resources to Impala queries**

Chapter Topics

Managing Resources

- Managing Resources Overview
- Node Labels
- Configuring cgroups
- The Capacity Scheduler
- Managing Queues
- Impala Query Scheduling
- Essential Points
- **Hands-On Exercise: Using The Capacity Scheduler**

Hands-On Exercise: Using The Capacity Scheduler

- **In this exercise, you will run YARN jobs in different pools and observe how the Capacity Scheduler handles the jobs. Impala Admission control will also be utilized with Dynamic Resource Pools.**
- **Please refer to the Hands-On Exercise manual**

Planning Your Cluster

Chapter 11

Course Chapters

Planning Your Cluster

After completing this chapter, you will be able to

- **Summarize issues to consider when planning a Hadoop cluster**
- **Identify types of hardware used in a cluster**
- **Configure your network topology for optimal performance**
- **Select the right OS and Hadoop distribution**
- **Plan for cluster management**

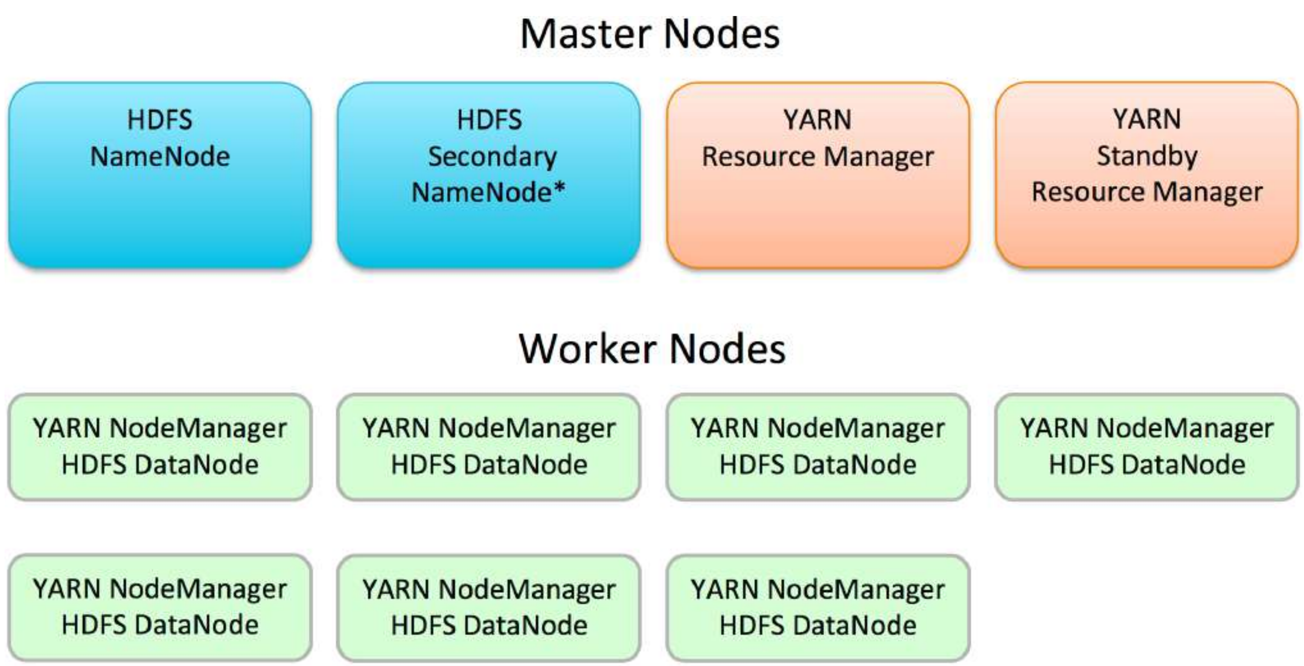
Chapter Topics

Planning Your Cluster

- **General Planning Considerations**
- Choosing the Right Hardware
- Network Considerations
- CDP Private Cloud Considerations
- Configuring Nodes
- Essential Points

Basic Cluster Configuration

- CDP Private Cloud Base is an on-premises version of Cloudera Data Platform
- A scalable and customizable platform where you can securely run many types of workloads
- Supports a variety of hybrid solutions where compute tasks are separated from data storage



* in this configuration, HDFS is not highly available. HA details discussed later.

Thinking About the Problem

- **Hadoop can run on a single machine**
  - For testing and development
  - Not intended for production systems
- **Many organizations start with a small cluster and grow it as required**
  - Perhaps initially just eight or ten machines
- **Grow the cluster when you need to increase**
  - Computation power
  - Data storage
  - Memory
- **Growth needs depend on cluster use cases**

Increasing Cluster Storage Capacity

- **Increased storage capacity is the most common reason for expanding a cluster**
- **Example: Data is growing by approximately 3 TB per week**
  - HDFS replicates each block three times
  - Add overhead of 10-25% of the amount data stored
  - Result: About 650-780 TB per year (10-11 TB per week + 25% overhead) of extra storage space
  - 8-10 additional machines with 16 3 TB hard drives required every year

Planning Your Cluster

- **Cloudera publishes reference architectures for several types of deployments such as**
  - Physical (“bare metal”) and VMware virtual machine deployments
  - Cloud deployments on Amazon AWS, Microsoft Azure, and Google Cloud Platform
  - Private cloud deployments
  - Deployments with and without high availability
- **See Cloudera’s documentation website for specifics**

Chapter Topics

Planning Your Cluster

- General Planning Considerations
- **Choosing the Right Hardware**
- Network Considerations
- CDP Private Cloud Considerations
- Configuring Nodes
- Essential Points

Planning Hardware Needed for Different Host Types (1)

- **Number and configuration of hosts depends on types of services**
- **Worker hosts**
  - Typically run HDFS DataNode, YARN NodeManager, and Impala Server daemons
- **Master hosts**
  - Typically run a NameNode, Standby NameNode or Secondary NameNode, Impala StateStore, Impala Catalog, or ResourceManager daemon
  - May combine multiple roles such as NameNode and ResourceManager in smaller clusters

Planning Hardware Needed for Different Host Types (2)

- **Utility hosts**
  - Store configurations for client applications
  - Typically run Cloudera Manager and Management services or Hive
  - May also run additional services such as Hue, Oozie, or HiveServer2
- **More machines may be required for any additional cluster services, such as HBase, Kudu, Kafka, and database systems**

Worker Hosts—Recommended Configurations

- **Vary based on services and the type of workload**
- **Example of a typical configuration**
  - 12-24 x 1-4 TB hard drives for data storage and overhead (non-RAID*, JBOD†) configuration
  - Two drives for the OS (RAID-1 mirroring)
  - 12-14 core CPU
  - 256 GB RAM for running MR and Spark or Impala
  - 384 GB RAM for running MR and Spark and Impala
  - 10 Gigabit Ethernet or 10 GbE Bonded for throughput
- **The number of cores should be equal to, or more than, the number of disks**
- **The amount of memory depends on number of logical containers used for Spark or MapReduce with extra for certain processes**
- **There is no one-size-fits-all approach that works for everyone**

* RAID: Redundant Array of Independent Disks
† JBOD: Just a Bunch Of Disks

Worker Hosts—CPU

- **Hadoop applications are typically disk- and network-I/O bound**
  - Top-of-the-range CPUs are usually not necessary
- **Hyper-threading and quick-path interconnect (QPI) should be enabled**
- **Some types of Hadoop jobs make heavy use of CPU resources**
  - Clustering and classification
  - Complex text mining
  - Natural language processing
  - Feature extraction
  - Image manipulation
- **Your workload may need more processing power**
- **Rule of thumb: Number of physical processor cores = total number of tasks/ executors plus one**
  - This is a starting point, not a definitive rule for all clusters

Worker Hosts—RAM (1)

- **Worker host configuration limits the amount of memory and number of cores ApplicationsMasters and tasks/executors can use on that host**
- **Each ApplicationMaster typically takes at least 1 GB of RAM**
  - Could be much more depending on type of application
- **No hosts should use virtual memory/swap space**
- **Ensure you have enough RAM to run all tasks**
  - Plus overhead for the DataNode and NodeManager daemons, and the operating system

Worker Hosts—RAM (2)

- **Impala and Spark are more memory-intensive than MapReduce**
- **HDFS caching can also take advantage of extra RAM on worker hosts**
- **Equip your worker hosts with as much RAM as you can**
  - Memory configurations of 1 - 2 TB are common for workloads with high memory requirements

Worker Hosts—Disk (1)

- **Hadoop's architecture impacts disk space requirements**
  - By default, HDFS data is replicated three times
  - Temporary data storage typically requires 20-30 percent of a cluster's raw disk capacity
- **In general, more spindles (disks) is better**
- **2 disks >= 500 GB for OS and logs (RAID 1)**
- **Maximum JBOD drives and capacities supported for Hadoop storage:**
  - 24 disks <= 4TB (recommended) *or*
  - 12 disks <= 8TB
- **Maximum 100 TB per DataNode host**
- **5,000-10,000 RPM SATA drives**
  - 15,000 RPM drives are not necessary
- **8 x 1.5TB drives is likely to be better than 6 x 2TB drives**
  - Different tasks are more likely to be accessing different disks

Worker Hosts—Disk (2)

- **Consider node density when allocating disks**
  - More storage requires more network traffic if a host dies and blocks must be re-replicated
  - For example, a good practical maximum is 36 TB per worker host with 10 Gb per second NIC *
  - More than that will result in massive network traffic
- **Recommendation: dedicate 1 disk for OS and logs (potentially mirrored)**
  - Use the other disks for Hadoop data
- **Using SSDs for non-compressed intermediate shuffle data leads to significant performance gains**

* Network Interface Controller

Host Failure

- **Worker hosts are expected to fail at some point**
  - This assumption is built into Hadoop
  - NameNode will automatically re-replicate blocks that were on the failed host to other hosts in the cluster to maintain the configured replication factor
  - ApplicationMasters will automatically re-assign tasks that were running on failed hosts
  - ResourceManagers will relaunch ApplicationMasters running on failed hosts
- **Master hosts are single points of failure if not configured for HA**
  - If the NameNode goes down, the cluster is inaccessible
  - If the ResourceManager goes down, no new jobs can run on the cluster
- **Configure the NameNode and ResourceManager for HA when running production workloads**

Master Host Hardware Recommendations

- **Carrier-class hardware**
- **Dual power supplies**
- **4–6 1TB hard disks in a JBOD configuration**
  - 2 >= 500 GB for the OS (RAID 1)
  - 2 >= 1 TB for NameNode metadata (RAID 1)
  - 1 >= 1 TB for JournalNodes (RAID 0 or JBOD)
  - 1 >= 1 TB for Apache ZooKeeper (RAID 0 or JBOD, no SSD)
- **Dual Ethernet cards**
  - Bonded to provide failover
- **8 core CPU at >= 2.6 GHz**
- **Reasonable amount of RAM**
  - 256 GB minimum

Chapter Topics

Planning Your Cluster

- General Planning Considerations
- Choosing the Right Hardware
- **Network Considerations**
- CDP Private Cloud Considerations
- Configuring Nodes
- Essential Points

General Network Considerations (1)

- **Hadoop is very bandwidth-intensive!**
  - Often, all hosts are communicating with each other at the same time
- **Use dedicated switches for your Hadoop cluster**
- **Hosts should be connected to a top-of-rack switch**
- **Hosts should be connected at a minimum speed of 10Gb/sec**

General Network Considerations (2)

- **Racks are interconnected using core switches**
- **Core switches should connect to top-of-rack switches at 10Gb/sec or faster**
- **Avoid oversubscription in top-of-rack and core switches**
- **Consider bonded Ethernet to mitigate against failure and increase throughput**
- **At least two top-of-rack and core switches**
  - Consider four for throughput and redundancy

Hostname Resolution

- **You will identify hosts during initial cluster setup**
- **Use host names, not IP addresses, to identify hosts**
- **Each host must be able to**
  - Perform a forward lookup on its own hostname
  - Perform a reverse lookup using its own IP address
  - Forward and reverse lookups *must* work correctly
- **DNS is preferred over definitions in `/etc/hosts` for hostname resolution**
  - Set host names to fully qualified domain name (FQDN)

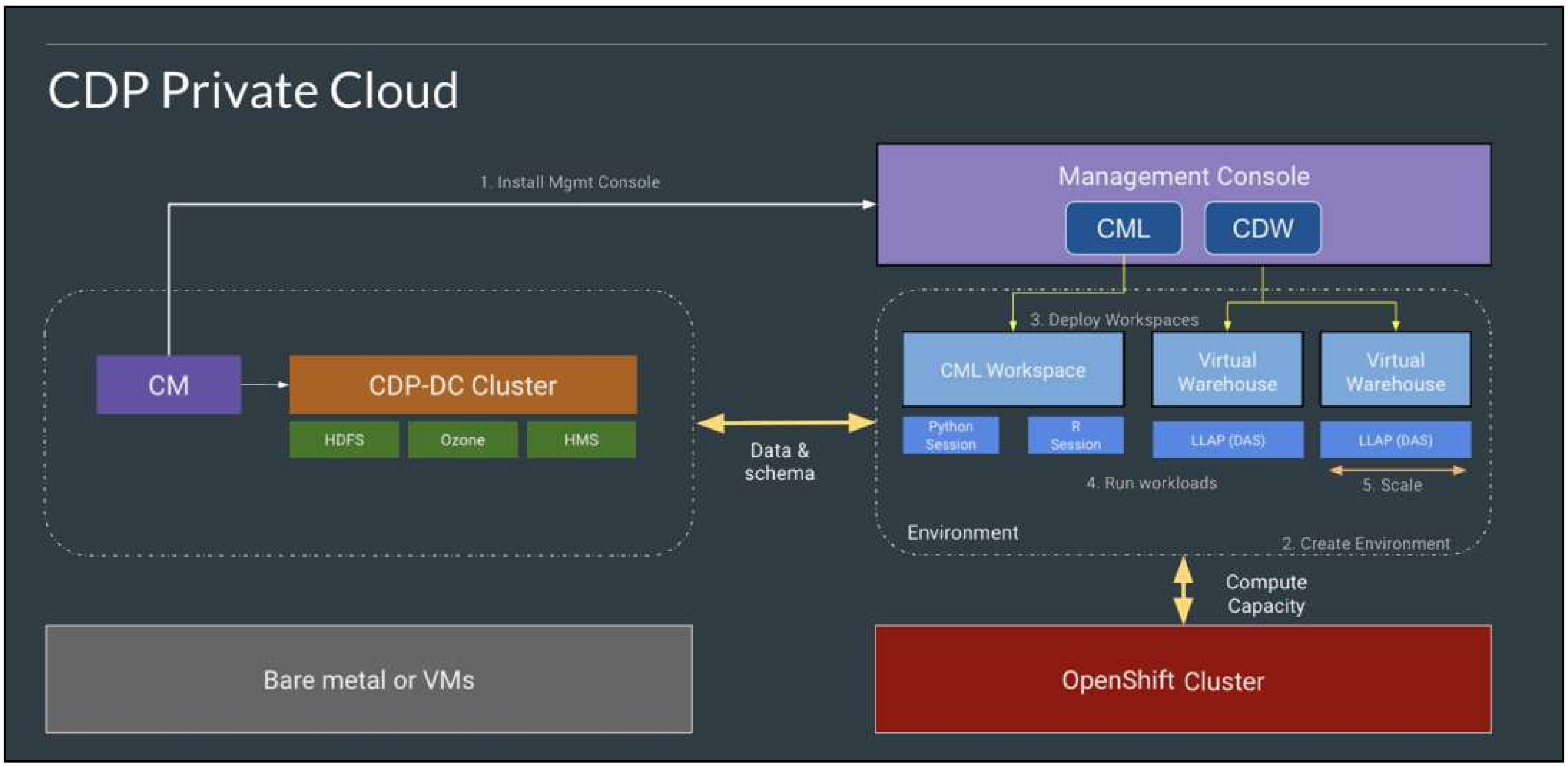
Chapter Topics

Planning Your Cluster

- General Planning Considerations
- Choosing the Right Hardware
- Network Considerations
- **CDP Private Cloud Considerations**
- Configuring Nodes
- Essential Points

Planning Considerations for Upgrade to Private Cloud

- **The Private Cloud Management Console is deployed from a Private Cloud Base Cloudera Manager instance**
- **See the option Private Cloud (new) in the Navigation menu**
- **Solution Architecture:**



Openshift

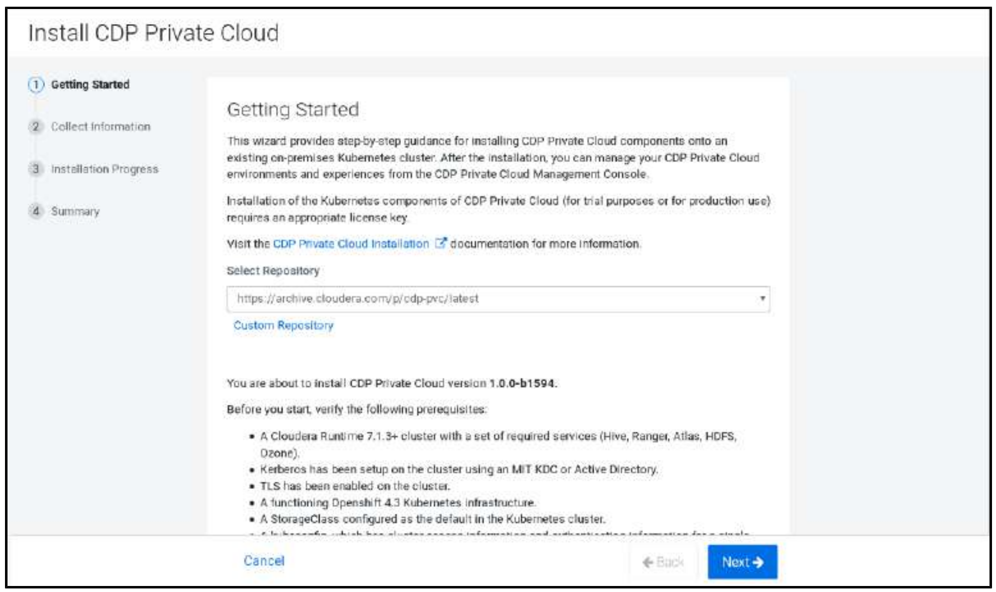
- **Private Cloud requires one OpenShift cluster for the control plane and the environments:**
  - The OpenShift Container Platform will be provided or it will be installed separately by Cloudera
  - The OpenShift cluster must be dedicated to Cloudera Private Cloud.
  - Openshift is a cloud development Platform as a Service (PaaS) developed by Red Hat
  - It enables the developers to develop and deploy their applications on cloud infrastructure
  - It is very helpful in developing cloud-enabled services

Pre-Installation Steps

- **You will complete the following steps to prepare for installation:**
  - Start with an existing Cloudera Manager
  - Set up the Management Console on OpenShift
  - Use the Management Console to define an environment
  - Bring up workloads (CML workspace, CDW Virtual Warehouses) in that environment

Private Cloud Management Console

- **Deployed from a Private Cloud Base Cloudera Manager**
- **Manage Cloudera Data Platform environments, users, and services**
- **The Private Cloud Base may have one or more data lake clusters**
- **From the Private Cloud Management Console, you can create one or more environments associated with any of the data lakes from the base clusters**



Chapter Topics

Planning Your Cluster

Operating System Recommendations

- **Choose a distribution that you are comfortable administering**
- **CentOS: geared towards servers instead of individual workstations**
  - Conservative about package versions
  - Widely used in production
- **RedHat Enterprise Linux (RHEL): RedHat-supported analog to CentOS**
  - Includes support contracts, for a price
- **Oracle Linux (OL)**
- **See release notes for details on supported operating system versions for each release of CDP Private Cloud Base**
- **Ubuntu: Very popular distribution, based on Debian (starting in CDP 7.1.3)**
  - Both desktop and server versions available
  - Not supported for Cloudera Schema Registry

Host Machine Configuration (1)

- **Do not use Linux's LVM (Logical Volume Manager) to make all your disks appear as a single volume**
  - As with RAID 0, this limits speed to that of the slowest disk
  - Can also result in the loss of all data on the host if a single disk fails
- **Configure BIOS* settings for best performance**
  - For example, make sure IDE emulation is not enabled for SATA drives
- **Test disk I/O speed with `hdparm`**
  - Example: `hdparm -t /dev/sda1`
  - You should see speeds of 100 MB/second or more
    - Slower speeds may indicate problems

* BIOS: Basic Input/Output System

Host Machine Configuration (2)

- **Reduce `vm.swappiness` to 1**
  - Set in `/etc/sysctl.conf`
- **Configure IPTables if required by your security policies—Hadoop requires many ports for communication**
  - From the Cloudera Manager **Cluster** page, choose **Configuration** > **All Port Configurations** to see all ports used
- **Open ports required by Cloudera in system firewall**
  - See `https://docs.cloudera.com/cdp-private-cloud-base/7.1.3/installation/topics/cdpdc-ports.html` for a list of required ports

Host Machine Configuration (3)

- **Hadoop has no specific disk partitioning requirements**
  - One partition per disk is recommended
- **Mount disks with the `noatime` option**
- **Common directory structure for data mount points**

```
/data/n/dfs/nn
/data/n/dfs/dn
/data/n/dfs/snn
/data/n/yarn/nm
```

Host Machine Configuration (4)

- **Disable Transparent Huge Page compaction**
  - Can degrade the performance of Hadoop workloads
  - Disable in `defrag` file of your OS
    - Red Hat/CentOS: `/sys/kernel/mm/redhat_transparent_hugepage/defrag`
    - Ubuntu, OEL, SLES: `/sys/kernel/mm/transparent_hugepage/defrag`
  - Make sure `never` is selected in brackets
    - For example: `[never] always`

Host Machine Configuration (5)

- **Disable IPv6**
- **Disable SELinux if possible**
  - Incurs a performance penalty on a Hadoop cluster
  - Configuration is complicated
  - Disable it on each host before deploying CDP on the cluster
  - Confirm setting with `sestatus` command
- **Install and configure Network Time Protocol (NTP)**
  - Ensures the time on all hosts is synchronized
    - Important for HBase, ZooKeeper, Kerberos
    - Useful when using logs to debug problems

Filesystem Considerations

- **Cloudera recommends that you use one of the following filesystems tested on the supported operating systems**
  - ext3: The most tested underlying filesystem for HDFS
  - ext4: Scalable extension of ext3, supported in more recent Linux releases
  - XFS: The default filesystem in RHEL 7
  - S3: Amazon Simple Storage Service
- **Kudu is supported on ext4 and XFS**

Java Virtual Machine (JVM) Requirements

- **Supports 64-bit versions of both Oracle JDK and OpenJDK**
- **Running Runtime nodes within the same cluster on different JDK releases is not supported**
- **All cluster hosts must use the same JDK update level**
- **For version specific information see**
  - `https://www.cloudera.com/documentation/enterprise/release-notes/topics/rn_consolidated_pcm.html#pcm_jdk`

The Cloudera Manager Host Inspector

- **The Host Inspector checks for many of the items just discussed**
  - Validates select OS settings, networking settings, system time, user and group settings, and component versions
- **Run host inspector whenever a new host is added to the cluster**
  - Runs automatically during initial installation

Chapter Topics

Planning Your Cluster

- General Planning Considerations
- Choosing the Right Hardware
- Network Considerations
- CDP Private Cloud Considerations
- Configuring Nodes
- **Essential Points**

Essential Points

- **Master hosts run the NameNode, Standby NameNode (or Secondary NameNode), and ResourceManager**
  - Provision with carrier-class hardware
- **Worker hosts run DataNodes and NodeManagers**
  - Provision with industry-standard hardware and lots of RAM
  - Consider your data storage growth rate when planning current and future cluster size
- **Make sure that forward and reverse domain lookups work when configuring a cluster**
- **Plan the cluster according to needs considering the right OS, Hadoop Distribution and network topology for optimal performance of needed workloads**

Advanced Cluster Configuration

Chapter 12

Course Chapters

- Introduction
- Cloudera Data Platform
- CDP Private Cloud Base Installation
- Cluster Configuration
- Data Storage
- Data Ingest
- Data Flow
- Data Access and Discovery
- Data Compute
- Managing Resources
- Planning Your Cluster
- **Advanced Cluster Configuration**
- Cluster Maintenance
- Cluster Monitoring
- Cluster Troubleshooting
- Security
- Private Cloud / Public Cloud
- Conclusion
- Appendix: Cloudera Manager API
- Appendix: Ozone Overview

Advanced Cluster Configuration

After completing this chapter, you will be able to

- **Configure port numbers used by Hadoop**
- **Tuning HDFS and MapReduce**
- **Managing Cluster Growth**
- **Erasure Coding**
- **Enable HDFS high availability**

Chapter Topics

Advanced Cluster Configuration

- **Configuring Service Ports**
- Tuning HDFS and MapReduce
- Managing Cluster Growth
- Erasure Coding
- Enabling High Availability for HDFS and YARN
- Essential Points
- Hands-On Exercise: Configuring HDFS for High Availability

Configuring Hadoop Ports (1)

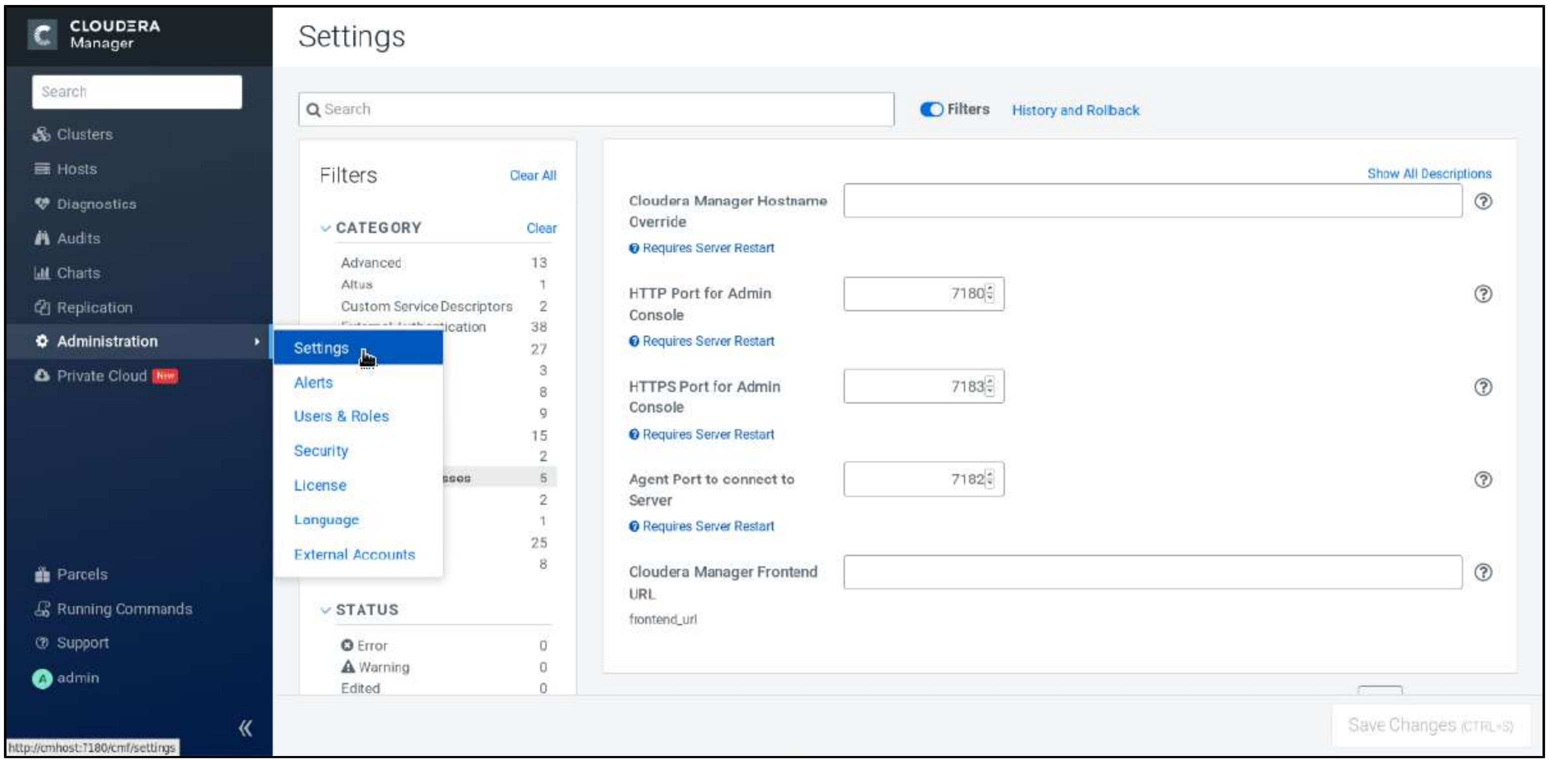
- **Many Hadoop daemons provide a web-based user interface**
  - Useful for both users and system administrators
- **Hadoop also uses various ports for components of the system to communicate with each other***
- **Cloudera Manager sets default port numbers**
- **UI examples**
  - Cloudera Manager: port 7180 on CM host
  - HDFS DataNode: port 9870 on worker hosts
  - Impala daemon: port 25000 on Impala worker hosts
- **Cluster communication examples**
  - NameNode file system metadata operations: port 8020
  - ResourceManager application submission: port 8032

* Full list of ports used by components of CDP: `https://docs.cloudera.com/cdp-private-cloud-base/7.1.3/installation/topics/cdpdc-ports.html`

Configuring Hadoop Ports (2)

- **Port numbers are configurable**
  - Override Cloudera Manager ports using the **Administration > Settings > Ports and Addresses** menu
  - Override service daemon ports on the cluster configuration page

Cloudera Manager Cluster Page



Chapter Topics

Advanced Cluster Configuration

- Configuring Service Ports
- **Tuning HDFS and MapReduce**
- Managing Cluster Growth
- Erasure Coding
- Enabling High Availability for HDFS and YARN
- Essential Points
- Hands-On Exercise: Configuring HDFS for High Availability

Advanced Configuration Parameters

- **These generally fall into one of several categories**
  - Optimization and performance tuning
  - Capacity management
  - Access control
- **The configuration recommendations in this section are baselines**
  - Use them as starting points, then adjust as required by the job mix in your environment

HDFS NameNode Tuning

`dfs.namenode.handler.count`
Set in HDFS / NameNode Group / Performance

- The number of server threads for the NameNode that listen to requests from clients
- Threads used for RPC calls from clients and DataNodes (heartbeats and metadata operations)
- Cloudera Manager default: 30 (Apache default: 10)
- Recommended: Natural logarithm of the number HDFS nodes x 20
- Symptoms of this being set too low: “connection refused” messages in DataNode logs when transmitting block reports to the NameNode
- Used by the NameNode

HDFS DataNode Tuning

`dfs.datanode.failed.volumes.tolerated`
Set in HDFS / DataNode Group

- The number of volumes allowed to fail before the DataNode takes itself offline (all of its blocks will be re-replicated)
- Cloudera Manager Default: 0
- For each DataNode, set to (number of mountpoints on DataNode host) / 2
- Used by DataNodes

`dfs.datanode.max.locked.memory`
Set in HDFS / DataNode Group / Resource Management

- The maximum amount of memory (in bytes) a DataNode can use for caching
- Cloudera Manager Default: 4GB
- Must be less than the value of the OS configuration property **`ulimit -l`** for the DataNode user
- Used by DataNodes

File Compression

`io.compression.codecs`
Set in HDFS / Service-Wide

- List of compression codecs that Hadoop can use for file compression
- If you are using another codec, add it here
- Cloudera Manager default value includes the following **`org.apache.hadoop.io.compress`** codecs: **`DefaultCodec`**, **`GzipCodec`**, **`BZip2Codec`**, **`DeflateCodec`**, **`SnappyCodec`**, **`Lz4Codec`**
- Used by clients and all nodes running Hadoop daemons

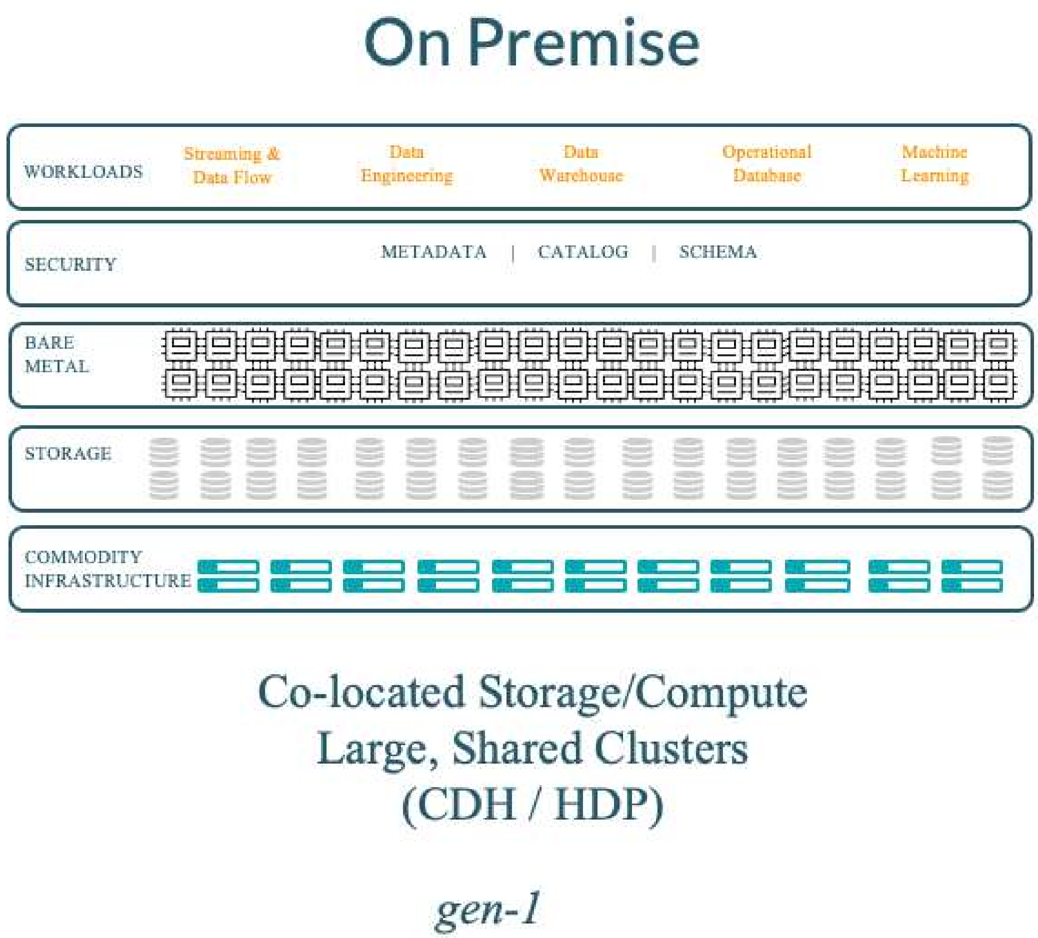
Chapter Topics

Advanced Cluster Configuration

- Configuring Service Ports
- Tuning HDFS and MapReduce
- **Managing Cluster Growth**
- Erasure Coding
- Enabling High Availability for HDFS and YARN
- Essential Points
- Hands-On Exercise: Configuring HDFS for High Availability

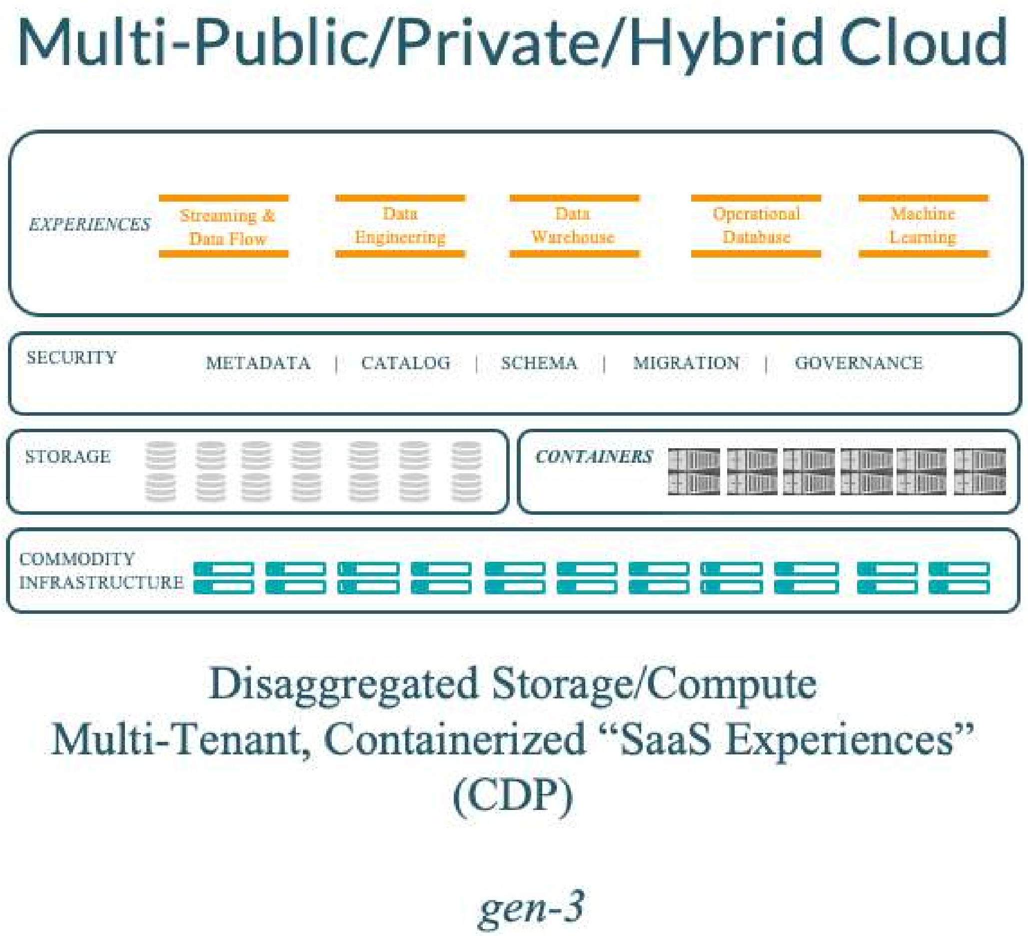
Evolution of Architecture: The First Decade

- **We co-located compute and storage in an on-premise deployment**



Evolution of Architecture: The Second Decade

- **We disaggregate the software stack — storage, compute, security and governance**

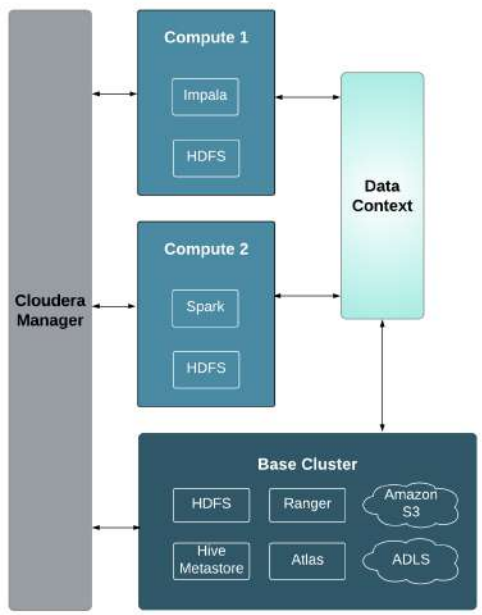


Separate Compute and Storage

- **Important advantages for many workloads**
  - More options for deploying computational and storage resources
  - Tailor the deployment resources using on-premise servers, containers, virtual machines, or cloud resources
  - Provision a Compute cluster with hardware for computational workloads
  - Base cluster can use hardware that emphasizes storage capacity
- **Ephemeral clusters**
  - When deploying clusters on cloud infrastructure, you can temporarily shut down the compute clusters and avoid unnecessary expense
  - While still leaving the data available to other applications
- **Workload Isolation**
  - Compute clusters can help to resolve resource conflicts
  - Longer running or resource intensive workloads can be isolated to run in dedicated compute clusters
  - Grouped into clusters that allow IT to allocate costs to the teams that use the resources

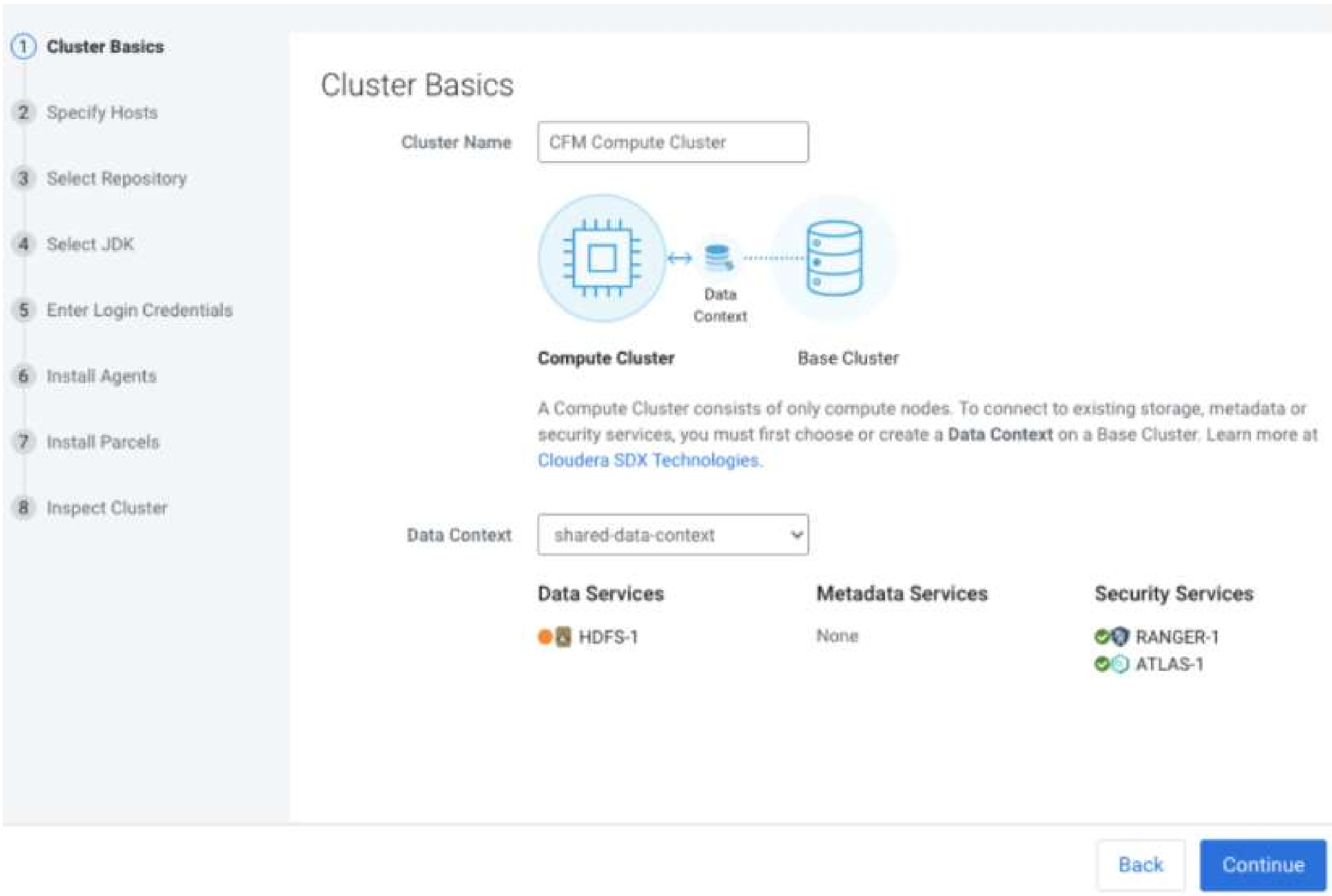
Architecture

- **A Compute cluster is configured with compute resources such as YARN, Spark, Hive Execution, or Impala**
  - Workloads access data by connecting to a Data Context for the Base cluster
  - Data Context is a connector that defines the data, metadata and security
  - Compute cluster and Base cluster are managed by the same instance of CM
  - Only HDFS, Hive, Atlas, Ranger, Amazon S3, and ADLS can be shared using the data context



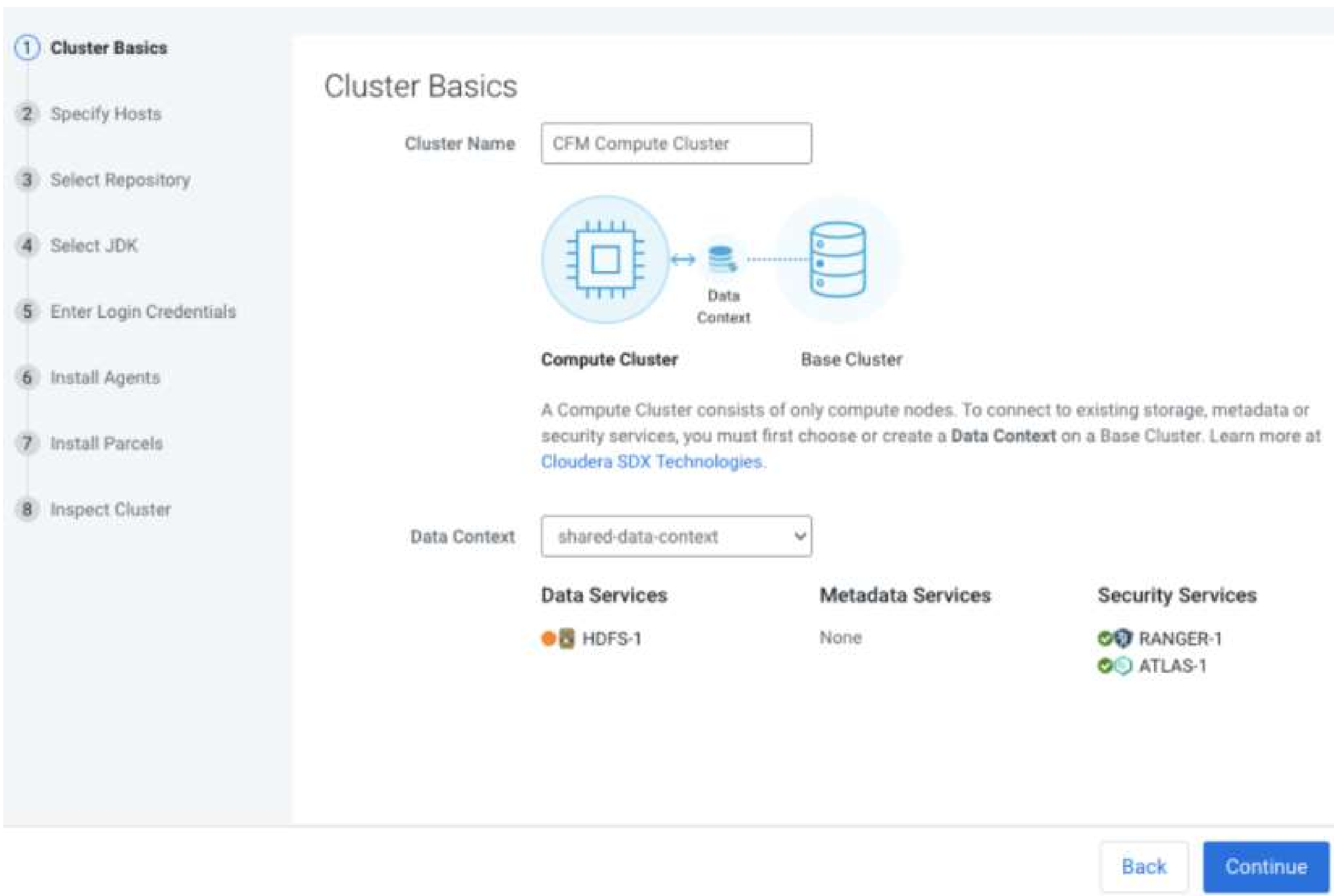
Context

- **You can select a context when creating a Compute Cluster**



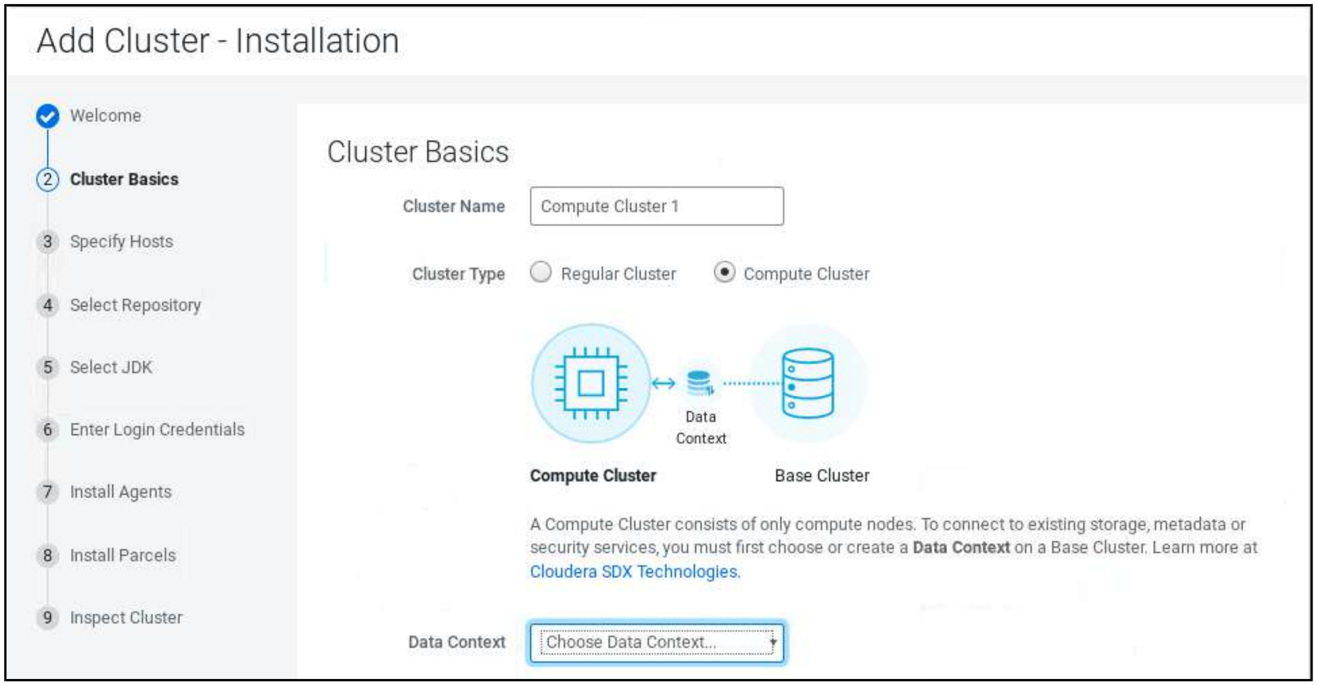
Context

- **You can create a context when creating a Compute Cluster**



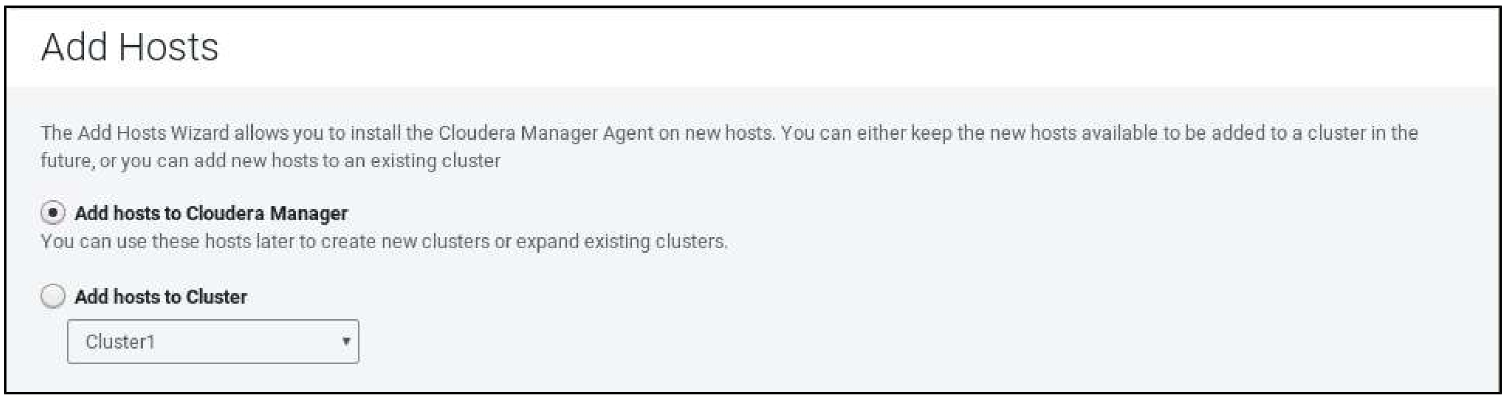
Add a Cluster

- **From the Cluster Status page, select** Add > Cluster
- **Enter the Cluster Name**
- **Select Cluster Type**
- **Choose or Create the Data Context**



Add a Host

- **You can add one or more hosts to your cluster using the Add Hosts wizard**
- **Installs the Oracle JDK, CDP, and Cloudera Manager Agent software**
- **The Add Hosts wizard does not create roles on the new host**
- **Either add roles, one service at a time, or apply a host template**



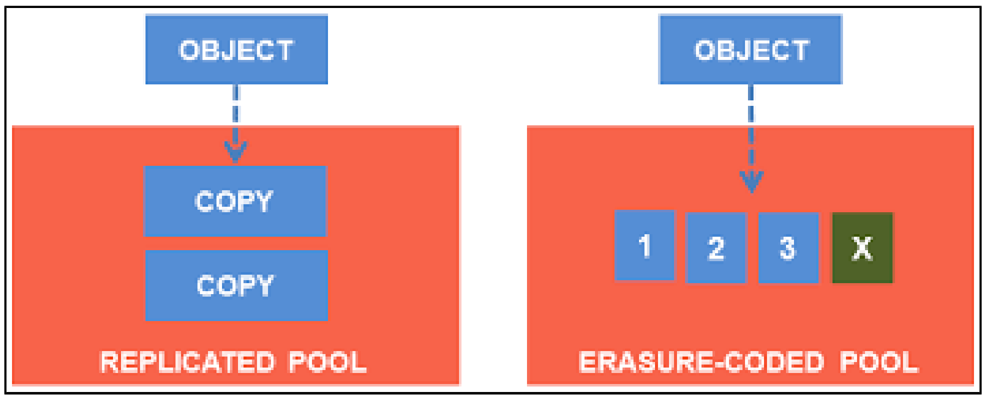
Chapter Topics

Advanced Cluster Configuration

- Configuring Service Ports
- Tuning HDFS and MapReduce
- Managing Cluster Growth
- **Erasure Coding**
- Enabling High Availability for HDFS and YARN
- Essential Points
- Hands-On Exercise: Configuring HDFS for High Availability

Data Durability

- **How resilient data is to loss**
- **CDP provides two options for data durability**
  - Replication through HDFS
  - Erasure Coding (EC)



Basics of Erasure Coding

- Provides same level of fault-tolerance as 3X replication
- Uses less storage space - Overhead is not more than 50%
- 3X replication scheme add 200% overhead in storage and network bandwidth
- EC using striping - Redundant Array of Independent Disks (RAID)
- Calculates and stores parity cells for each stripe of original data cells
- Errors on any striping cell is recovered using a decoding calculation
- Recover is based on the surviving data and parity cells
- EC supports: Hive, MapReduce and Spark

Erasure Coding - Recovery

- **NameNode responsible for tracking any missing blocks**
- **NameNode assigns the task of recovering the blocks to DataNodes**
- **Client requests data and block missing**
  - Additional read requests are issued
  - Fetch the parity blocks
  - Decode data
- **Recovery task is passed as a heartbeat response to DataNodes**
- **Process is similar to how replicated blocks are recovered after failure**

Erasure Coding - Limitations

- Erasure Coding (EC) is set on a per-directory basis
- Erasure coding works only on new data written to a directory
- Existing files continue using the default replication scheme.
- Might impact the performance of a cluster due to consuming considerable CPU resources and network bandwidth
- It is recommended that you use erasure coding only for cold data
- Moving a file from a non-EC directory to an EC directory, or from an EC directory to a non-EC directory does NOT change the file's EC or replication strategy

Erasure Coding - Converting Files

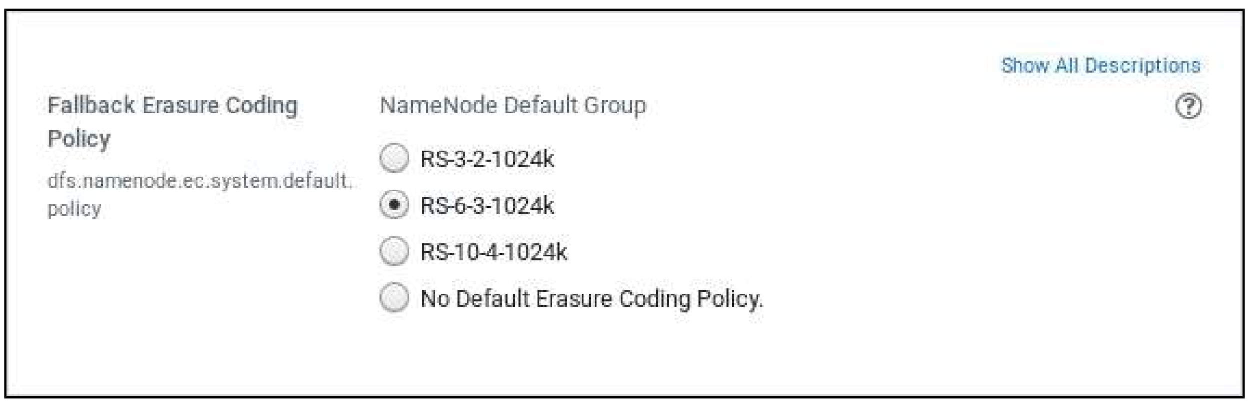
- **Setting an EC policy on a new/existing directory does not affect existing data**
- **Setting EC policy on a non-empty directory, does NOT convert existing files to use Erasure Coding**
- **To convert an existing file from non-EC to EC - copy the file into a directory that has an EC policy**
- **Use distcp to copy (convert) files**

HDFS Erasure Coding - Considerations

- **For fault-tolerance at the rack level:**
  - Important to have at least as many racks as the configured EC stripe width
  - Need minimum 9 racks
  - Recommended 10+ racks to handle planned/unplanned outages
- **For clusters with fewer racks than the stripe width, fault tolerance at the rack level cannot be maintained**

Implementing Erasure Coding

- **Understanding erasure coding policies**
  - Codec: The erasure codec that the policy uses. CDP currently supports Reed-Solomon (RS)
  - Number of Data Blocks: The number of data blocks per stripe
  - Number of Parity Blocks: The number of parity blocks per stripe
  - Cell Size: The size of one basic unit of striped data
  - For example, a RS-6-3-1024k policy
    - Codec: Reed-Solomon
    - Number of Data Blocks: 6
    - Number of Parity Blocks: 3
    - Cell Size: 1024k



Planning for Erasure Coding

- **Before enabling erasure coding on your data:**
  - Note the limitations for EC
  - Determine which EC policy you want to use
  - Determine if you want to use EC for existing data or new data
  - If you want to use EC for existing data, you need to replicate that data with distcp or BDR
  - Verify that your cluster setup meets the rack and node requirements

Chapter Topics

Advanced Cluster Configuration

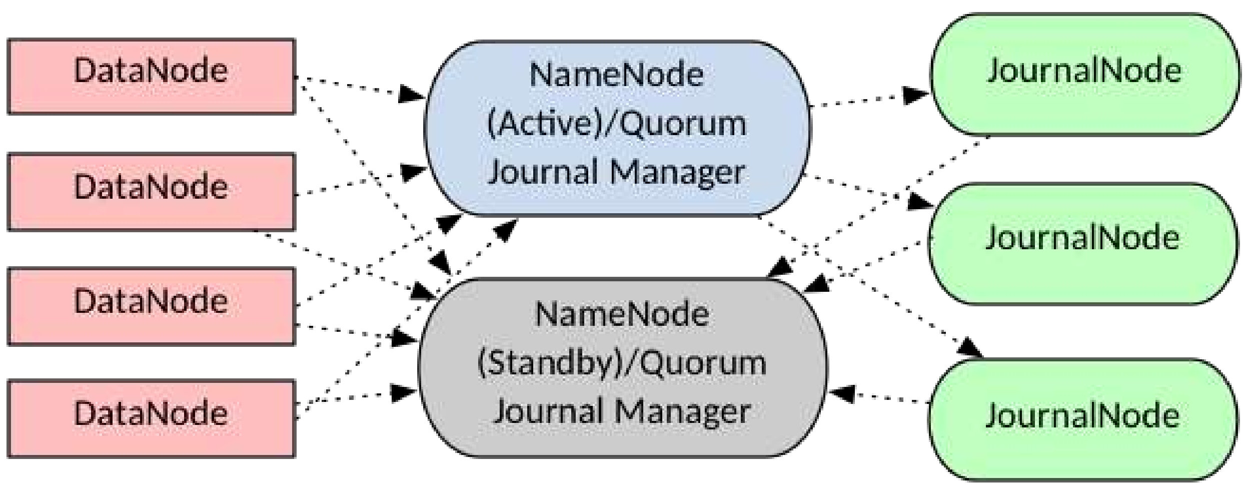
- Configuring Service Ports
- Tuning HDFS and MapReduce
- Managing Cluster Growth
- Erasure Coding
- **Enabling High Availability for HDFS and YARN**
- Essential Points
- Hands-On Exercise: Configuring HDFS for High Availability

HDFS High Availability Overview

- **A single NameNode is a single point of failure**
- **Two ways a NameNode can result in HDFS downtime**
  - Unexpected NameNode crash (rare)
  - Planned maintenance of NameNode (more common)
- **HDFS High Availability (HA) eliminates this SPOF**
- **Additional daemons in HDFS HA mode**
  - NameNode (active)
  - NameNode (standby)
  - Failover Controllers
  - Journal Nodes
- **No Secondary NameNode when HDFS High Availability is enabled**
  - Standby NameNode performs checkpointing

HDFS High Availability Architecture (1)

- **HDFS High Availability uses a pair of NameNodes**
  - One *active* and one *standby*
  - Clients only contact the active NameNode
  - DataNodes send heartbeats to both NameNodes
  - Active NameNode writes metadata changes to a quorum of JournalNodes
    - With HA, edits log is kept on JournalNodes instead of NameNode
  - Standby NameNode reads from the JournalNodes to stay in sync with the active NameNode



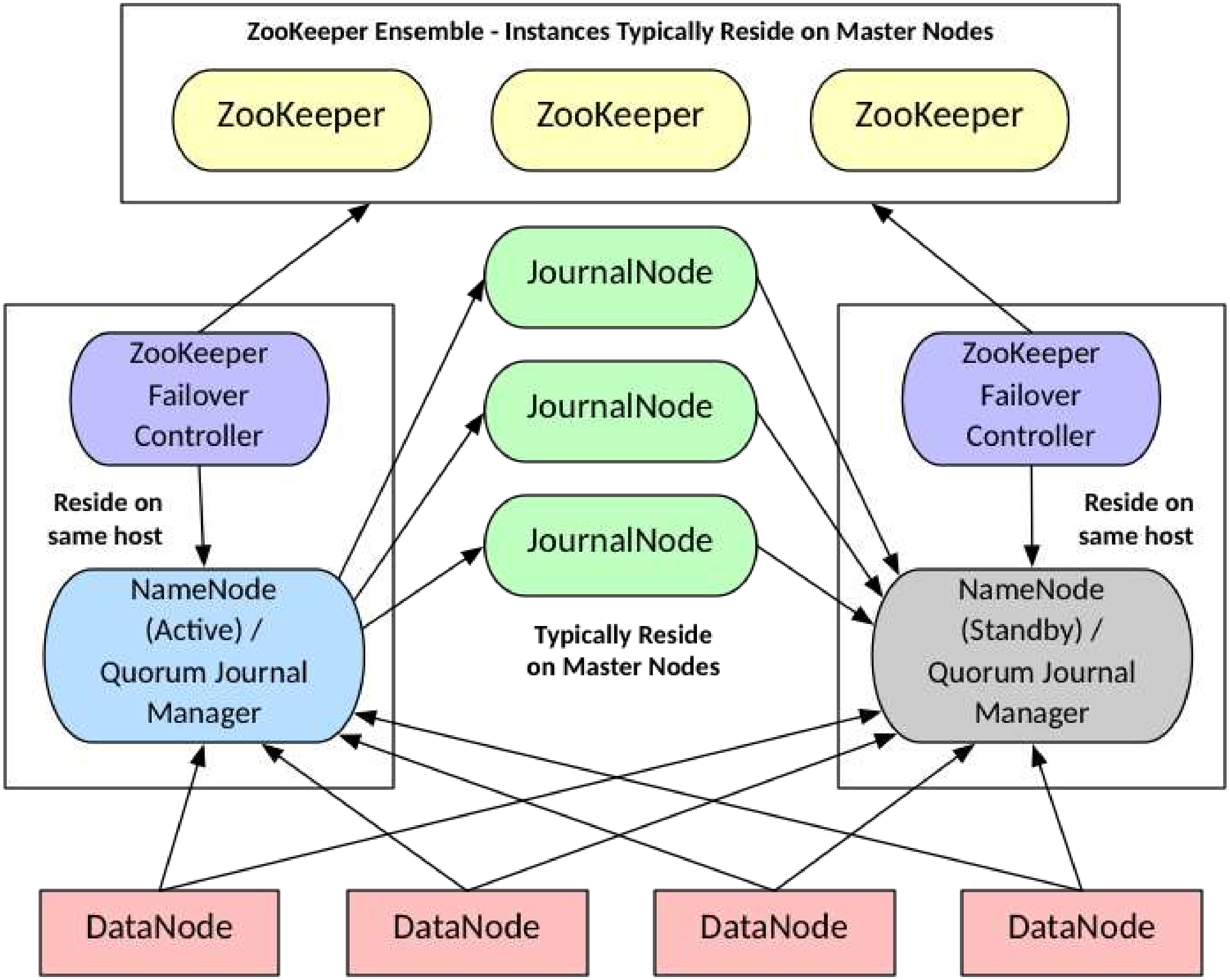
HDFS High Availability Architecture (2)

- **Active NameNode writes to local edits directories on the JournalNodes**
  - Managed by the *Quorum Journal Manager (QJM)*
    - Built in to NameNode
  - Waits for a success acknowledgment from the majority of JournalNodes
    - A single crashed or lagging JournalNode will not impact NameNode latency
  - Uses the *Paxos* algorithm to ensure reliability even if edits are being written as a JournalNode fails

Failover

- **Only one NameNode is active at any given time**
  - The other is in standby mode
- **The standby maintains a copy of the active NameNode's state**
  - So it can take over when the active NameNode goes down
- **Two types of failover**
  - Manual
    - Detected and initiated by an administrator
  - Automatic
    - A ZooKeeper Failover Controller (ZKFC) daemon runs on each NameNode host
    - Initiates failover when it detects Active NameNode failure
- **Cloudera Manager enables automatic failover by default**
  - `dfs.ha.automatic-failover.enabled`

HDFS HA With Automatic Failover



Enabling HDFS HA with Cloudera Manager

- **Ensure the ZooKeeper service is installed and enabled for HDFS**
- **From the HDFS Instances page, run the** Enable High Availability **wizard**
  - Specify the hosts for the two NameNodes and the JournalNodes
  - Specify the JournalNode edits directory for each host
  - The wizard performs the necessary steps to enable HA
    - Including the creation and configuration of new Hadoop daemons

After Enabling HDFS HA

- **After enabling HDFS HA, some manual configurations may be needed**
  - For Hive
    - Update Hive Metastore nodes
    - Consult the Cloudera documentation for details
  - For Impala
    - Run `INVALIDATE METADATA` command from the Impala shell after updating Hive metastore
  - For Hue
    - Add the **HttpFS** role (if not already on the cluster)

YARN High Availability

- **ResourceManager High Availablity removes a single point of failure**
  - Active-standby ResourceManager pair
  - Automatic failover option
- **Protects against significant performance effects on running applications**
  - Machine crashes
  - Planned maintenance events on the ResourceManager host machine
- **If failover to the standby occurs**
  - In-flight YARN applications resume from the last saved state store
- **To enable**
  - From the Cloudera Manager YARN page, select **Actions** > **Enable High Availability**, and select the host for the standby ResourceManager

High Availability Options

- **Not all Hadoop components currently support high availability configurations**
- **Some currently SPOF components can be configured to restart automatically in the event of a failure - Auto-Restart**
  - Hive Metastore
  - Impala Catalog
  - Impala Statestore
  - Spark Job History Server
  - YARN Job History Server
- **Services that do allow for HA:**
  - Cloudera Manager Server (load balancer)
  - HBase Master
  - Hue (load balancer)
  - Impalad (load balancer)
- **Several components have external databases - consider availability**

Chapter Topics

Advanced Cluster Configuration

- Configuring Service Ports
- Tuning HDFS and MapReduce
- Managing Cluster Growth
- Erasure Coding
- Enabling High Availability for HDFS and YARN
- **Essential Points**
- Hands-On Exercise: Configuring HDFS for High Availability

Essential Points

- **Configuring port numbers is available**
- **Cluster growth can be managed by adding hosts and adjusting configuration**
- **Erasure Coding provides the same level of fault-tolerance as 3x replication using less space, but requires a larger cluster**
- **HDFS advanced configuration properties can improve performance**
- **Service daemons use network ports to serve UI applications and communicate with other daemons**
  - Manage daemon port assignments in Cloudera Manager
- **HDFS can be configured for high availability with automatic failover capability**
  - Tune NameNode, DataNode, and file compression settings

Chapter Topics

Advanced Cluster Configuration

- Configuring Service Ports
- Tuning HDFS and MapReduce
- Managing Cluster Growth
- Erasure Coding
- Enabling High Availability for HDFS and YARN
- Essential Points
- **Hands-On Exercise: Configuring HDFS for High Availability**

Hands-On Exercise: Configuring HDFS for High Availability

- **In this exercise, you will configure your Hadoop cluster for HDFS high availability**
- **Please refer to the Hands-On Exercise Manual for instructions**
- **Cluster deployment after exercise completion (only a subset of the daemons on the cluster are shown):**

	master-1	master-2	worker-1	worker-2	worker-3	cmhost
HDFS NameNode	✓					
HDFS Secondary NameNode		✓				
HttpFS	✓					
HDFS Balancer	✓					
HDFS DataNode			✓	✓	✓	
YARN Resource Manager	✓					
YARN JobHistory Server	✓					
YARN NodeManager			✓	✓	✓	
Zookeeper Server	✓	✓	✓			
Sqoop	✓					
Cloudera Manager Server						✓
Cloudera Manager Server Database						✓
Cloudera Management Services						✓
Cloudera Manager Agent	✓	✓	✓	✓	✓	✓

Cluster Maintenance

Chapter 13

Course Chapters

- Introduction
- Cloudera Data Platform
- CDP Private Cloud Base Installation
- Cluster Configuration
- Data Storage
- Data Ingest
- Data Flow
- Data Access and Discovery
- Data Compute
- Managing Resources
- Planning Your Cluster
- Advanced Cluster Configuration
- **Cluster Maintenance**
- Cluster Monitoring
- Cluster Troubleshooting
- Security
- Private Cloud / Public Cloud
- Conclusion
- Appendix: Cloudera Manager API
- Appendix: Ozone Overview

Cluster Maintenance

After completing this chapter, you will be able to

- **Check status of HDFS**
- **Copy data between clusters**
- **Rebalance data on the cluster**
- **Take HDFS snapshots**
- **Maintaining Hosts**
- **Upgrading a Cluster**

Chapter Topics

Cluster Maintenance

- **Checking HDFS Status**
- Copying Data Between Clusters
- Rebalancing Data in HDFS
- HDFS Directory Snapshots
- Hands-On Exercise: Creating and Using a Snapshot
- Host Maintenance
- Upgrading a Cluster
- Essential Points
- Hands-On Exercise: Upgrade the Cluster

Checking for Corruption in HDFS (1)

- **`hdfs fsck` checks for missing or corrupt data blocks**
  - Unlike Linux `fsck`, does not attempt to repair errors
- **Can be configured to list all files**
  - Also all blocks for each file, all block locations, all racks
- **Examples**

```
$ hdfs fsck /

$ hdfs fsck / -files

$ hdfs fsck / -files -blocks

$ hdfs fsck / -files -blocks -locations

$ hdfs fsck / -files -blocks -locations -racks
```

Checking for Corruption in HDFS (2)

- **Good idea to run `hdfs fsck` regularly**
  - Choose a low-usage time to run the check
- **`move` option moves corrupted files to `/lost+found`**
  - A corrupted file is one where all replicas of a block are missing
- **`delete` option deletes corrupted files**

The `hdfs dfsadmin` Command

- **A tool for performing administrative operations on HDFS**
- **Sample commands**
  - Get safemode status

```
$ hdfs dfsadmin -safemode get
```

  - Perform a NameNode metadata backup (must be in safemode)

```
$ hdfs dfsadmin -fetchImage fsimage.backup
```

  - Set a quota on the storage in a specific HDFS directory

```
$ hdfs dfsadmin -setSpaceQuota n-bytes somedir
```

Using `dfsadmin` (1)

- **The `hdfs dfsadmin` command provides a number of useful administrative features, such as**
  - List information about HDFS on a per-datanode basis

```
$ hdfs dfsadmin -report
```

Using `dfsadmin` (2)

- **Manage *safe mode***
  - Read-only—no changes can be made to the metadata
  - Does not replicate or delete blocks
- **HDFS starts up in safe mode automatically**
  - Leaves safe mode when the (configured) minimum percentage of blocks satisfy the minimum replication condition
- **Use `dfsadmin` to start or leave safe mode manually**

```
$ hdfs dfsadmin -safemode enter
```

```
$ hdfs dfsadmin -safemode leave
```

- **Use `dfsadmin wait` in scripts to wait until safe mode is exited**
  - Blocks until HDFS is no longer in safe mode

```
$ hdfs dfsadmin -safemode wait
```

Using `dfsadmin`

- **Save the NameNode metadata to disk and reset the edit log**
  - HDFS must be in safe mode

```
$ hdfs dfsadmin -saveNamespace
```

Other HDFS Command Line Capabilities

- **The `hdfs` command provides many additional features, such as the ability to**
  - Change file owner, group, and permissions
  - Show used and available storage
  - Change file replication levels
  - Detect and repair file system corruption
  - Review and manage configuration properties
- **Use `-help` to see all options for `hdfs` or its subcommands.**

```
$ hdfs -help
```

```
$ hdfs dfs -help
```

Chapter Topics

Cluster Maintenance

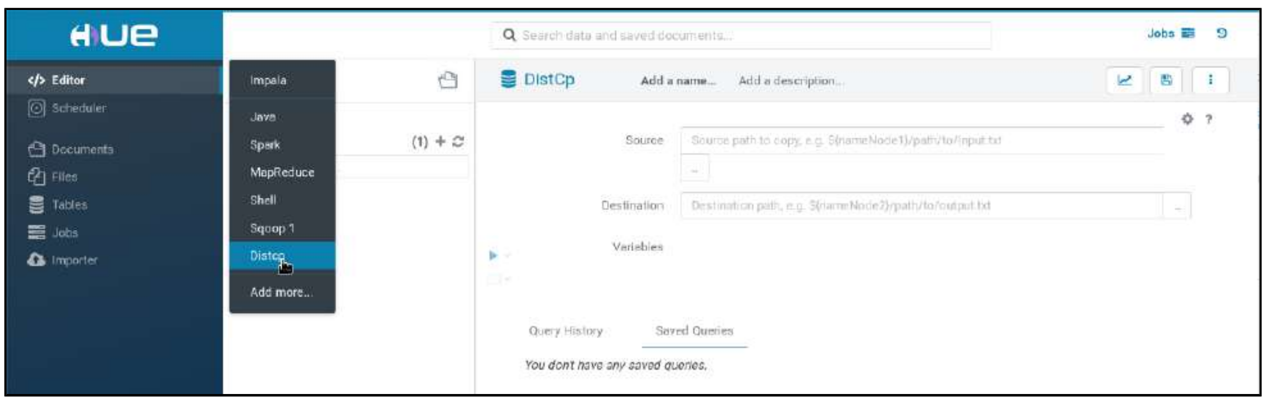
- Checking HDFS Status
- **Copying Data Between Clusters**
- Rebalancing Data in HDFS
- HDFS Directory Snapshots
- Hands-On Exercise: Creating and Using a Snapshot
- Host Maintenance
- Upgrading a Cluster
- Essential Points
- Hands-On Exercise: Upgrade the Cluster

Copying Data

- **Hadoop clusters can hold massive amounts of data**
- **A frequent requirement is to back up the cluster for disaster recovery**
- **Ultimately, this is not a Hadoop problem!**
  - It's a "managing huge amounts of data" problem
- **Cluster could be backed up to tape or other medium if necessary**
  - Custom software may be needed

Copying Data with `distcp`

- **`distcp` copies data within a cluster, or between clusters**
  - Used to copy large amounts of data
  - Turns the copy process into a MapReduce job
- **Copies files or entire directories**
  - Files previously copied will be skipped
    - Note that the only check for duplicate files is that the file's name, size, and checksum are identical
- **Can use the `DISTCP` command from Hue**



distcp Examples (1)

- **Copy data from one cluster to another**

```
$ hadoop distcp \
  hdfs://cluster1_nn:8020/path/to/src \
  hdfs://cluster2_nn:8020/path/to/destination
```

- **Copy data within the same cluster**

```
$ hadoop distcp /path/to/source /path/to/destination
```

`distcp` Examples (2)

- **Copy data from one cluster to another when the clusters are running different versions of Hadoop**
  - HA HDFS example using HttpFS

```
$ hadoop distcp \
  hdfs://cluster-nn:8020/path/to/src \
  webhdfs://httpfs-server:14000/path/to/dest
```

  - Non-HA HDFS example using WebHDFS

```
$ hadoop distcp \
  hdfs://cluster1_nn:8020/path/to/src \
  webhdfs://cluster2_nn:9870/path/to/dest
```

Best Practices for Copying Data

- **In practice, many organizations do not copy all their data between clusters**
- **Instead, they write their data to two clusters as it is being imported**
  - This is often more efficient
  - Not necessary to run all jobs that modify or save data on the backup cluster
    - As long as the source data is available, all derived data can be regenerated later

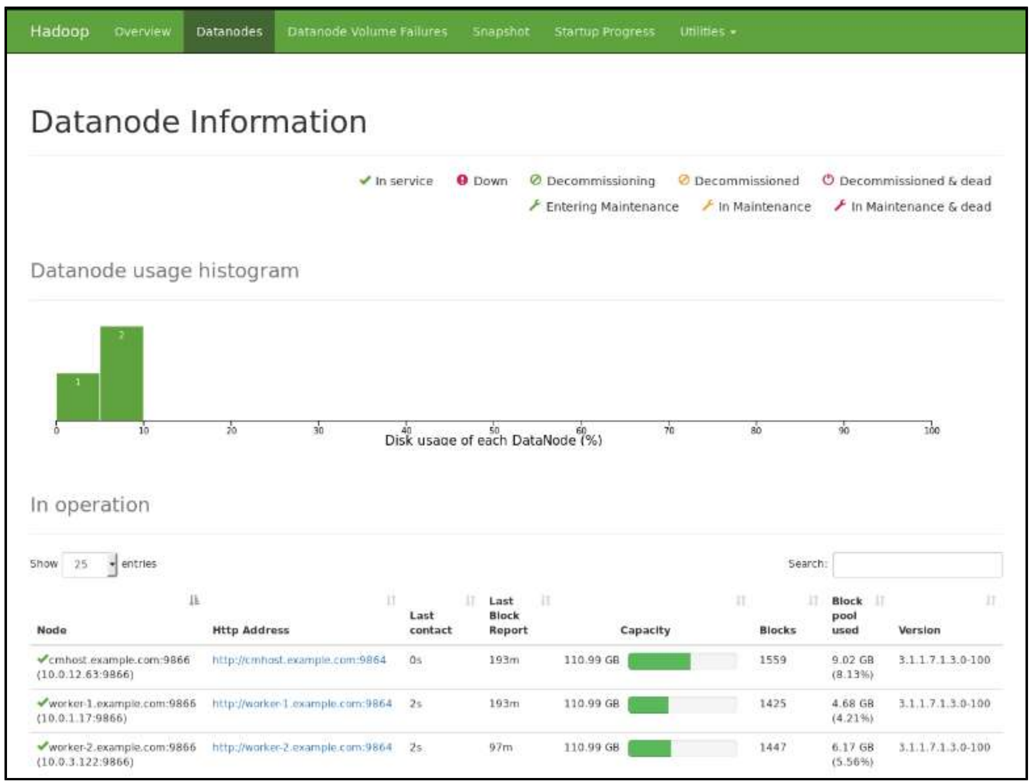
Chapter Topics

Cluster Maintenance

- Checking HDFS Status
- Copying Data Between Clusters
- **Rebalancing Data in HDFS**
- HDFS Directory Snapshots
- Hands-On Exercise: Creating and Using a Snapshot
- Host Maintenance
- Upgrading a Cluster
- Essential Points
- Hands-On Exercise: Upgrade the Cluster

Cluster Rebalancing (1)

- **HDFS DataNodes on cluster can become unbalanced**
  - Some nodes have much more data on them than others
  - Such as when a new host is added to the cluster
  - You can view the capacities from the NameNode GUI DataNode tab



Cluster Rebalancing (2)

- **Balancer adjusts blocks to ensure all nodes are within the "Rebalancing Threshold"**
- **"Used space to total capacity" ratio *on each DataNode* will be brought to within the threshold of ratio *on the cluster***
- **Balancer does not balance between individual volumes on a single DN**
- **Configure bandwidth usage with `dfs.datanode.balance.bandwidthPerSec`**
  - Default: 10MB
  - Recommendation: approximately 10% of network speed
    - For a 1 Gbps network, set values to 100MB/sec
  - Balancer bandwidth can be set temporarily for the current session

```
$ dfsadmin -setBalancerBandwidth bits-per-second
```

When To Rebalance

- **Balancer does not run automatically, even when the rebalance threshold is exceeded**
  - The balancer must be run manually
  - Run from Cloudera Manager (HDFS **Actions** menu) or command line

```
$ sudo -u hdfs hdfs balancer
```

- **Rebalance immediately after adding new nodes to the cluster**
- **Rebalance during non-peak usage times**
  - Rebalancing does not interfere with running services and applications
  - However, it does use bandwidth

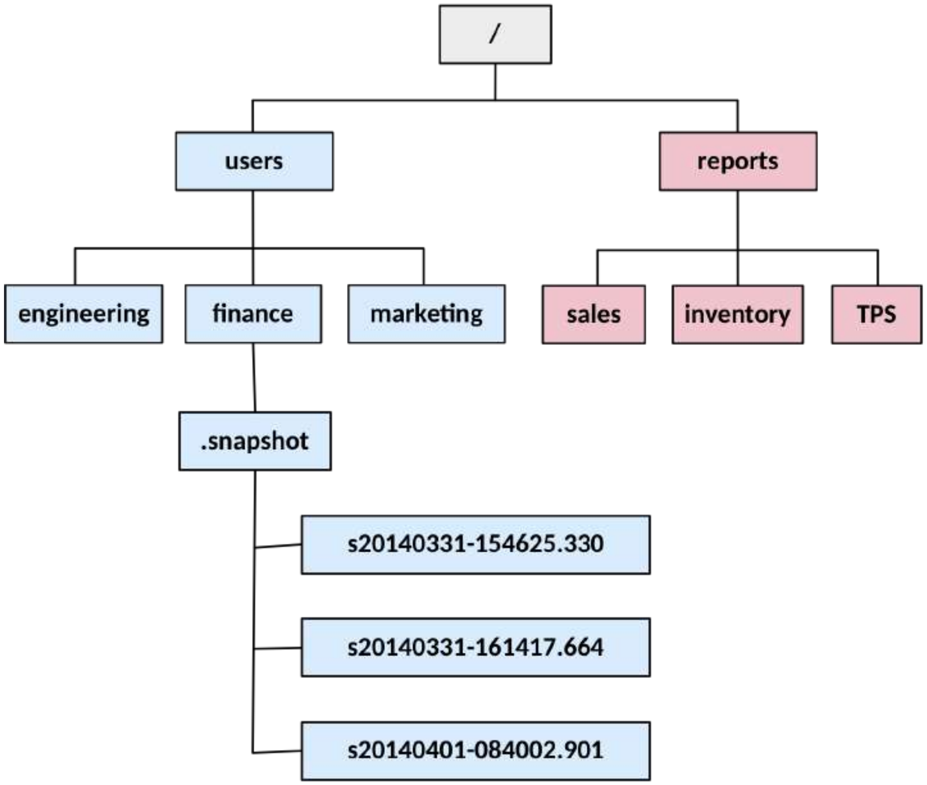
Chapter Topics

Cluster Maintenance

- Checking HDFS Status
- Copying Data Between Clusters
- Rebalancing Data in HDFS
- **HDFS Directory Snapshots**
- Hands-On Exercise: Creating and Using a Snapshot
- Host Maintenance
- Upgrading a Cluster
- Essential Points
- Hands-On Exercise: Upgrade the Cluster

Snapshots

- **A Snapshot is a read-only copy of an HDFS directory at a point in time**
  - Useful for data backup, disaster recovery
  - You can also snapshot the entire HDFS filesystem
- **Snapshots appear on the filesystem as read-only directories**
  - Data is not copied
  - Snapshot notes the list of blocks
- **Snapshots can be deleted**



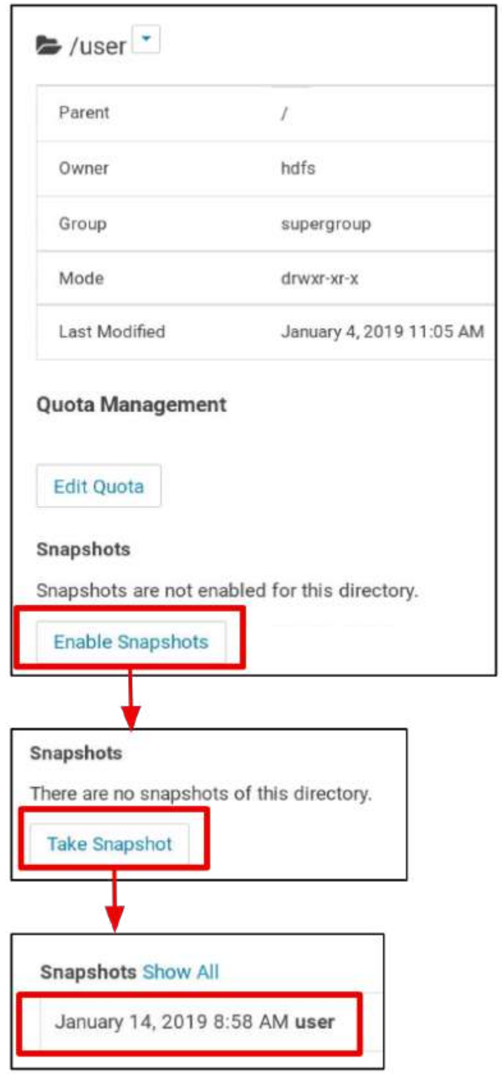
HBase and HDFS Snapshots

- You can create HBase and HDFS snapshots
- HBase snapshots allow you to create point-in-time backups of tables without making data copies
- HDFS snapshots allow you to create point-in-time backups of directories or the entire filesystem without actually cloning the data
- Can improve data replication performance
- Prevent errors caused by changes to a source directory
- Snapshots appear on the filesystem as read-only directories

Enabling and Taking Snapshots

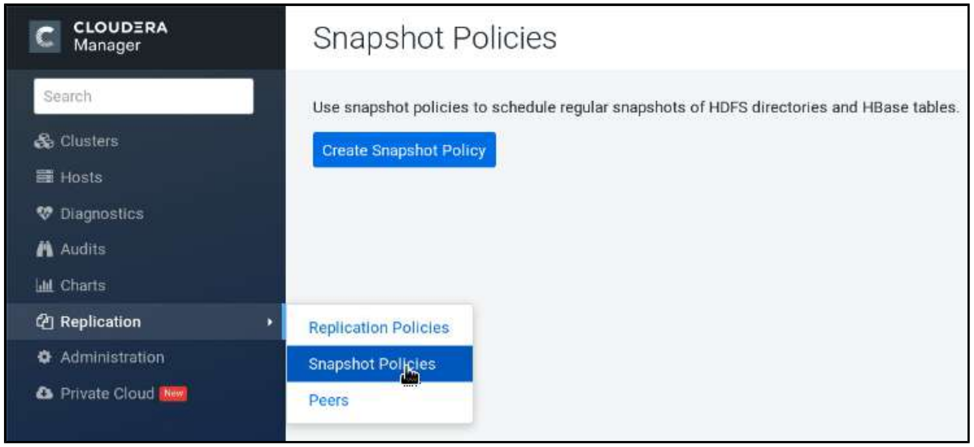
- **Enable snapshotting for an HDFS directory for snapshotting in Cloudera Manager's HDFS File Browser tab**

- **After snapshotting is enabled you can take a snapshot**



Snapshot Policies (1)

- **Cloudera Manager allows you to create snapshot policies**
  - To manage snapshot policies go to **Replication** > **Snapshot Policies**



- **Snapshot policies define**
  - HDFS directories to be snapshotted
  - Intervals at which snapshots should be taken
  - Number of snapshots to keep for each snapshot interval
- **Example policy**
  - Take snapshots daily and retain for seven days
  - Take snapshots weekly and retain for four weeks

Snapshot Policies (2)

- **Option to configure alerts on snapshot attempts**
  - For example: send an alert of the snapshot attempt failed
- **Managing snapshots**
  - If the snapshot policy includes a limit on the number of snapshots to keep, Cloudera Manager deletes older snapshots as needed
  - If you edit or delete a snapshot policy
    - Files or directories previously included in the policy may leave orphaned snapshots
    - These must be deleted manually
- **Avoid orphaned snapshots by deleting these snapshots *before* editing or deleting the associated snapshot policy**

Chapter Topics

Cluster Maintenance

- Checking HDFS Status
- Copying Data Between Clusters
- Rebalancing Data in HDFS
- HDFS Directory Snapshots
- **Hands-On Exercise: Creating and Using a Snapshot**
- Host Maintenance
- Upgrading a Cluster
- Essential Points
- Hands-On Exercise: Upgrade the Cluster

Hands-On Exercise: Creating and Using a Snapshot

- **In this exercise, you will learn to take and compare HDFS snapshots as well as configure a snapshot policy**
- **Please refer to the Hands-On Exercise Manual for instructions**

Chapter Topics

Cluster Maintenance

- Checking HDFS Status
- Copying Data Between Clusters
- Rebalancing Data in HDFS
- HDFS Directory Snapshots
- Hands-On Exercise: Creating and Using a Snapshot
- **Host Maintenance**
- Upgrading a Cluster
- Essential Points
- Hands-On Exercise: Upgrade the Cluster

Maintenance Mode

- **Allows you to suppress alerts for a host, service, role, or an entire cluster**
- **Maintenance mode does not prevent events from being logged**
- **If you set a service into maintenance mode, then its roles are put into effective maintenance mode**
- **If you set a host into maintenance mode, then any roles running on that host are put into effective maintenance mode**
- **To enter Maintenance Mode:**
  - In the left menu, click **Clusters > ClusterName**
  - In the cluster's Action menu, select **Enter Maintenance Mode**
  - Confirm that you want to do this
- **To enter Maintenance Mode for a service/role select Enter Maintenance Mode from the Action menu of that service/role**
- **To exit Maintenance Mode, select Exit Maintenance Mode from the Action menu of the entity**

View Host Status

- **View summary information about the hosts managed by Cloudera Manager**
  - Click **All Hosts (Hosts menu)** in the left menu
  - The information provided varies depending on which columns are selected
  - To change the columns, click the **Columns: n Selected** drop-down
  - Utilize the **Filters** section at the left of the page
- **Viewing the Hosts in a Cluster**
  - Select **Clusters > Cluster name > Hosts**
  - In the Home screen, click **Hosts**
- **You can view detailed information about an individual host by clicking the host's link**

View Host Role Assignments

- **You can view the assignment of roles to hosts as follows:**
  - In the left menu, click **Hosts > Roles**

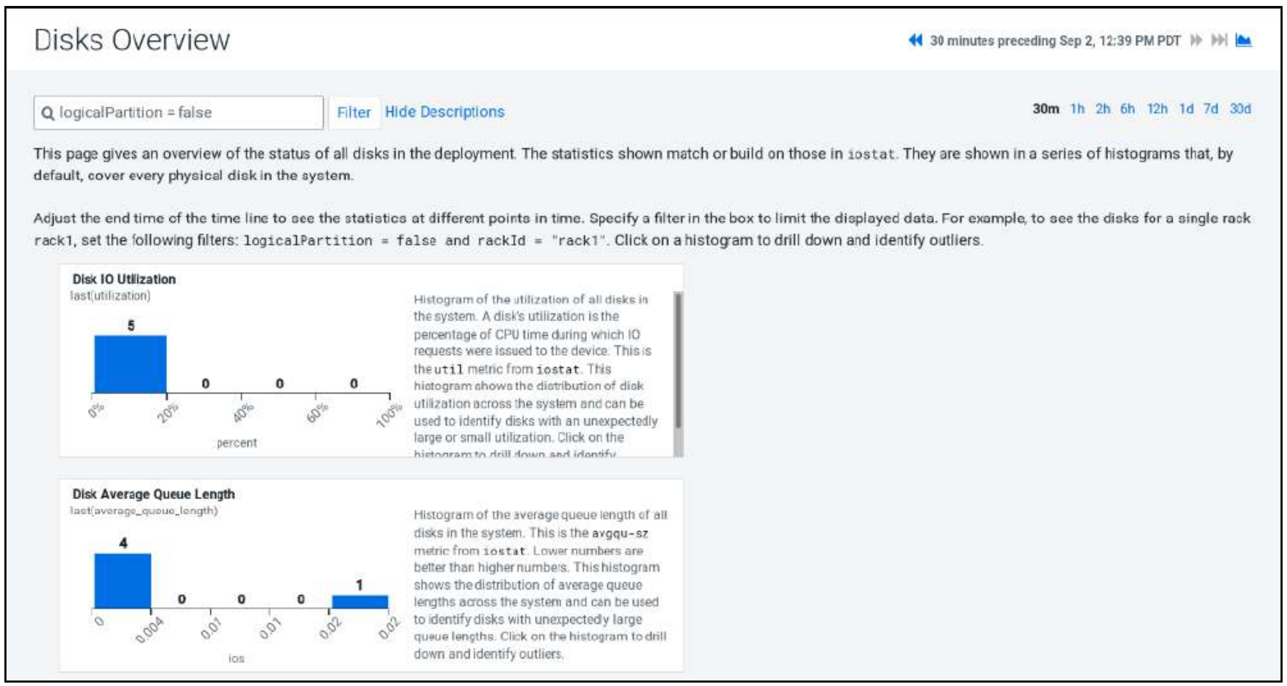
Roles

This table is grouped by hosts having the same roles assigned to them.

Hosts	Count	Roles
cmhost.example.com	1	DN, G, HMS, LB, HS, AP, ES, HM, RM, SM, G, NM, S
master-1.example.com	1	B, HFS, NN, G, G, JHS, RM
master-2.example.com	1	SNN, G, HMS, HS2, ICS, ISS, OS, YA..., YA..., HS
worker-[1-2].example.com	2	DN, G, ID, NM

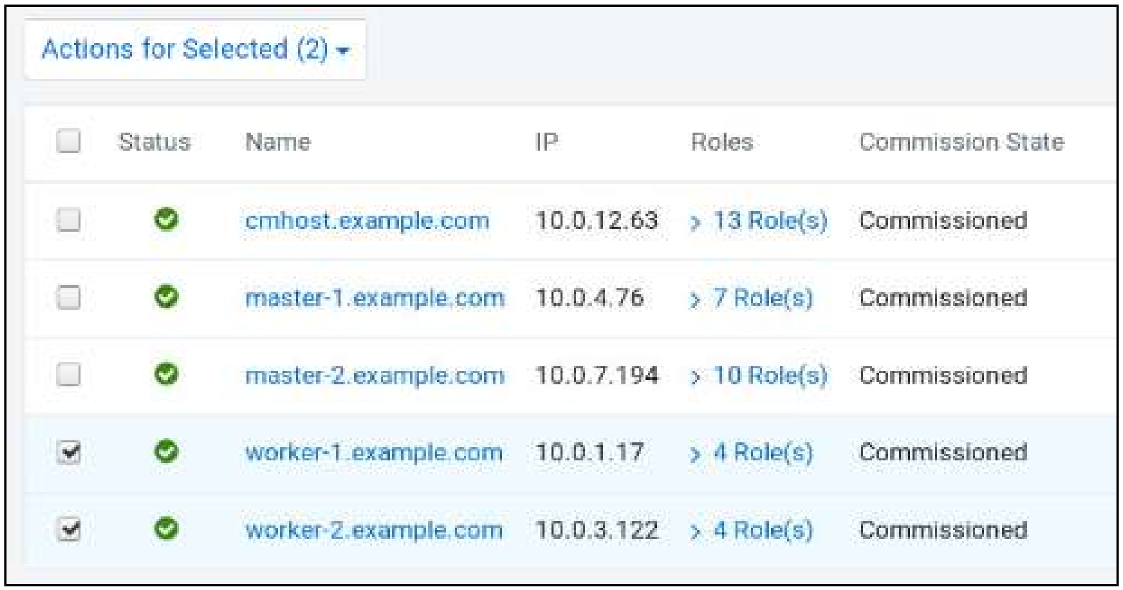
Host Disks Overview

- **View the status of all disks in a cluster:**
  - Click **Hosts > Disks Overview** to display an overview of the status of all disks
  - The statistics exposed and are shown in a series of histograms that by default cover every physical disk in the system
  - Adjust endpoints of the time line to see statistics for different time periods
  - Specify a filter in the box to limit the displayed data



Start or Stop All Roles on Host

- **You can Start/Stop all the roles on a host from the Hosts page**
  - Click the **Hosts** tab
  - Select one or more hosts
  - Select **Actions for Selected** > **Start/Stop Roles on Hosts**


Actions for Selected (2)

	Status	Name	IP	Roles	Commission State
☐		cmhost.example.com	10.0.12.63	13 Role(s)	Commissioned
☐		master-1.example.com	10.0.4.76	7 Role(s)	Commissioned
☐		master-2.example.com	10.0.7.194	10 Role(s)	Commissioned
☑		worker-1.example.com	10.0.1.17	4 Role(s)	Commissioned
☑		worker-2.example.com	10.0.3.122	4 Role(s)	Commissioned

Changing Host Names

- **You may need to update the names of the hosts**
  - The process requires Cloudera Manager and cluster downtime
  - Any user-created scripts reference specific hostnames must also be updated
  - Changing cluster hostnames is not recommended by Cloudera
- **Be very careful in creating a naming scheme for your servers**
- **Changing names is a very complex and lengthy process, especially in a Kerberos enabled cluster**

Moving a Host between Clusters

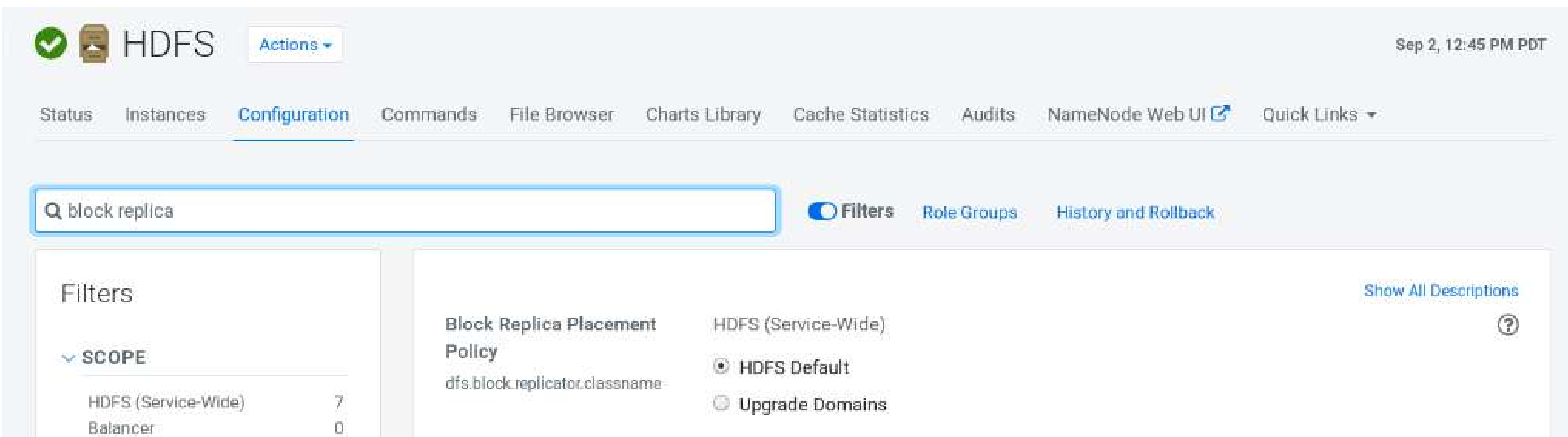
- **To move a host between clusters:**
  - Decommission the host
  - Remove all roles from the host (except for the Cloudera Manager management roles)
  - Remove the host from the cluster but leave it available to Cloudera Manager
  - Add the host to the new cluster
  - Add roles to the host (use templates)

Upgrade Domains (1)

- Upgrade Domains allow the grouping of cluster hosts for optimal performance during restarts and upgrades
- Upgrade Domains enable faster cluster restart
- Faster Cloudera Runtime upgrades
- Seamless OS patching and hardware upgrades across large clusters
- An alternative to the default HDFS block placement policy
- Select Upgrade Domains as the block placement policy
- Assign an Upgrade Domain group to each DataNode host
- Useful for very large clusters, or for clusters where rolling restarts happen frequently
- Example: if HDFS is configured with the default replication factor of 3, the NameNode places the replica blocks on DataNode hosts in 3 different Upgrade Domains and on at least two different racks

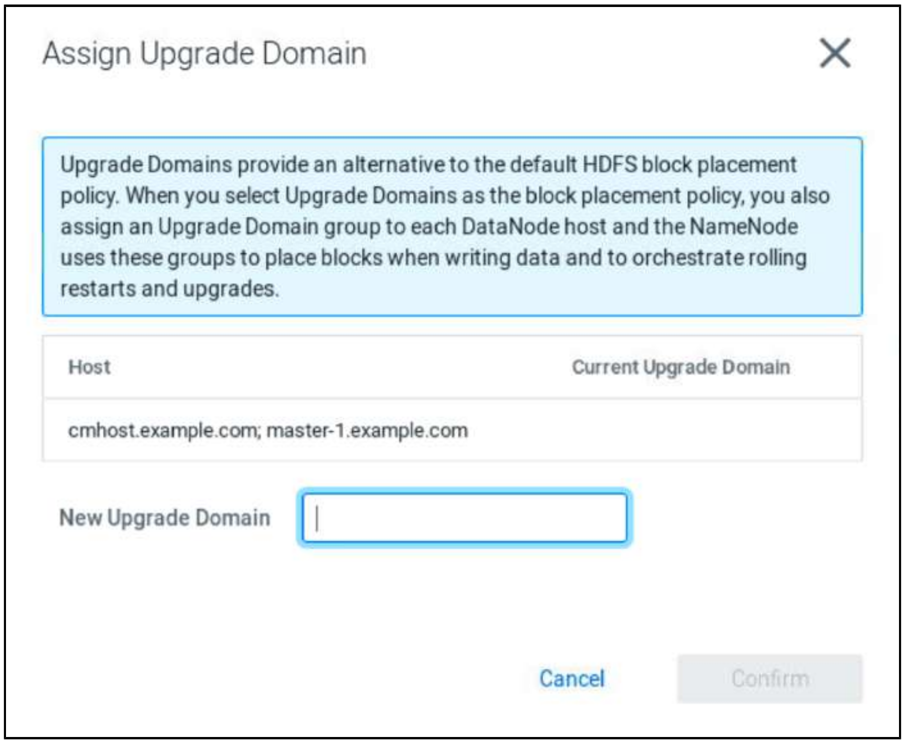
Upgrade Domains (2)

- **Configure the Upgrade Domains for all hosts**
- **Set the HDFS Block Replica Placement Policy:**
  - Go to the HDFS service Status page
  - Click the Configuration tab
  - Search for HDFS Block Replica Placement Policy parameter
  - Select Upgrade Domains and Save Changes



Configuring Upgrade Domains

- **Steps to configure Upgrade Domains:**
  - Click *Hosts > All Hosts*
  - Select the hosts you want to add to an Upgrade Domain
  - Click *Actions for Selected > Assign Upgrade Domain*
  - Enter the name of the Upgrade Domain in the *New Upgrade Domain* field
  - Click the *Confirm* button



Chapter Topics

Cluster Maintenance

General Upgrade Process

- **Cloudera release numbering example: CDP 7.1.1**
  - **7** = major version
  - **1** = minor update
  - **1** = maintenance update
- **Cloudera recommends upgrading when a new version update is released**
- **Upgrade installations can use parcels or packages**
  - Parcels installed by Cloudera Manager
  - Package installation is manual
- **Cloudera Manager minor version and Cloudera Runtime version do not need to match, but check the documentation to confirm compatibility between version**

Maintenance Release Upgrade—General Procedures (1)

- **Upgrading to a new maintenance release**
  - Such as CDP 7.1.0 to CDP 7.1.1
- **Before Upgrading CDP—general procedure**
  1. Back up key service data such as metastore database, NameNode and DataNode configuration, and ZooKeeper data
  2. Run the Host Inspector (fix any issues)
  3. Run the Security Inspector (fix any issues)
  4. Run `hdfs fsck /` and `hdfs dfsadmin -report` (fix any issues)
  5. Reserve a maintenance window
  6. Enable maintenance mode before starting the upgrade
     - Avoids alerts during the upgrade
     - Remember to exit maintenance mode when upgrade is complete
- **Run the Upgrade Cluster wizard**

Maintenance Release Upgrade—General Procedures (2)

- **If using parcels, the wizard *automatically* completes the necessary steps**
  1. Confirms parcel availability
  2. Downloads and distributes parcel to nodes
  3. Shuts down services
  4. Activates new parcel
  5. Upgrades services as necessary
  6. Deploys client configuration files
  7. Restarts services
  8. Runs the host inspector
  9. Reports results of the upgrade
- **If using packages**
  - Manually create the needed repository file pointing to the CDP software
  - Manually install needed CDP packages (using yum, apt-get, or zypper)
  - Run the upgrade wizard in Cloudera Manager

Minor Release Upgrade—General Procedures

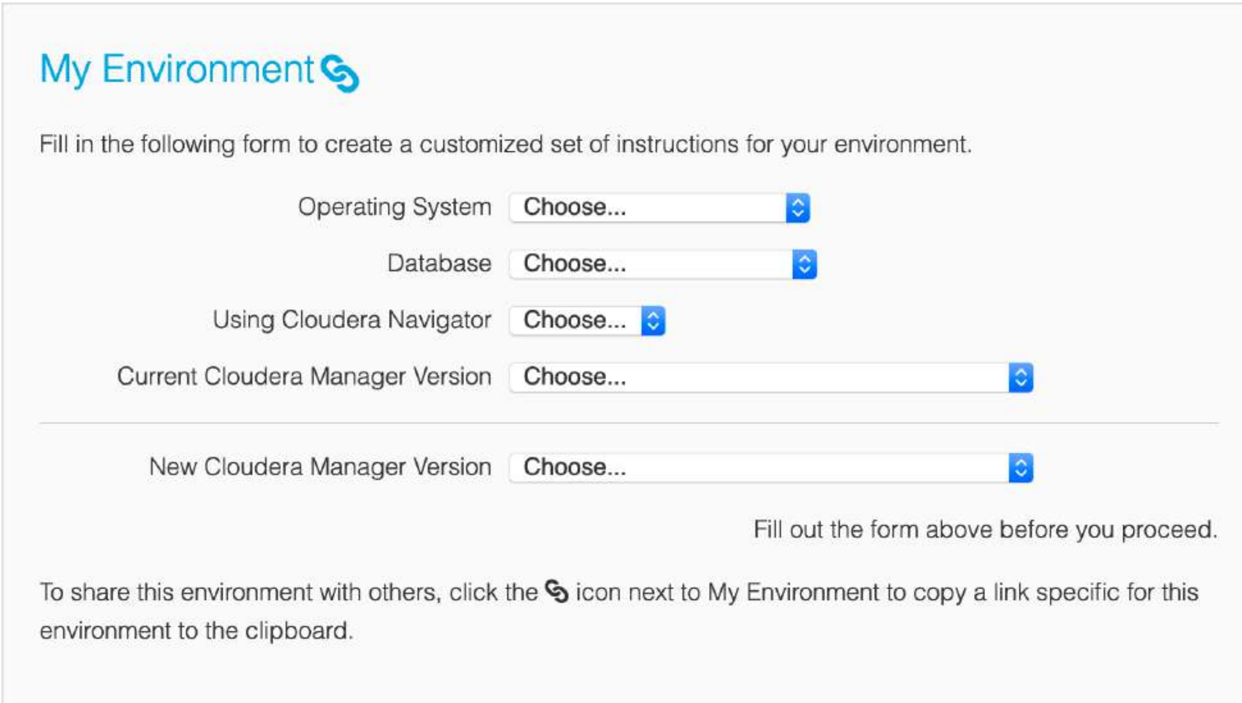
- **Minor release upgrade**
  - For example, from CDP 7.0 to CDP 7.1
- **Same as for a maintenance release, but with some additional steps**
  - After enabling maintenance mode
    - Stop cluster services
    - Back up the NameNode’s HDFS metadata
  - After running the upgrade wizard
    - If using packages, remove old CDP version packages
    - Finalize the HDFS metadata upgrade (button on Cloudera Manager’s NameNode page)
- **The above provides a general overview of the software upgrade process**
  - See the CDP documentation for details for upgrading from one specific release to another

Rolling Upgrades with Cloudera Manager

- **Cloudera Manager allows you to upgrade and restart upgraded services with no downtime**
  - Upgrade using parcels
  - Requires HDFS be configured for high availability
  - Supports minor version upgrades only
- **General procedure**
  - Download, distribute, and activate new parcel
  - Do a rolling restart on individual services or entire cluster

Upgrading Cloudera Manager

- **The Cloudera documentation provides customisable instructions**



Establish Access to the Software (1)

- **Cloudera Manager needs access to a package repository that contains the updated software packages**
  1. Log in to the Cloudera Manager Server host
  2. Remove any older files in the existing repository directory

```
$ sudo rm /etc/yum.repos.d/cloudera*manager.repo*
```

  3. Create a file named /etc/yum.repos.d/cloudera-manager.repo with the following content

```
[cloudera-manager]
# Packages for Cloudera Manager
name=Cloudera Manager
baseurl=https://archive.cloudera.com/p/cm7/7.1.2/redhat7/
yum/
gpgkey=https://archive.cloudera.com/p/cm7/7.1.2/redhat7/
yum/RPM-GPG-KEY-cloudera
gpgcheck=1
```

Establish Access to the Software (2)

- A Cloudera Manager upgrade can introduce new package dependencies
- Your organization may have restrictions or require prior approval for installation of new packages
- You can determine which packages may be installed or upgraded by running the following command
  `yum deplist cloudera-manager-agent`

Upgrade the Cloudera Manager Server (1)

1. Stop the Cloudera Management Service
   a. Log in to the Cloudera Manager Admin Console
   b. Select Clusters > Cloudera Management Service
   c. Select Actions > Stop
2. Ensure that you have disabled any scheduled replication or snapshot jobs
3. Wait for any running commands to complete

Upgrade the Cloudera Manager Server (2)

4. **Stop the Cloudera Manager Server and Agent on the host(s) running Cloudera Manager**
   a. **Log in to the Cloudera Manager Server host**
   b. **Stop the Cloudera Manager Server**

```
$ sudo systemctl stop cloudera-scm-server
```

   c. **Stop the Cloudera Manager Agent**

```
$ sudo systemctl stop cloudera-scm-agent
```

5. **Upgrade the packages**

```
$ sudo yum clean all
```

```
$ sudo yum upgrade cloudera-manager-server cloudera-manager-
daemons cloudera-manager-agent
```

Upgrade the Cloudera Manager Server (3)

6. **You might be prompted about your configuration file version:**

```
Configuration file '/etc/cloudera-scm-agent/config.ini'
==> Modified (by you or by a script) since installation.
==> Package distributor has shipped an updated version.
What would you like to do about it ? Your options are:
Y or I : install the package maintainer's version
N or O : keep your currently-installed version
D : show the differences between the versions
Z : start a shell to examine the situation
The default action is to keep your current version.
```

You may receive a similar prompt for `/etc/cloudera-scm-server/db.properties`. Answer `N` to both prompts.

Upgrade the Cloudera Manager Server (4)

7. **Verify that you have the correct packages installed**

```
$ rpm -qa 'cloudera-manager-*'
```

8. **Start the Cloudera Manager Agent**

```
$ sudo systemctl start cloudera-scm-agent
```

9. **Start the Cloudera Manager Server.**

```
$ sudo systemctl start cloudera-scm-server
```

Upgrade the Cloudera Manager Agents

1. **Use a Web browser to open the Cloudera Manager Admin Console**

```
http://cloudera_Manager_server_hostname:7180/cmf/upgrade
```

2. **If the Cloudera Management Service is still running, stop it**
   a. **Log in to the Cloudera Manager Admin Console**
   b. **Select Cluster > Cloudera Management Service**
   c. **Select Actions > Stop**

3. **Click Upgrade Cloudera Manager Agent packages and follow the wizard steps**

Using a Cloudera Manager Template

- **You can create a new cluster by exporting a cluster template from an existing cluster**
- **Use cluster templates to:**
  - Duplicate clusters for use in developer, test, and production environments
  - Quickly create a cluster for a specific workload
  - Reproduce a production cluster for testing and debugging

Steps to Use a Template to Create a Cluster

- **Tasks to create a template and a new cluster:**
  - Export the cluster configuration from the source cluster
  - The exported configuration is a JSON file
  - Set up new hosts by installing CM agents and JDK
  - Create any local repositories required for the cluster.
  - Complete the instantiator section of the configuration JSON
  - Import the cluster template to the new cluster

Exporting the Cluster Configuration

- **Run the following command to download the JSON configuration file to a convenient location for editing**

```
curl -u adminuser:adminpass "http://
myCluster-1.myDomain.com:7180/api/v12/clusters/
Cluster1/export" > myCluster1-template.json
```

- **Modify the instantiator section of the JSON file you downloaded**

```
"instantiator" : {
  "clusterName" : "<changeme>",
  "hosts" : [ {
    "hostName" : "<changeme>",
    "hostTemplateRefName" : "<changeme>",
    "roleRefNames" : [ "HDFS-1-NAMENODE-0be88b55f5dedbf7bc74d61a86c0253e" ]
  }, {
    "hostName" : "<changeme>",
    "hostTemplateRefName" : "<changeme>"
  }, {
    "hostNameRange" : "<HOST[0001-0002]>",
    "hostTemplateRefName" : "<changeme>"
  } ],
  "variables" : [ {
    "name" : "HDFS-1-NAMENODE-BASE-dfs_name_dir_list",
    "value" : "/dfs/nn"
```

Importing the Template to a New Cluster

- **Complete the steps below to import the cluster template:**
  - Log in to the Cloudera Manager server as root
  - Run the following command to import the template

```
$curl -X POST -H "Content-Type: application/
json" -d@path_to_template/template_filename.json
 http://admin_user:admin_password@cloudera_manager_url:
 cloudera_manager_port/api/v12/cm/importClusterTemplate
```

  - You should see a response similar to the following:

```
${
  "id" : 17,
  "name" : "ClusterTemplateImport",
  "startTime" : "2016-03-09T23:44:38.491Z",
  "active" : true,
  "children" : {
    "items" : [ ]}
```

Chapter Topics

Cluster Maintenance

- Checking HDFS Status
- Copying Data Between Clusters
- Rebalancing Data in HDFS
- HDFS Directory Snapshots
- Hands-On Exercise: Creating and Using a Snapshot
- Host Maintenance
- Upgrading a Cluster
- **Essential Points**
- Hands-On Exercise: Upgrade the Cluster

Essential Points

- **You can check the status of HDFS with the `hdfs fsck` command**
  - Reports problems but does not repair them
- **You can use the `distcp` command to copy data within a cluster or between clusters**
- **Rebalance to adjust block placement across HDFS to ensure better utilization**
  - Especially after adding new DataNodes
- **A Snapshot is a read-only copy of an HDFS directory at a point in time**
- **Maintenance Mode allows you to suppress alerts for a host, service, role, or an entire cluster**

Chapter Topics

Cluster Maintenance

- Checking HDFS Status
- Copying Data Between Clusters
- Rebalancing Data in HDFS
- HDFS Directory Snapshots
- Hands-On Exercise: Creating and Using a Snapshot
- Host Maintenance
- Upgrading a Cluster
- Essential Points
- **Hands-On Exercise: Upgrade the Cluster**

Hands-On Exercise: Upgrade the Cluster

- **In this exercise, you will upgrade your cluster**
- **Please refer to the Hands-On Exercise Manual for instructions**

Cluster Monitoring

Chapter 14

Course Chapters

- Introduction
- Cloudera Data Platform
- CDP Private Cloud Base Installation
- Cluster Configuration
- Data Storage
- Data Ingest
- Data Flow
- Data Access and Discovery
- Data Compute
- Managing Resources
- Planning Your Cluster
- Advanced Cluster Configuration
- Cluster Maintenance
- **Cluster Monitoring**
- Cluster Troubleshooting
- Security
- Private Cloud / Public Cloud
- Conclusion
- Appendix: Cloudera Manager API
- Appendix: Ozone Overview

Cluster Monitoring

By the end of this chapter, you will be able to

- **Summarize monitoring features in Cloudera Manager**
- **Explore health notifications**
- **Use and customize Cloudera Manager dashboards**
- **Configure notification thresholds and alerts**
- **Review charts and reports to identify potential issues**

Chapter Topics

Cluster Monitoring

- **Cloudera Manager Monitoring Features**
- Health Tests
- Hands-On Exercise: Breaking the Cluster
- Events and Alerts
- Charts and Reports
- Monitoring Recommendations
- Essential Points
- Hands-On Exercise: Confirm Cluster Healing and Configuring Email Alerts

Monitoring with Cloudera Manager

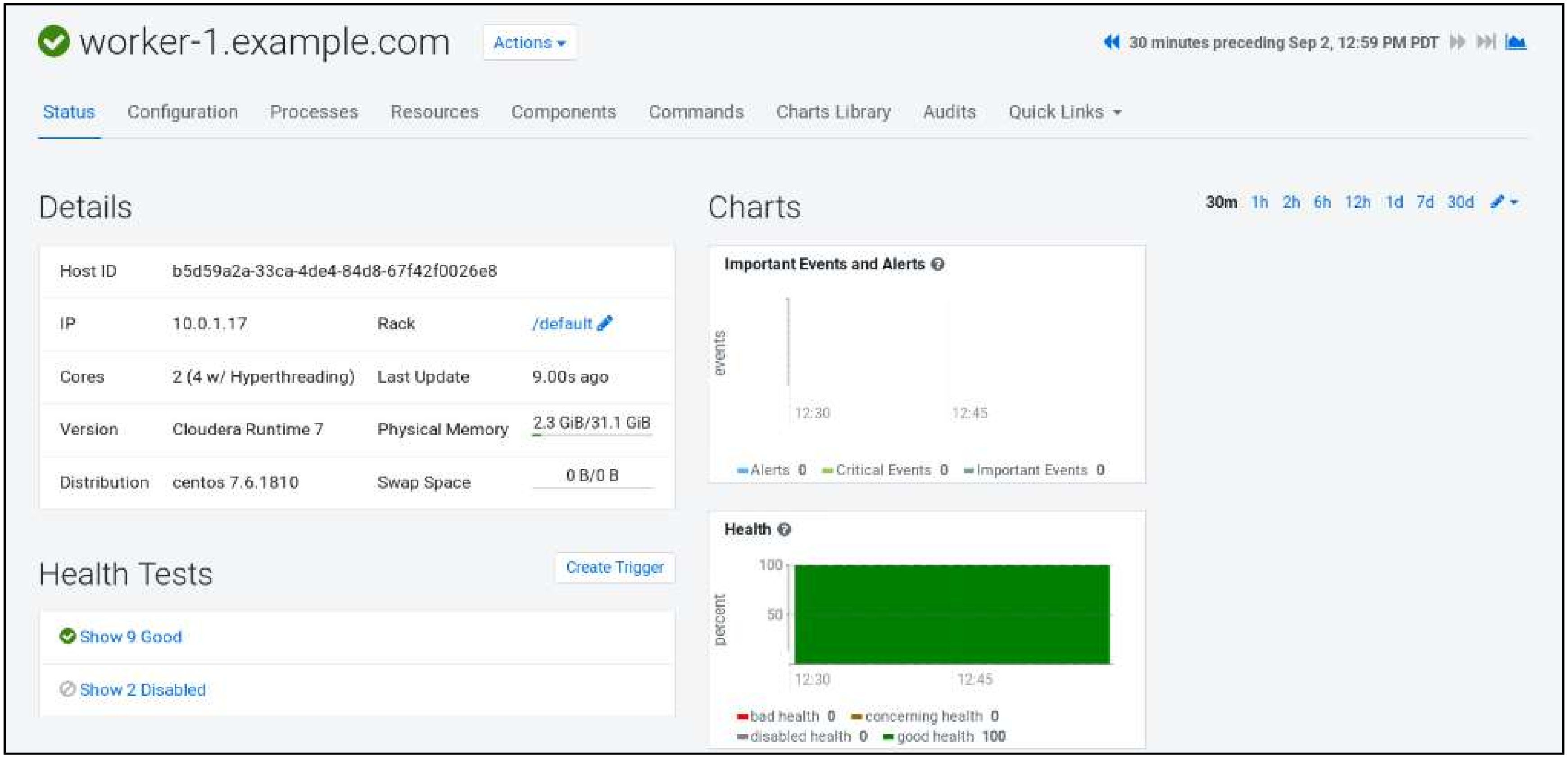
- **Use Cloudera Manager to monitor health and performance of your cluster**
  - Monitor cluster health
  - Identify configuration issues
  - Track metrics and resource usage with charts and dashboards
  - View event logs
  - Generate alerts
  - Audit Cloudera Manager events
  - Generate reports

Monitoring Terminology

- **Entity—a Cloudera Manager component with metrics associated with it**
  - Examples: clusters, services, roles and role instances, hosts
- **Metric—a property that can be measured**
  - Cloudera Manager monitors performance metrics cluster entities such as hosts and services
  - Examples: RAM utilization, total HDFS storage capacity
- **Chart—customizable display aggregated metrics for entities over time**
- **Dashboard—a page displaying key entity information and charts**

Entity Status Tab

- **Pages for clusters, services, hosts, roles, and other entities have** Status **tabs**
  - Customizable dashboards with key details about the current state



Chapter Topics

Cluster Monitoring

- Cloudera Manager Monitoring Features
- **Health Tests**
- Hands-On Exercise: Breaking the Cluster
- Events and Alerts
- Charts and Reports
- Monitoring Recommendations
- Essential Points
- Hands-On Exercise: Confirm Cluster Healing and Configuring Email Alerts

Health Tests

- **Cloudera Manager monitors the health of services, roles, and hosts**
- **Pass-fail tests—entity is either "good" or "bad"**
  - *Canary* test: Does service appear to be working correctly?
  - Yes-no test: Check for a specific property
  - Example: Are all DataNodes connected to a NameNode?
- **Metric tests—compare numeric value to a configurable *threshold***
  - Result is "good", "bad", or "concerning"
  - Example: does sufficient HDFS capacity remain?
    - "Concerning" threshold: < 60% remaining
    - "Bad" threshold: < 80% remaining

Monitoring Health Issues in Cloudera Manager (1)

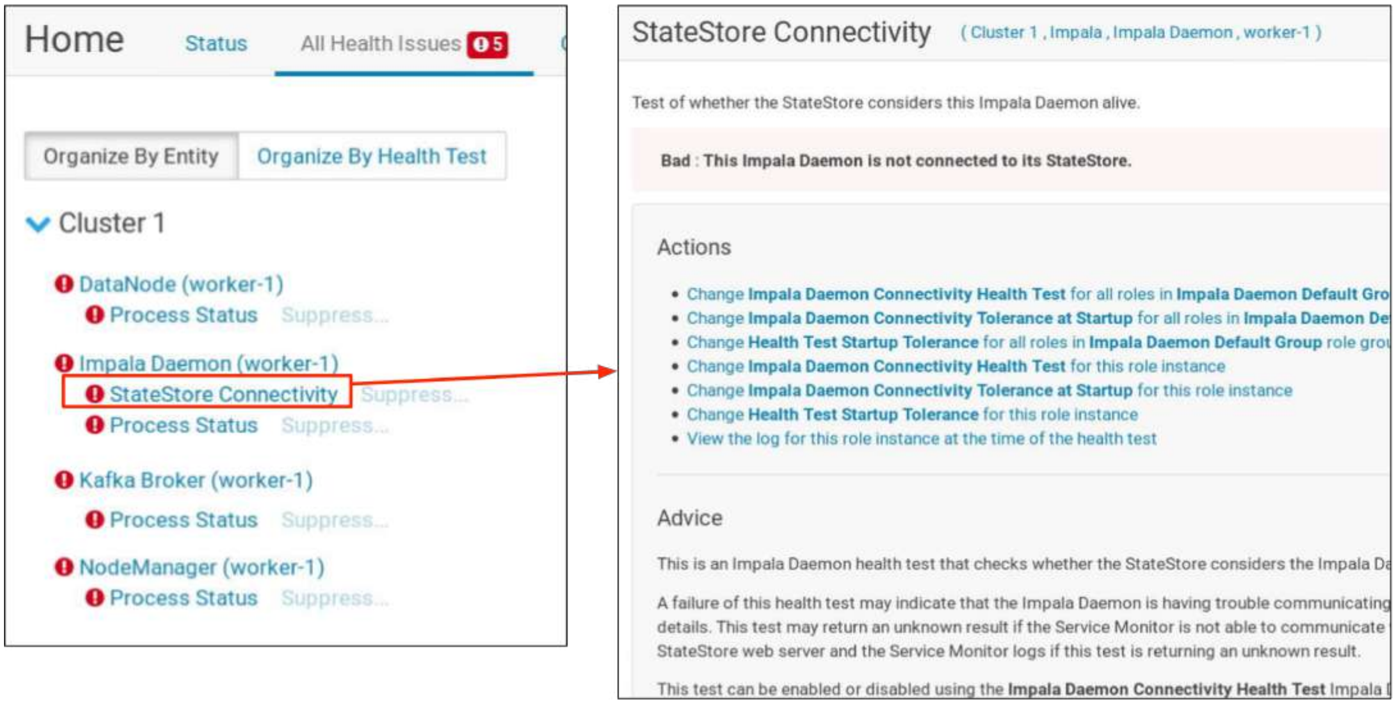
- **Cluster Health Status Indicators**
  - Good—test result above all thresholds
  - Concerning—test result below the "warning" threshold
  - Bad—test result below the "critical" threshold

Cluster Status

Monitoring Health Issues in Cloudera Manager (2)

- **You can suppress health tests or view test status, corrective actions, and advice**

Health Issues Detail



Chapter Topics

Cluster Monitoring

- Cloudera Manager Monitoring Features
- Health Tests
- **Hands-On Exercise: Breaking the Cluster**
- Events and Alerts
- Charts and Reports
- Monitoring Recommendations
- Essential Points
- Hands-On Exercise: Confirm Cluster Healing and Configuring Email Alerts

Breaking the Cluster

- **In this exercise, you will explore and configure a number of Cloudera Manager monitoring features**
  - Please refer to the Hands-On Exercise Manual for instructions

Chapter Topics

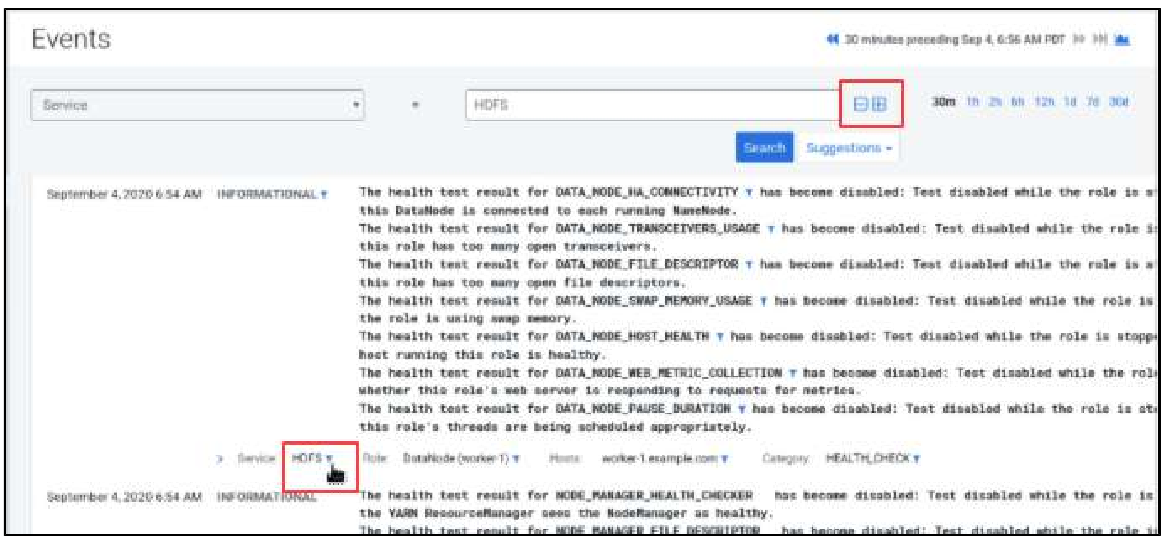
Cluster Monitoring

- Cloudera Manager Monitoring Features
- Health Tests
- Hands-On Exercise: Breaking the Cluster
- **Events and Alerts**
- Charts and Reports
- Monitoring Recommendations
- Essential Points
- Hands-On Exercise: Confirm Cluster Healing and Configuring Email Alerts

Events

- **An *event* is a record of something of interest that occurred**
  - Default settings enable the capture of many events
- **Event types include (click on + or the filter icon to add an event)**
  - `ACTIVITY_EVENT`—jobs that fail or run slowly
  - `AUDIT_EVENT`—actions taken in Cloudera Manager such as starting a role
  - `HEALTH_CHECK`—health test results
  - `LOG_MESSAGE`—log messages from HDFS, HBase, or MapReduce

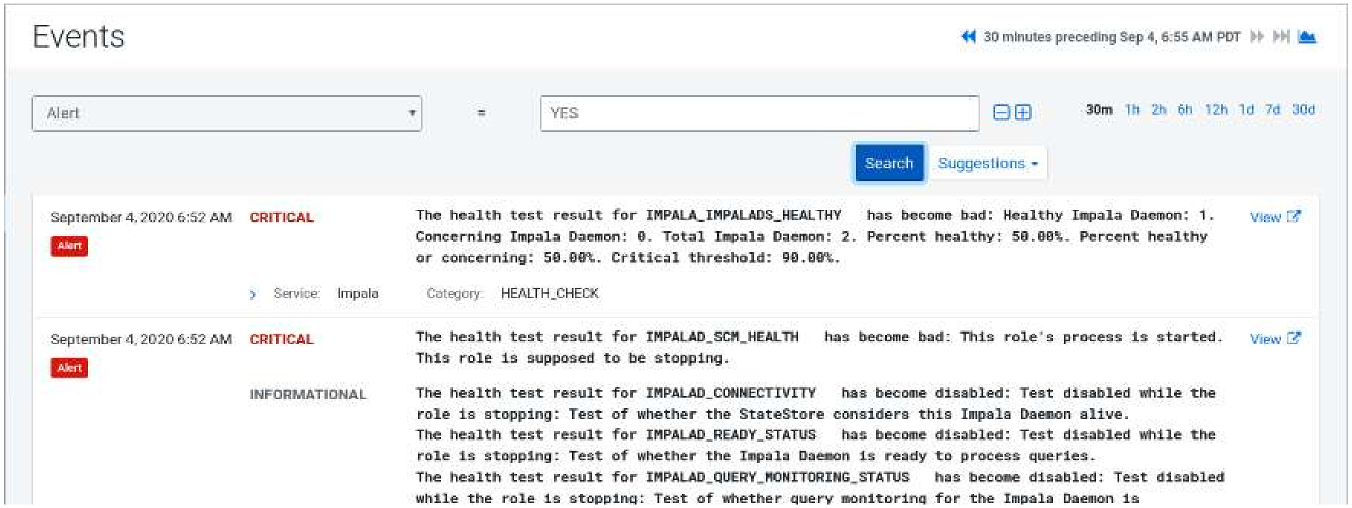
Diagnostics > Events



Alerts

- **Alerts are events triggered by “noteworthy” events**
- **Alerts can be configured for**
  - Activity running too slowly
  - A configuration was changed
  - Health condition thresholds not met on a role or host
  - Log messages or events that match a condition you define
- **Alerts are noted in the Cloudera Manager event viewer**

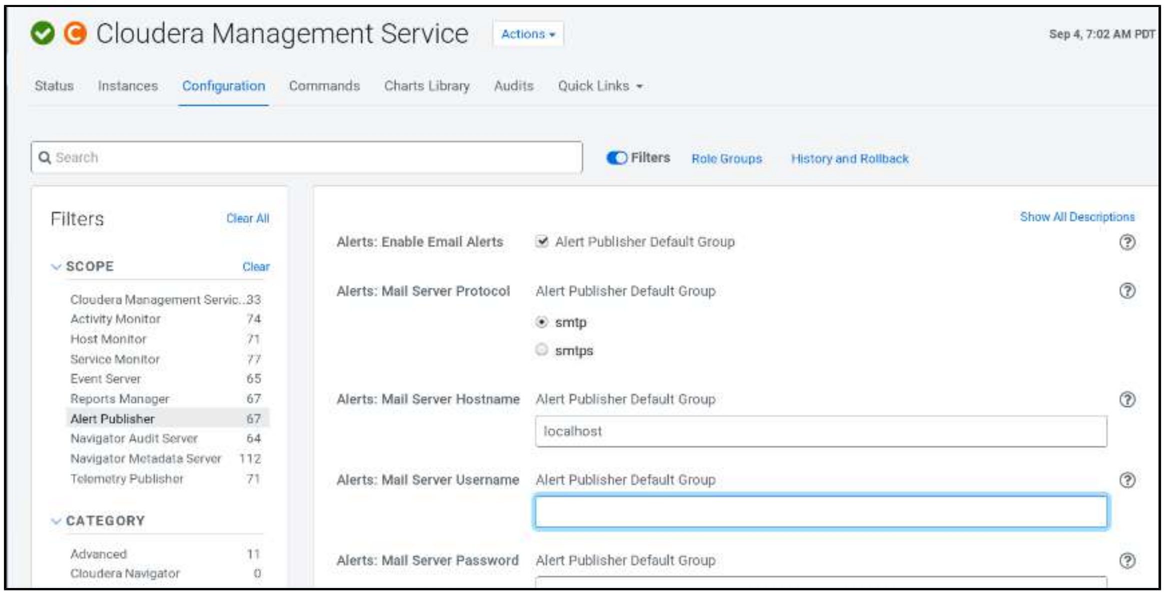
Diagnostics > Events



Alert Delivery

- **Alert delivery options**
  - Send email to an address you configure
  - Send SNMP* traps for external monitoring systems
- **Enable and configure event delivery in Cloudera Manager**

Cloudera Manager Service > Configuration tab OR Administration > Alerts

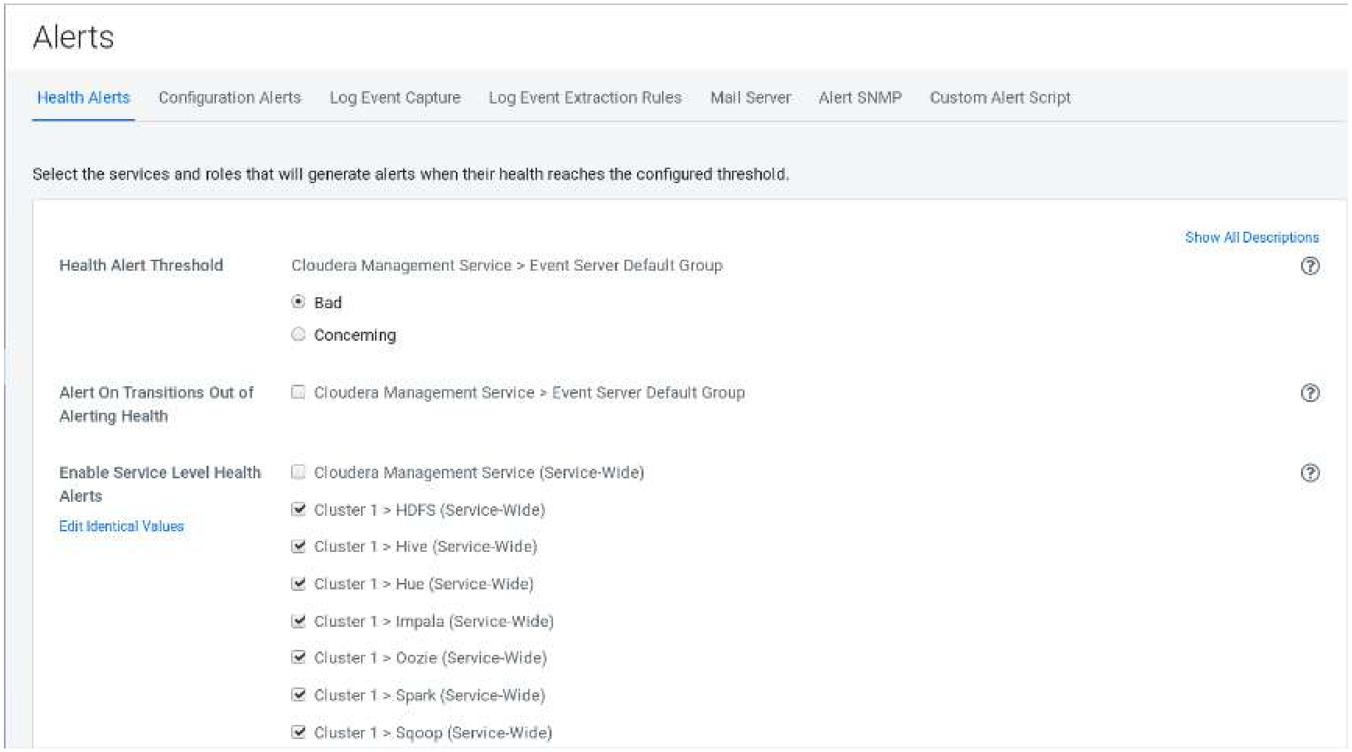


*Simple Network Management Protocol

Viewing Enabled and Disabled Alerts

- **View enabled and disabled alerts, organized by type and service**

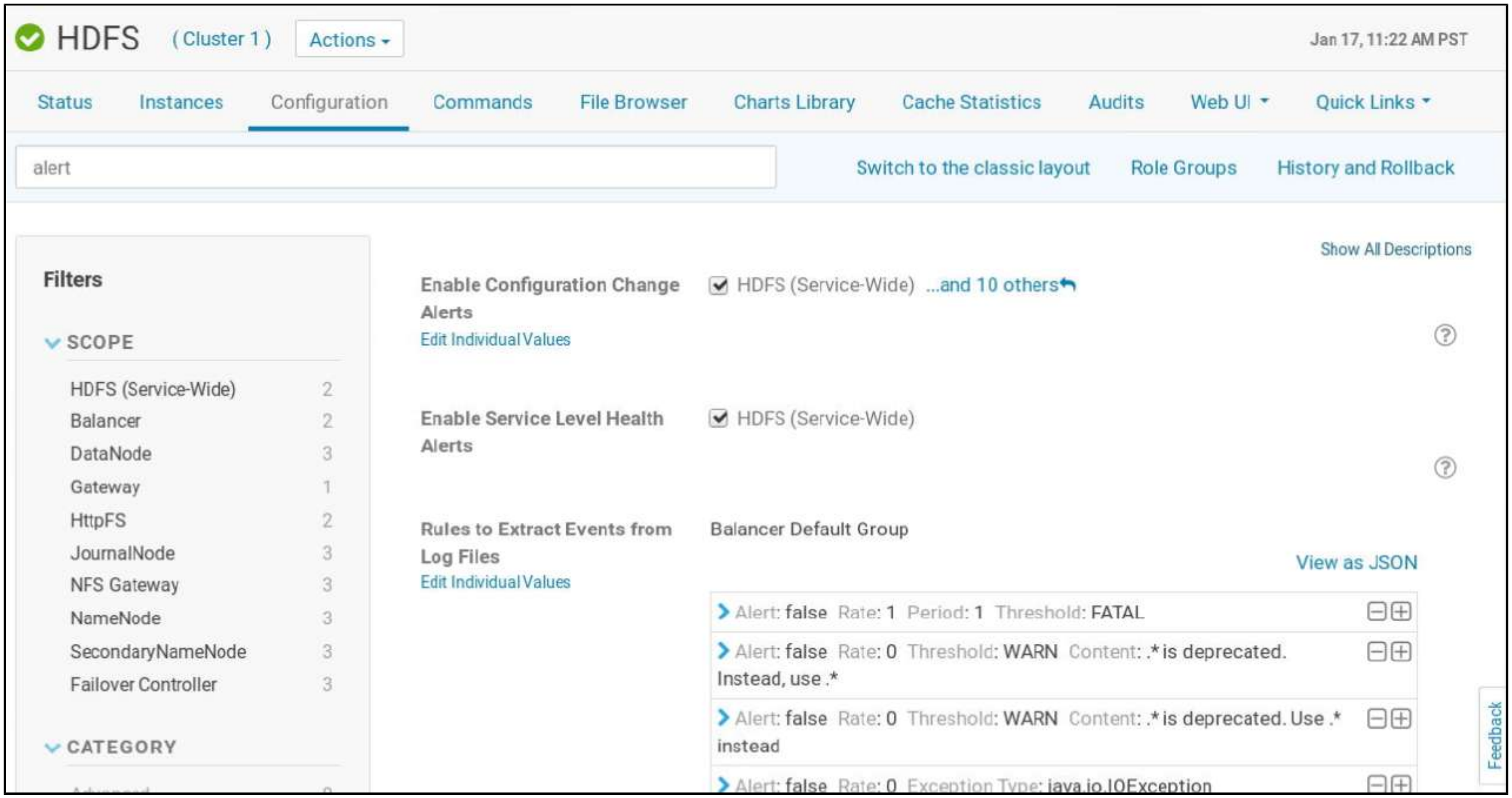
Administration > Alerts



Configuring Alerts

- **Enable and configure alerts for services on the** Configuration **tab**
- **Many alerts are enabled by default**

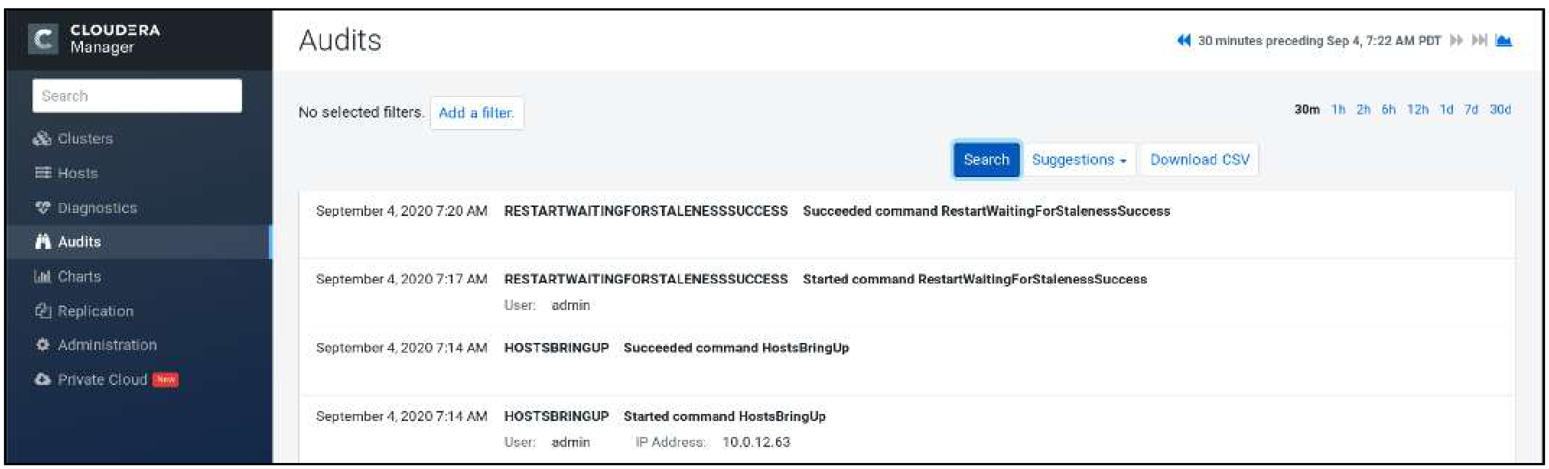
HDFS Configuration tab



Audit Events

- **Audit events describe actions that occurred on a host or for a service or role**
  - **Lifecycle events**—such as starting or stopping a role or host, installing or upgrading services, and activating parcels
  - **Security events**—such as login success or failure, and adding or deleting users
- **Audit events include important details such as time, user, command, and IP address**

Cloudera Manager Audits Page



Items to View Using Cloudera Manager (1)

- **Monitoring Cloudera Runtime Services**
  - View the results of health tests at both the service and role instance level
- **Monitoring Hosts**
  - Look at a summary view for all hosts in your cluster or drill down for extensive details about an individual host
- **Activities**
  - View the activities running on the cluster, both at the current time and through dashboards that show historical activity
- **Events**
  - Filter events by time range, service, host, keyword

Items to View Using Cloudera Manager (2)

- **Alerts**
  - Cloudera Manager to generate alerts from certain events
- **Lifecycle and Security Auditing**
  - Events such as creating a role or service, making configuration revisions for a role or service, decommissioning and recommissioning hosts, and running commands
- **Logs**
  - Easily view the relevant log entries that occurred on the hosts used by the job while the job was running
- **Reports**
  - View historical information aggregated over selected time periods

Chapter Topics

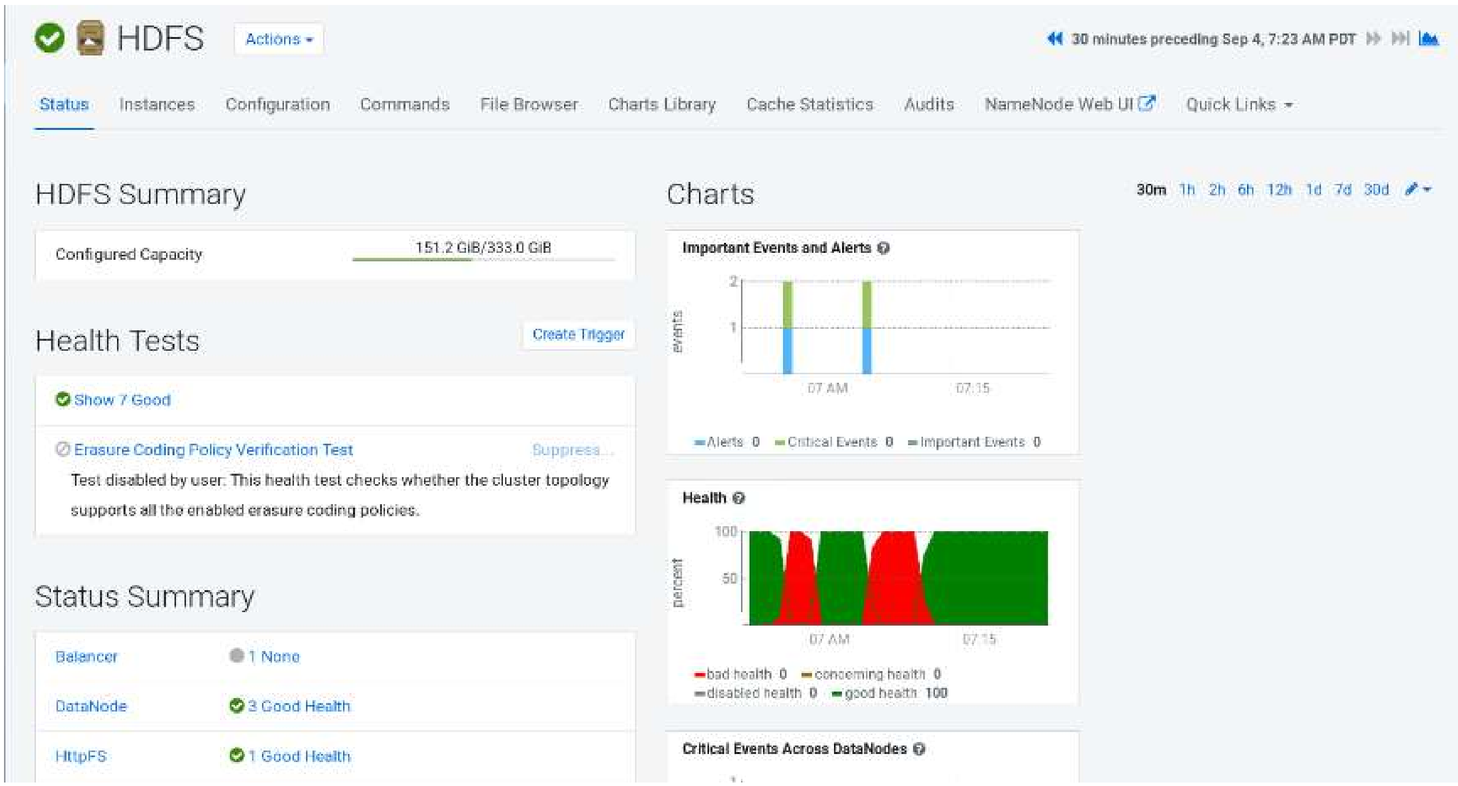
Cluster Monitoring

- Cloudera Manager Monitoring Features
- Health Tests
- Hands-On Exercise: Breaking the Cluster
- Events and Alerts
- **Charts and Reports**
- Monitoring Recommendations
- Essential Points
- Hands-On Exercise: Confirm Cluster Healing and Configuring Email Alerts

Pre-Built Dashboards

- **Entity status tabs display default dashboards**
  - Customizable with pre-built or custom charts

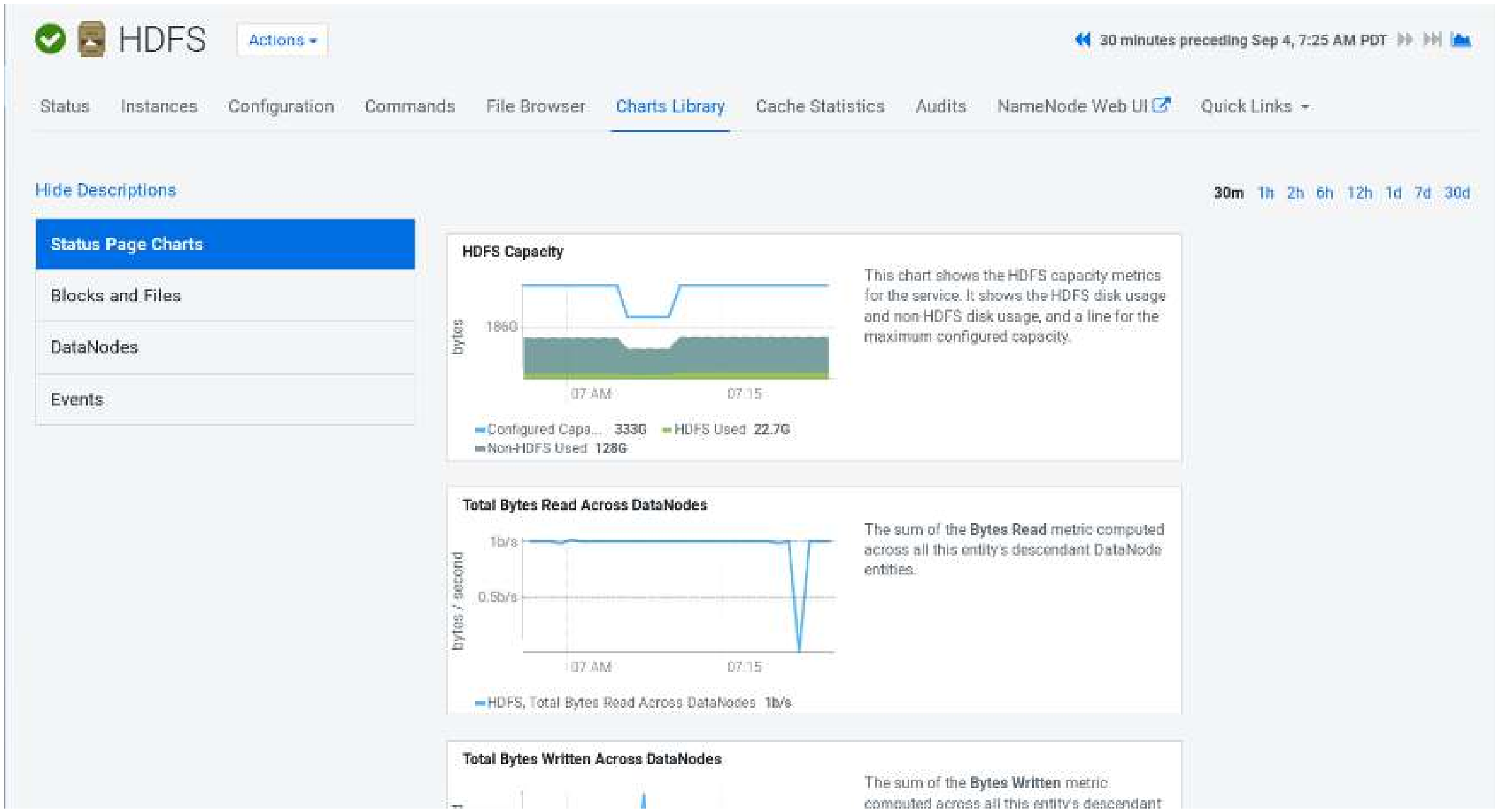
HDFS Status tab



Charts Library

- **Cloudera Manager charts library provides many pre-built charts**

HDFS Charts Library tab



Custom Charts

- **Create custom charts to add to dashboards using the *tsquery* language**
- **Example: Show read and write rates for DataNodes**
  - `SELECT bytes_read_rate, bytes_written_rate WHERE roleType=DATANODE AND serviceName=HDFS`

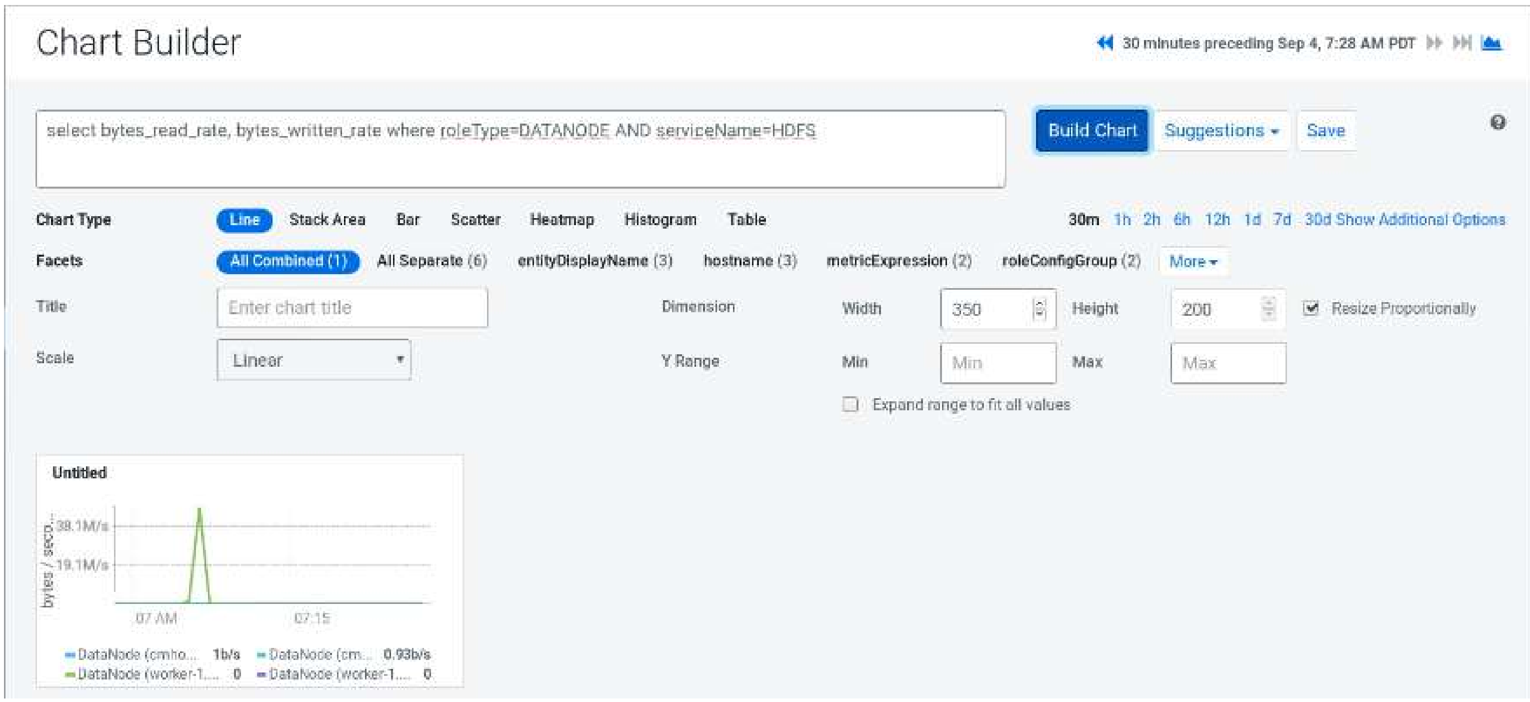
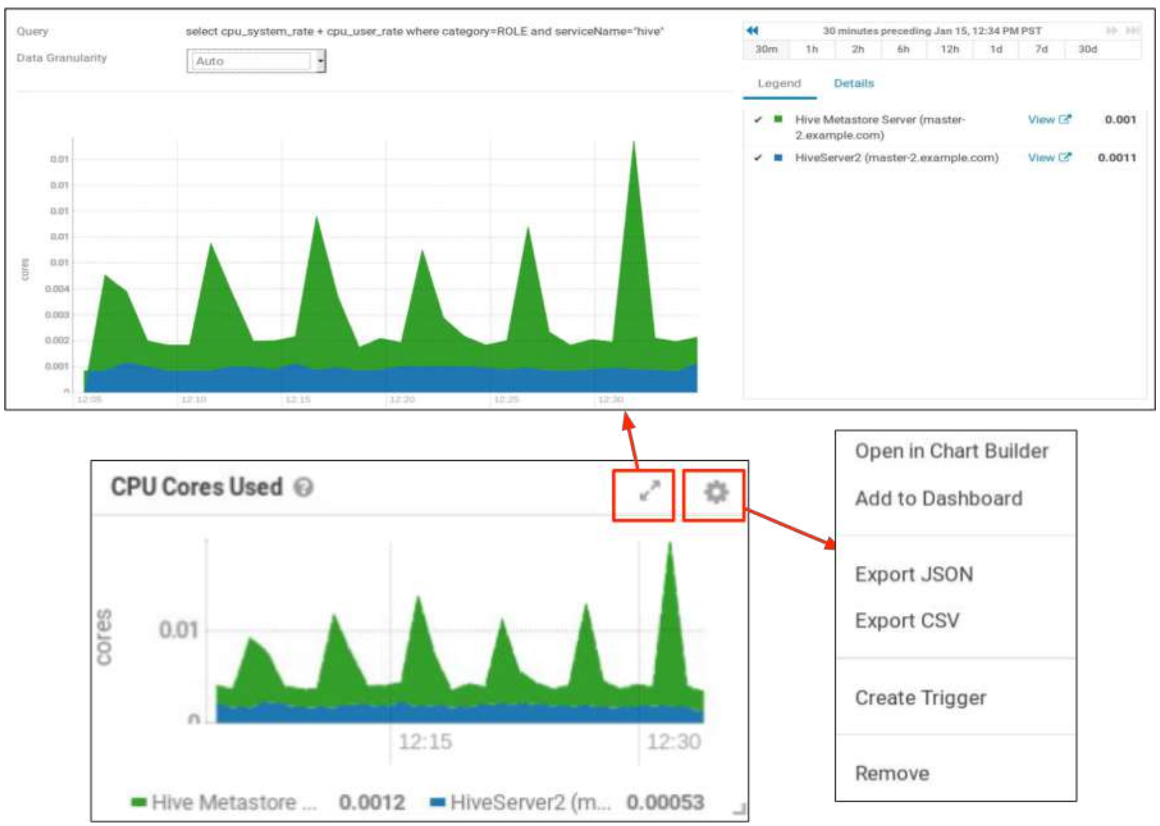


Chart Options

- **Hover over a chart to see options**
  - Click on the ↙↗ icon to expand the chart viewing area and details
  - Click on the ⚙ icon for more options

Hive Status Tab

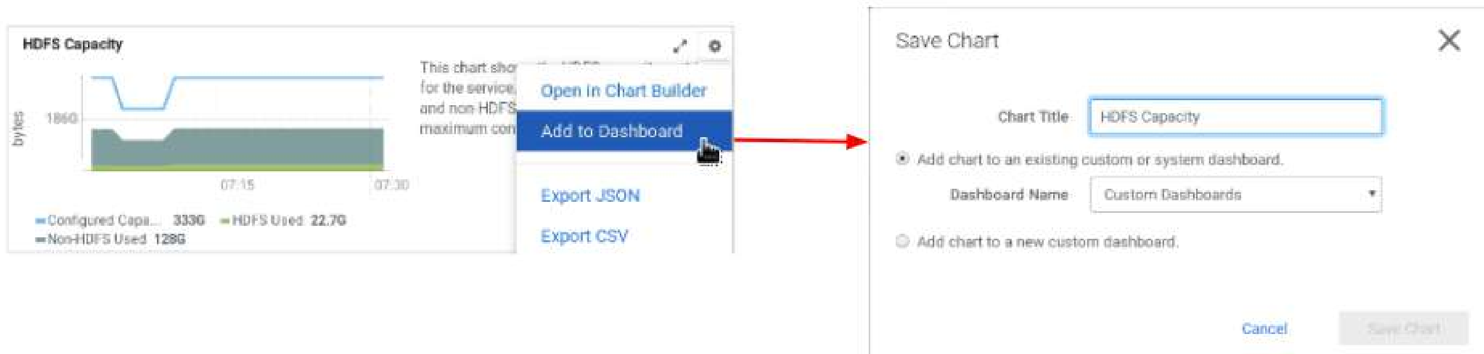


Dashboards (1)

- **A dashboard consists of a set of charts**
- **Cloudera provides many pre-configured dashboard**
  - For example, each service's status page has a dashboard
  - Click on the icon to manage an existing dashboard
    - Options to add or remove charts, change layout

Dashboards (2)

- **Custom dashboards add existing or custom charts**

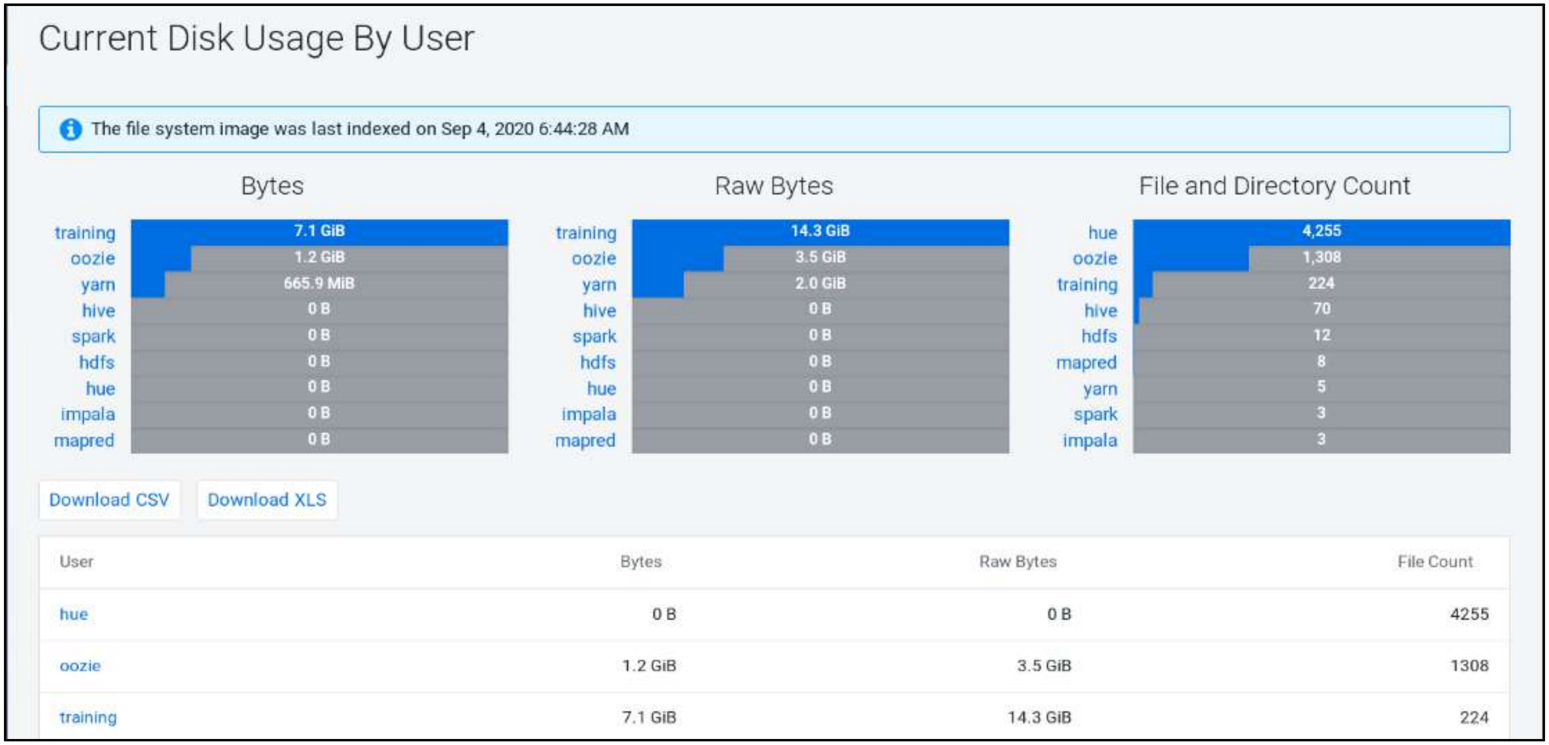


- **Manage your user-defined dashboards in** Charts

Reports

- **Per-cluster reports are available for disk usage, YARN applications, Impala queries, HDFS file access, and HBase tables and namespaces**
- **Download reports or view them in Cloudera Manager**

Clusters > Reports > Select the Report

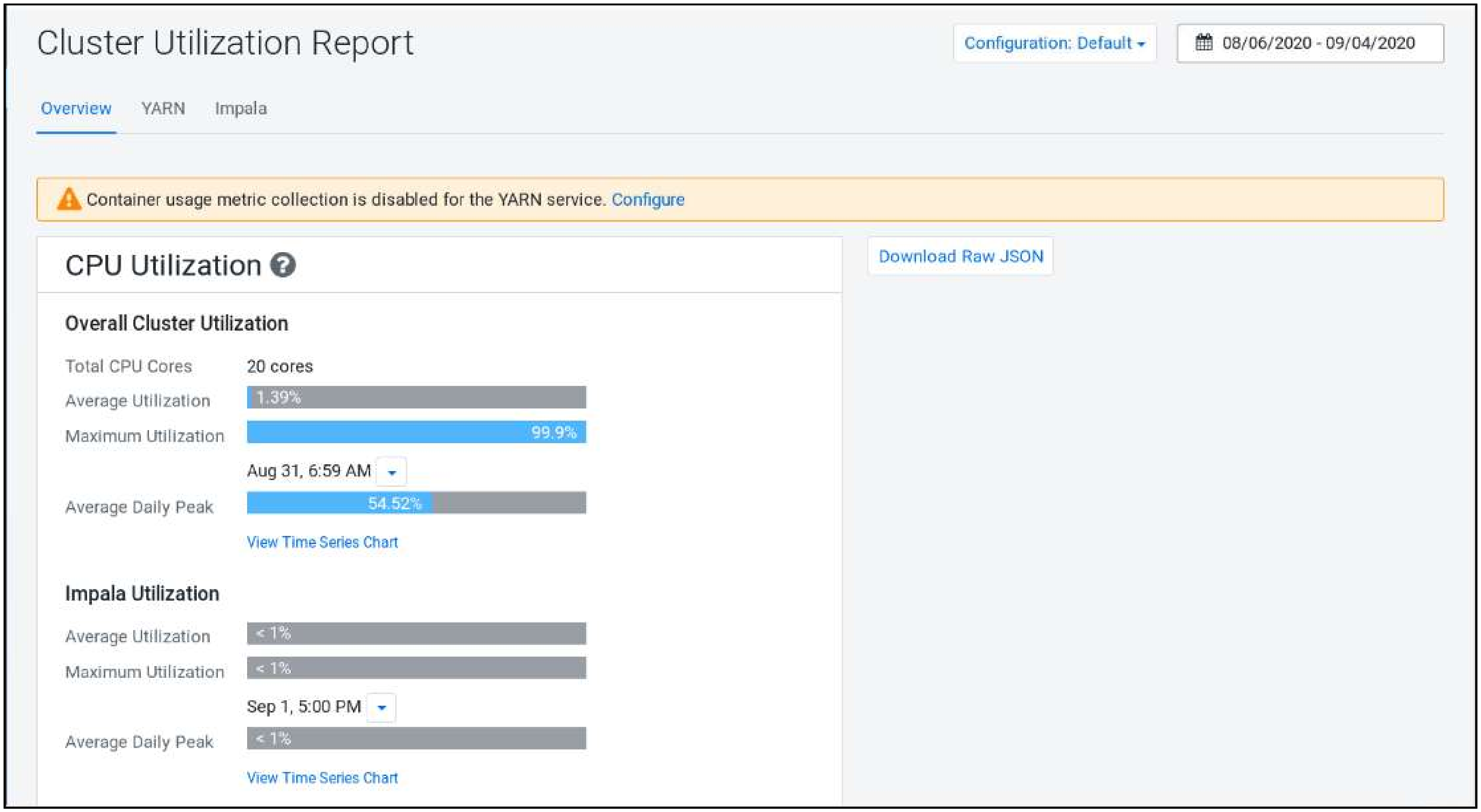


Cluster Utilization Report (1)

- **The *Cluster Utilization Report* displays information about resource utilization**
  - CPU utilization
  - Memory utilization
  - Resource allocation
  - Optional: YARN application metrics
    - MapReduce job periodically aggregates metrics by application
- **Metrics aggregated for a whole cluster or by *tenant* (user or resource pool)**

Cluster Utilization Report (2)

Clusters > Utilization Report



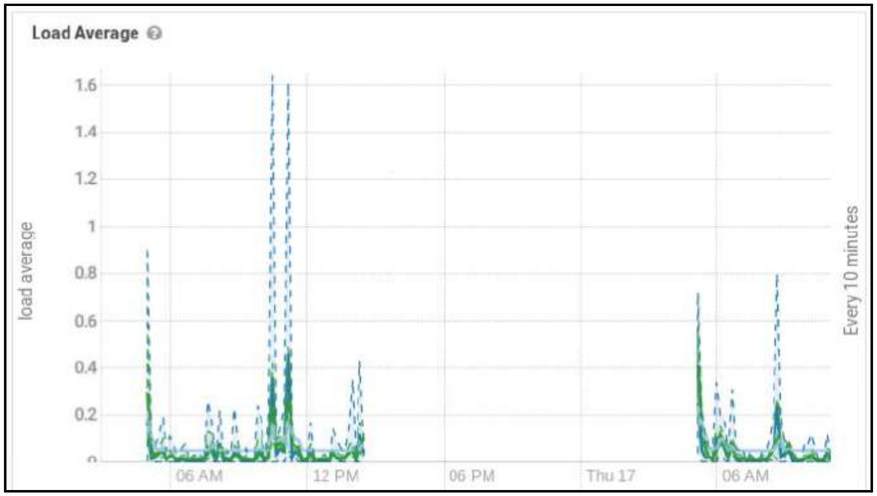
Chapter Topics

Cluster Monitoring

- Cloudera Manager Monitoring Features
- Health Tests
- Hands-On Exercise: Breaking the Cluster
- Events and Alerts
- Charts and Reports
- **Monitoring Recommendations**
- Essential Points
- Hands-On Exercise: Confirm Cluster Healing and Configuring Email Alerts

Monitoring Recommendations: Daemons and CPU Usage

- **Service daemons**
  - Recommendation: send an alert if a daemon goes down
    - Cloudera Manager default will alert for some but not all daemons down
- **CPU usage on master hosts**
  - Watch for excessive CPU usage and load averages for performance and utilization management
  - **Host CPU Usage**, **Roles CPU Usage**, and **Load Average** charts
  - Worker hosts will often reach 100% usage—this is okay



Monitoring Recommendations: HDFS

- **Monitor HDFS health**
  - HA configuration
    - Check the size of the edit logs on the JournalNodes
    - Monitor for failovers
  - Non-HA configuration
    - Check the age of the `fsimage` file and/or size of the `edits` file
- **DataNode alert settings**
  - Critical (“bad”) alert if a DataNode disk fails (enabled by default)
    - “DataNode Volume Failures Thresholds” property
  - Enable DataNode health test alerts (disabled by default)
  - Disk capacity alert: Warning (“concerning”) at 80% full, critical (“bad”) at 90% full
    - “DataNode Free Space Monitoring Thresholds” property

Monitoring Recommendations: Memory, Network, and Disk

- **Monitor network transfer speeds**
- **Monitor swapping on all hosts**
  - Warning alert if any swapping occurs—memory allocation is overcommitted
    - “Host Memory Swapping Thresholds” property
- **Monitor for disk failure and space available, especially for master nodes**

Log File Disk Usage (1)

- **HDFS and YARN daemon logs**
  - HDFS and YARN daemons use Log4j's Rolling File Appender, so log files are rotated
  - Configure appropriate size and retention policies in Cloudera Manager
- **YARN application logs**
  - Caution: applications by inexperienced developers will often create large container logs
  - Large task logs impact disk usage as they are written to local disks on worker hosts first
  - Ensure you have enough room locally for application logs

Log File Disk Usage (2)

- **Monitor local disks and HDFS to ensure that applications are not logging excessively**
- **Configure properties to control log growth**
  - `yarn.log-aggregation.retain-seconds`
    - Retention of log files on HDFS when log aggregation *is* enabled
    - Cloudera Manager default: seven days

Chapter Topics

Cluster Monitoring

- Cloudera Manager Monitoring Features
- Health Tests
- Hands-On Exercise: Breaking the Cluster
- Events and Alerts
- Charts and Reports
- Monitoring Recommendations
- **Essential Points**
- Hands-On Exercise: Confirm Cluster Healing and Configuring Email Alerts

Essential Points

- **Monitoring your cluster will help prevent, detect, and diagnose issues**
  - Such as daemons health, disk usage, CPU usage, swap, network usage
- **Cloudera Manager provides many cluster monitoring features**
  - Health tests—services and hosts noted as “good”, “bad”, or “concerning”
  - Alerts—can send notifications if certain events occur such as when a health condition crosses a threshold or a configuration change is made
  - Charts and reports—provide continuous information about how the cluster is functioning
  - Monitoring features are highly customizable

Chapter Topics

Cluster Monitoring

- Cloudera Manager Monitoring Features
- Health Tests
- Hands-On Exercise: Breaking the Cluster
- Events and Alerts
- Charts and Reports
- Monitoring Recommendations
- Essential Points
- **Hands-On Exercise: Confirm Cluster Healing and Configuring Email Alerts**

Confirm Cluster Healing and Configuring Email Alerts

- **In this exercise, you will confirm the healing of the cluster and setup email alerts**
  - Please refer to the Hands-On Exercise Manual for instructions

Cluster Troubleshooting

Chapter 15

Course Chapters

- Introduction
- Cloudera Data Platform
- CDP Private Cloud Base Installation
- Cluster Configuration
- Data Storage
- Data Ingest
- Data Flow
- Data Access and Discovery
- Data Compute
- Managing Resources
- Planning Your Cluster
- Advanced Cluster Configuration
- Cluster Maintenance
- Cluster Monitoring
- **Cluster Troubleshooting**
- Security
- Private Cloud / Public Cloud
- Conclusion
- Appendix: Cloudera Manager API
- Appendix: Ozone Overview

Cluster Troubleshooting

After completing this chapter, you will be able to

- **Detect and diagnose cluster problems using CM monitoring**
- **Troubleshoot issues with Cloudera Manager and CDP services**
- **Find and repair data corruption in HDFS**
- **Identify root causes of job failure using log files and stack dumps**
- **Describe common cluster misconfigurations and how to fix them**

Chapter Topics

Cluster Troubleshooting

- **Overview**
- Troubleshooting Tools
- Misconfiguration Examples
- Essential Points
- Hands-On Exercise: Troubleshooting a Cluster

Troubleshooting: The Challenges

- **An overt symptom is a poor indicator of the root cause of a failure**
- **Errors show up far from the cause of the problem**
- **Clusters have a lot of components**
- **Example**
  - Symptom: A YARN job that previously ran successfully is now failing
  - Cause: Disk space on many hosts has filled up, so intermediate data cannot be copied to reducers

Common Sources of Problems

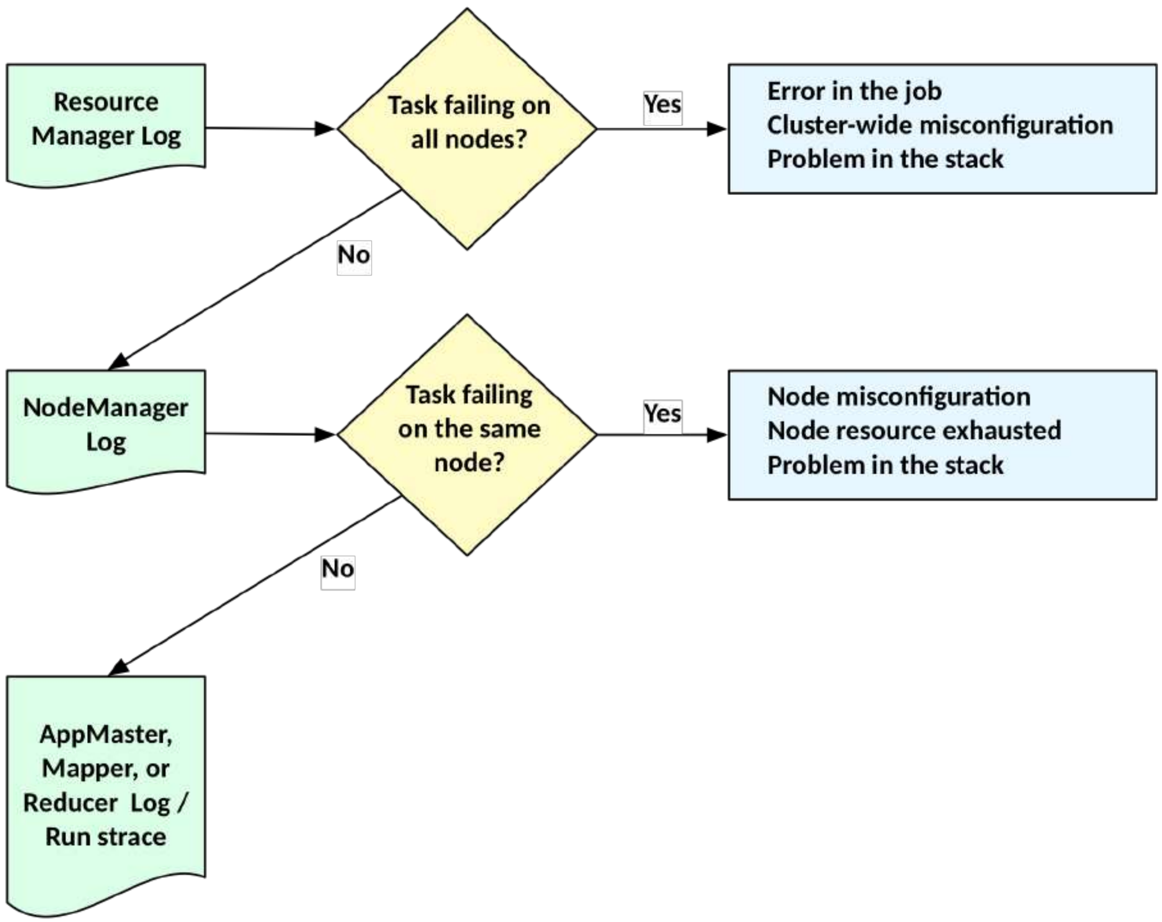
- **Misconfiguration**
- **Hardware failure**
- **Resource exhaustion**
  - Not enough disks, RAM, or network bandwidth
- **Inability to reach hosts on the network**
  - Naming issues
  - Network hardware issues
  - Network delays

Gathering Information About Problems

- **Are there any issues in the environment?**
- **What about dependent components?**
  - Example: YARN applications depend on the ResourceManager, which depends on the underlying OS
- **Do the failures have aspects in common?**
  - All from the same application?
  - All from the same NodeManager?
- **Is this a resource problem?**
  - Have you received an alert from Cloudera Manager?
- **What do the logs say?**
- **What does the CDP documentation, Cloudera Knowledge Base, or an internet search say about the problem?**

General Rule: Start Broad, Then Narrow the Scope

- **Example: MapReduce application failure**



Avoiding Problems

- **Misconfiguration**
  - Start with Cloudera recommended property values
    - Do not rely on Apache defaults!
  - Understand the precedence of overrides
  - Limit users' ability to make configuration changes
  - Test changes before putting them into production
  - Look for changes when deploying new releases of CDP
  - Automate management of the configuration
- **Hardware failure and exhaustion**
  - Monitor your systems
  - Benchmark systems to understand their impact on your cluster
- **Hostname resolution**
  - Test forward and reverse DNS lookups

Chapter Topics

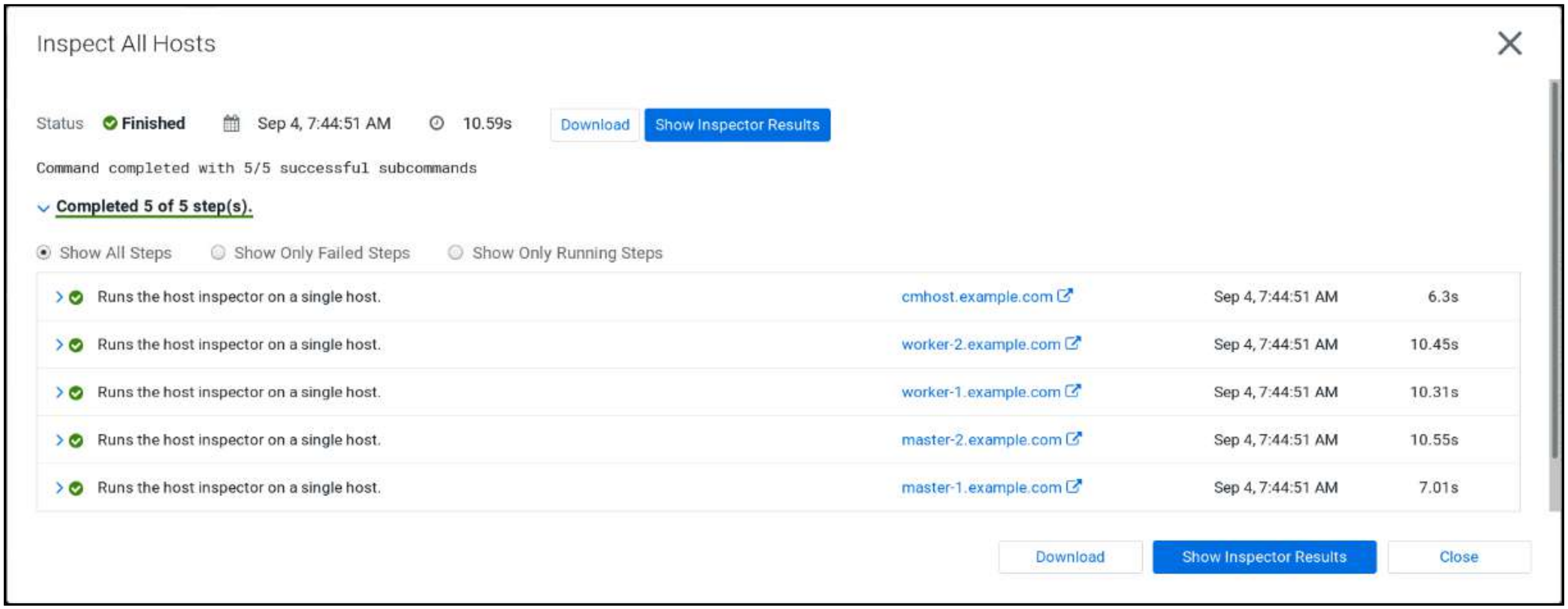
Cluster Troubleshooting

- Overview
- **Troubleshooting Tools**
- Misconfiguration Examples
- Essential Points
- Hands-On Exercise: Troubleshooting a Cluster

Cloudera Manager Host Inspector

- **The Host Inspector gathers information about host**
  - Such as HDFS settings, CDP Runtime component versions, system time, and networking configuration

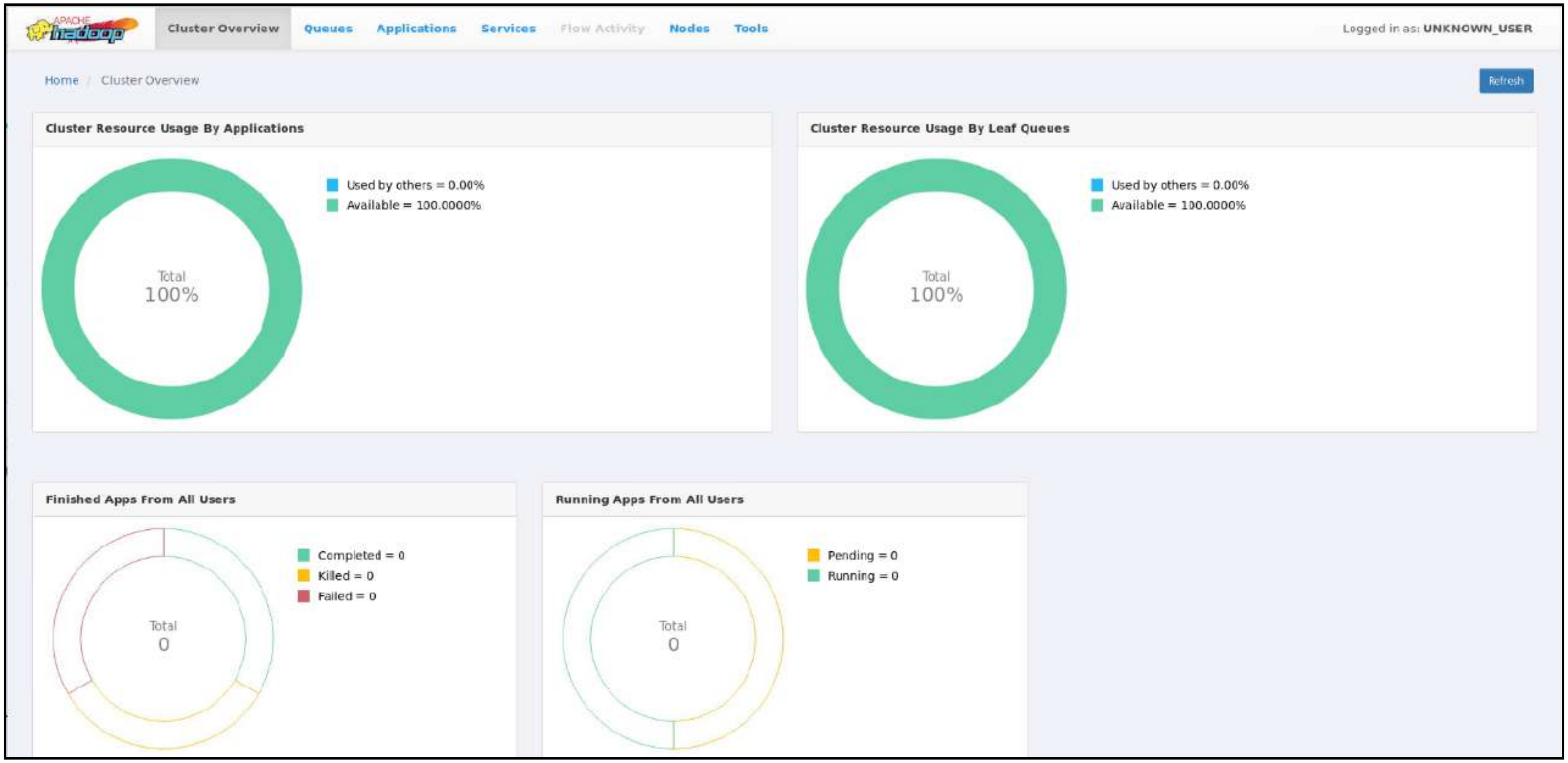
Hosts > All Hosts > Host Inspector



Service Web UIs

- **Many cluster services provide their own web UIs**
  - Often helpful for diagnosing problems
- **Example: Use the YARN ResourceManager and NodeManager UIs to detect YARN problems**

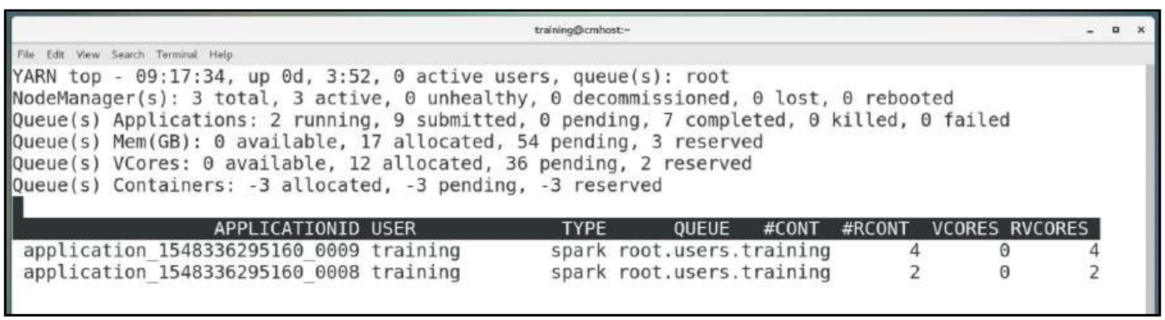
ResourceManager UI > All Applications



HDFS and YARN Commands

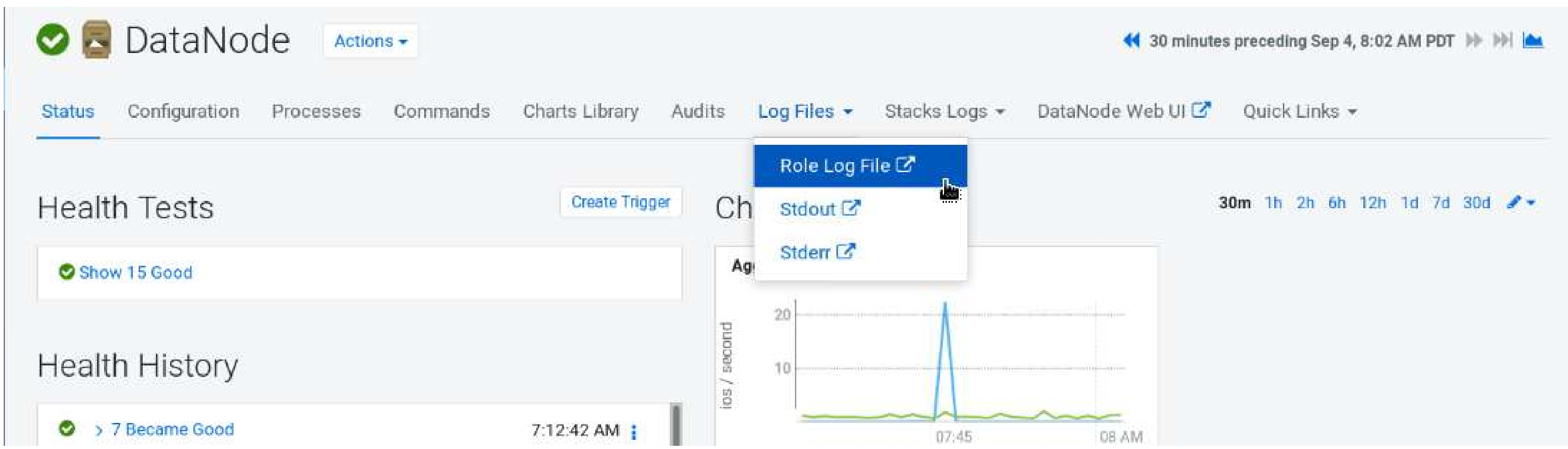
- **HDFS and YARN command line interfaces provide useful troubleshooting information**
- **`hdfs` subcommands include**
  - `getconf`: shows configuration values for a node
  - `dfsadmin -report`: checks for data corruption, shows status of file system and all DataNodes
  - `dfsadmin -safemode`: checks status, enters or leaves safemode
- **`yarn` subcommands include**
  - `node`, `container`, `application`, and `queue`: status reports
  - `top`: continuously displays queue and application information

`yarn top` command



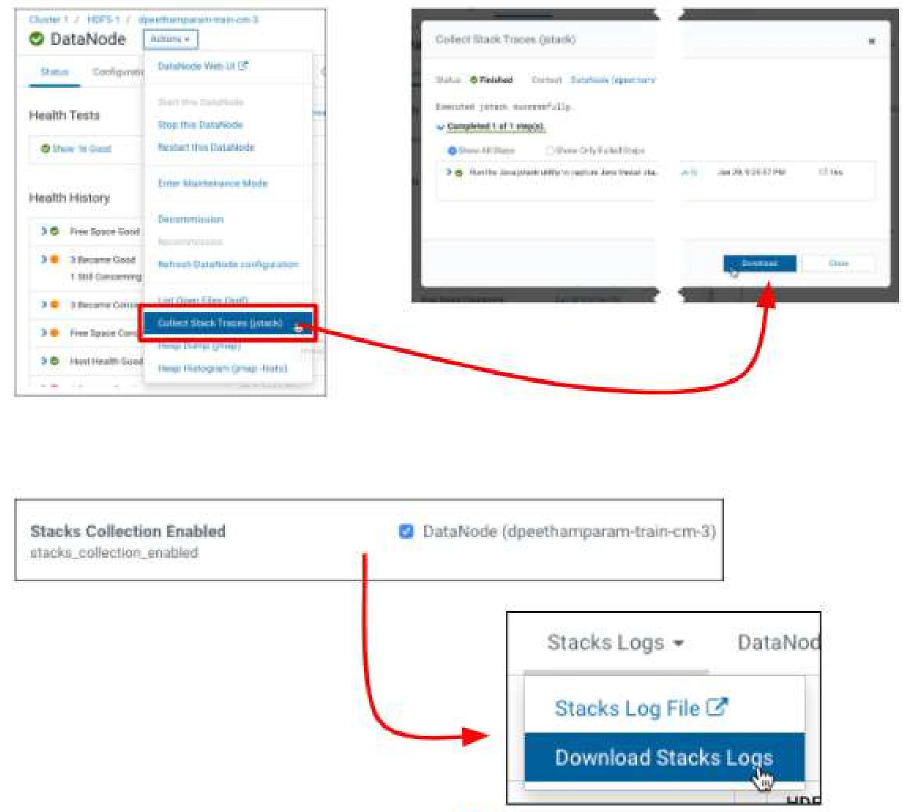
Accessing Role Logs

- **Role instance page**
  - Specific to that one role instance
  - Download the entire log
- **Log search page**
  - Filter criteria - service, role, host, and log level
  - Good for correlating across hosts or roles



Diagnostics - Thread Dump

- **Thread dump / stack trace**
  - Function call stack of each thread
  - High CPU / hung / slow response
  - Series of thread dumps
- **Single thread dump**
  - Actions menu of entity
- **Continual dump**
  - Configuration menu - Stacks Collection
  - Set the frequency or retention size
- **Retrieve from Stacks Logs menu**

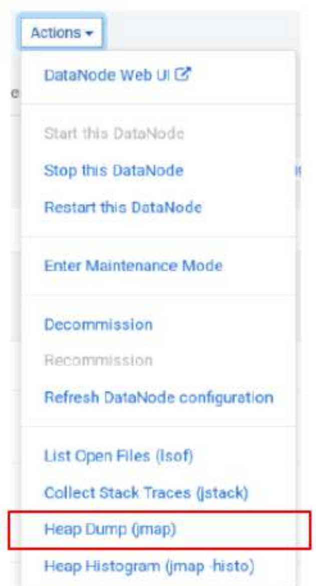


Diagnostics - Heap Dump / Heap Histogram

- **Heap dump - information about objects on JVM heap**
- **Heap histogram - summary of objects**
- **Single heap dump from Actions menu**
- **Cautions: Involve Cloudera support person**
  - Causes stop the world garbage collection
  - Performance hit with large files
  - Scratch dir is `/tmp` (tempfs)

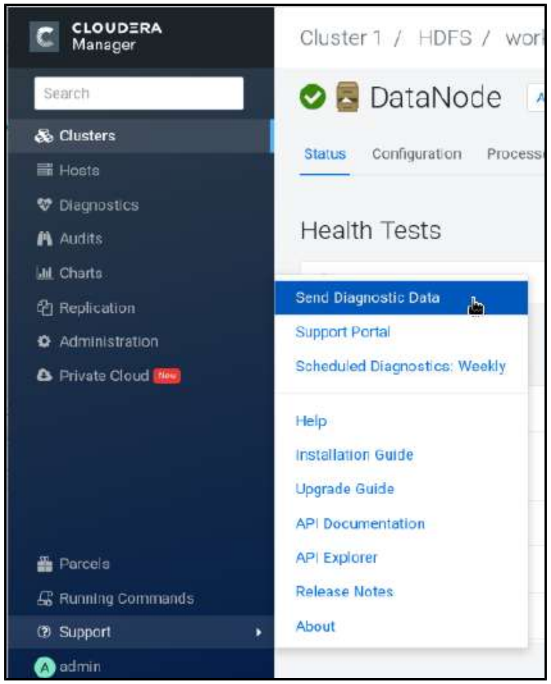
Diagnostic Commands for Roles

- **Cloudera Manager allows administrators to run diagnostic utility tools against most Java-based role processes:**
  - List Open Files (lsof) - Lists the open files of the process
  - Collect Stack Traces (jstack) - Captures Java thread stack traces
  - Heap Dump (jmap) - Captures a heap dump for the process
  - Heap Histogram (jmap -histo) - Produces a histogram of the heap
- **These commands are found on the Actions menu of the Cloudera Manager page for the instance of the role**



Diagnostic Bundle

- Also known As "support bundle" or "diagnostic bundle"
- Gathers events, logs, config, host information, host inspector output, cluster metadata and more
- Support, Send Diagnostic Data
- Two options: collect and sendcollect only



Log Messages (1)

- **Application and service logs provide valuable troubleshooting information**
- **Logs are generated by Java Log4J**
- **Log4J levels identify what kind of information is being logged**
  - `TRACE`
  - `DEBUG`
  - `INFO` (default level for daemons)
  - `WARN`
  - `ERROR`
  - `FATAL`
  - `OFF`

Log Messages (2)

- **Example: JournalNode process killed**

```
2:34:18.157 PM  INFO  TransferFsImage  Sending fileName:
                /dfs/jn/my-name-service/current/…
2:34:36.603 PM  ERROR  JournalNode   RECEIVED SIGNAL 1: SIGHUP
2:34:36.612 PM  INFO  JournalNode   SHUTDOWN_MSG:
   /************************************************************
   SHUTDOWN_MSG: Shutting down JournalNode at worker-2.example.com/10.0.8.250
   ************************************************************/
2:34:36.617 PM  ERROR  JournalNode  RECEIVED SIGNAL 15: SIGTERM
6:41:49.975 AM  INFO  JournalNode  STARTUP_MSG:
   /************************************************************
   STARTUP_MSG: Starting JournalNode
   STARTUP_MSG:   user = hdfs
   …
```

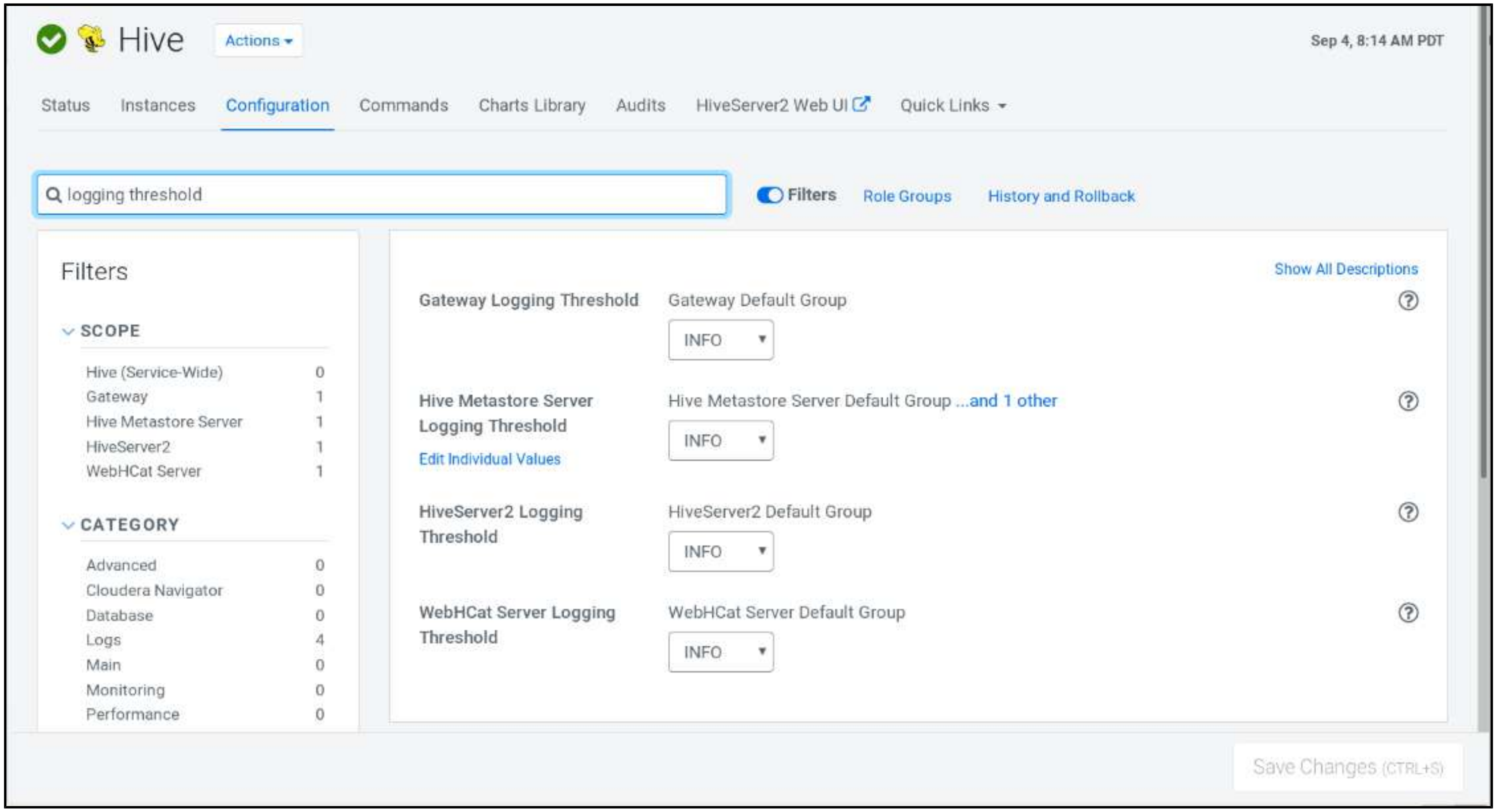
Options for Viewing Logs

- **Cluster daemon logs such as NameNode, DataNode, and NodeManager**
  - Cloudera Manager **Diagnostics > Logs**
    - Filter by time, log level, service, host, and keyword
  - Local log file on daemon host
    - `/var/log/`*`service-name-host-name`*`.log.out`
- **Cloudera Manager Server**
  - **Diagnostics > Server Logs**
- **YARN application logs such as Spark and MapReduce applications**
  - YARN ResourceManager UI
  - `yarn logs -applicationId` *`app-id`* command

Setting Daemon Log Levels

- **Set logging thresholds (levels) in Cloudera Manager**

Hive > Configuration



Java Stack Traces in Log Files

- **Shows the functions that were called when a run-time error occurred**
- **The top line is usually the most useful**
- **Example: MapReduce application out-of-memory error**

```
2019-03-11 18:51:13,810 FATAL [main] org.apache.hadoop.mapred.YarnChild:
Error running child : java.lang.OutOfMemoryError: Java heap space
at java.util.Arrays.copyOf(Arrays.java:2271)
at java.io.ByteArrayOutputStream.grow(ByteArrayOutputStream.java:113)
at java.io.ByteArrayOutputStream.ensureCapacity(ByteArrayOutputStream.java:93)
at java.io.ByteArrayOutputStream.write(ByteArrayOutputStream.java:140)
at java.io.OutputStream.write(OutputStream.java:75)
at HOTMapper.setup(HOTMapper.java:48)
at org.apache.hadoop.mapreduce.Mapper.run(Mapper.java:142)
at org.apache.hadoop.mapred.MapTask.runNewMapper(MapTask.java:764)
at org.apache.hadoop.mapred.MapTask.run(MapTask.java:340)
at org.apache.hadoop.mapred.YarnChild$2.run(YarnChild.java:165)
at java.security.AccessController.doPrivileged(Native Method)
at javax.security.auth.Subject.doAs(Subject.java:415)
```

Chapter Topics

Cluster Troubleshooting

- Overview
- Troubleshooting Tools
- **Misconfiguration Examples**
- Essential Points
- Hands-On Exercise: Troubleshooting a Cluster

Misconfigurations

- **Many Cloudera support tickets are due to misconfigurations**
- **This section provides some examples of a few configuration problems and suggested solutions**
- **These are just some of the issues that could occur**
  - And just a *few* possible causes and resolutions

MapReduce Task Out-of-Memory Error

- **Symptom**
  - A task fails with `Java heap space` error
- **Possible causes**
  - Poorly coded mapper or reducer
  - Map or reduce task has run out of memory
  - Memory leak in the code
- **Possible resolution**
  - Increase size of RAM allocated in `mapreduce.map.java.opts` and/or `mapreduce.reduce.java.opts`
  - Ensure `mapreduce.task.io.sort.mb` is smaller than RAM allocated in `mapreduce.map.java.opts`
  - Request the developer fix mapper or reducer

Not Able to Place Enough Replicas

```
WARN org.apache.hadoop.hdfs.server.namenode.FSNamesystem:
Not able to place enough replicas
```

- **Symptom**
  - Inadequate replication or application failure
- **Possible causes**
  - DataNodes do not have enough transfer threads
  - Fewer DataNodes available than the replication factor of the blocks
- **Possible resolutions**
  - Increase `dfs.datanode.max.transfer.threads`
  - Check replication factor

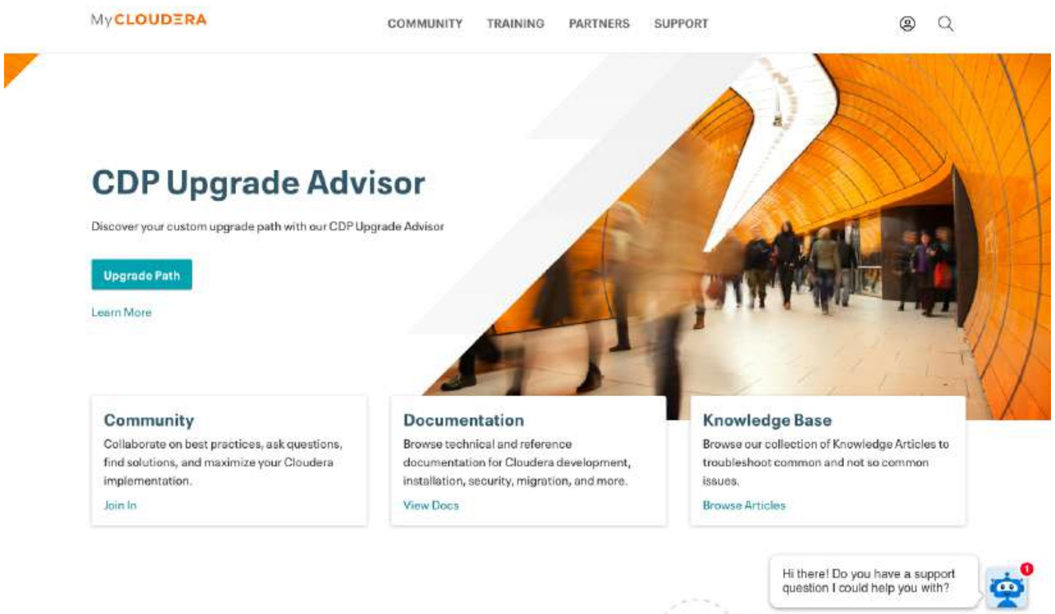
Where Did My File Go?

```
$ hdfs dfs -rm -r data
$ hdfs dfs -ls /user/training/.Trash
```

- **Symptom**
  - User cannot recover an accidentally deleted file from the trash
- **Possible causes**
  - Trash is not enabled
  - Trash interval is set too low
- **Possible resolution**
  - Set `fs.trash.interval` to a higher value

my.cloudera.com

- **We encourage everyone to signup for my.cloudera.com**
  - Community: Collaborate on best practices, ask questions, find solutions, and maximize your Cloudera implementation
  - Documentation: Browse technical and reference documentation for Cloudera development, installation, security, migration, and more
  - Knowledge Base: Browse our collection of Knowledge Articles to troubleshoot common and not so common issues



my.cloudera.com

- **Key information is available for customers such as:**
  - The size of their clusters
  - Version of our software used
  - Other cluster data
  - Upgrade Advisor
  - Validations: a new feature that enables customers to proactively identify cluster issues
  - The feature list of my.cloudera is ever improving

Chapter Topics

Cluster Troubleshooting

- Overview
- Troubleshooting Tools
- Misconfiguration Examples
- **Essential Points**
- Hands-On Exercise: Troubleshooting a Cluster

Essential Points

- **Troubleshooting Hadoop problems is a challenge, because symptoms do not always point to the source of problems**
- **Follow best practices for configuration management, benchmarking, and monitoring and you will avoid many problems**

Chapter Topics

Cluster Troubleshooting

- Overview
- Troubleshooting Tools
- Misconfiguration Examples
- Essential Points
- **Hands-On Exercise: Troubleshooting a Cluster**

Hands-On Exercise: Troubleshooting a Cluster

- **In this troubleshooting challenge, you will recreate a problem scenario, diagnose the problem, and fix the problem *if you have time***
- **Your instructor will provide direction as you go through the troubleshooting process**
- **Please refer to the Hands-On Exercise Manual for instructions**

Security

Chapter 16

Course Chapters

Security

After completing this chapter, you will be able to

- **Understand data governance with SDX**
- **Explain the objectives and main concepts of cluster security**
- **Identify security features available in Hadoop and the Cloudera's software**
- **Describe the role of Kerberos in securing a Hadoop cluster**
- **Summarize important considerations when planning cluster security**
- **Explain the basics of Ranger**
- **Explain the basics of Atlas**
- **Describe backup and recovery options**

Chapter Topics

Security

- **Data Governance with SDX**
- Hadoop Security Concepts
- Hadoop Authentication Using Kerberos
- Hadoop Authorization
- Hadoop Encryption
- Securing a Hadoop Cluster
- Apache Ranger
- Apache Atlas
- Backup and Recovery
- Essential Points

Security and Governance Challenges

Sharing data across workloads
- Requires multiple copies of data need to be created
- Each with its own set of data context

Burdensome admin effort
- Multiple clusters = multiple places to administer

One missing permission in one copy of the data can lead to **significant** financial and reputation risk

Difficult to **share data safely** for new analyses

Heavy new regulation such as **GDPR** makes the challenges even greater

Cloudera Data Platform with SDX

- **Benefits for IT infrastructure and operations**
  - Central control and security
  - Focus on curating not firefighting
- **Benefits for users**
  - Value from single source of truth
  - Bring the best tools for each joblistitem>

SDX Under the Hood: Security and Metadata

User management	Synchronize users and groups defined in corporate domain to CDP control plane through **SAML**
Network security	Kerberos, DNS, TLS enabled by default using a managed **FreeIPA** instance.
Encryption	Support for cloud storage native encryption at rest and in transit using TLS
SSO	Enable SSO access to all web UIs through **Apache Knox**
Storage security	Easily map corporate user identities to cloud IAM roles using **IDBroker** and Ranger's upcoming **RAZ**
Schema	Manage structured metadata using **Hive Metastore & Schema Registry**

SDX Under the Hood: Active Management

Stewardship

Replication

Workload Management

Encryption

Authorization	Restrict col / row / field access based on user, role, tag, or user attribute using **Apache Ranger**
Governance	Capture audit trail and lineage graphs (from ingest to model serving) using **Apache Atlas**
Data stewardship	Discover, manage, curate data assets with enhanced business metadata & workflows with **Data Catalog**
Data profiling	Automatically scan incoming data for sensitive fields and tag appropriately using **Data Catalog**
Workload management	Track workloads and jobs for easy troubleshooting, optimal performance using **WXM**
Replication	Move data along with all context across hybrid and multi-clouds using **Replication Manager**

SDX Video

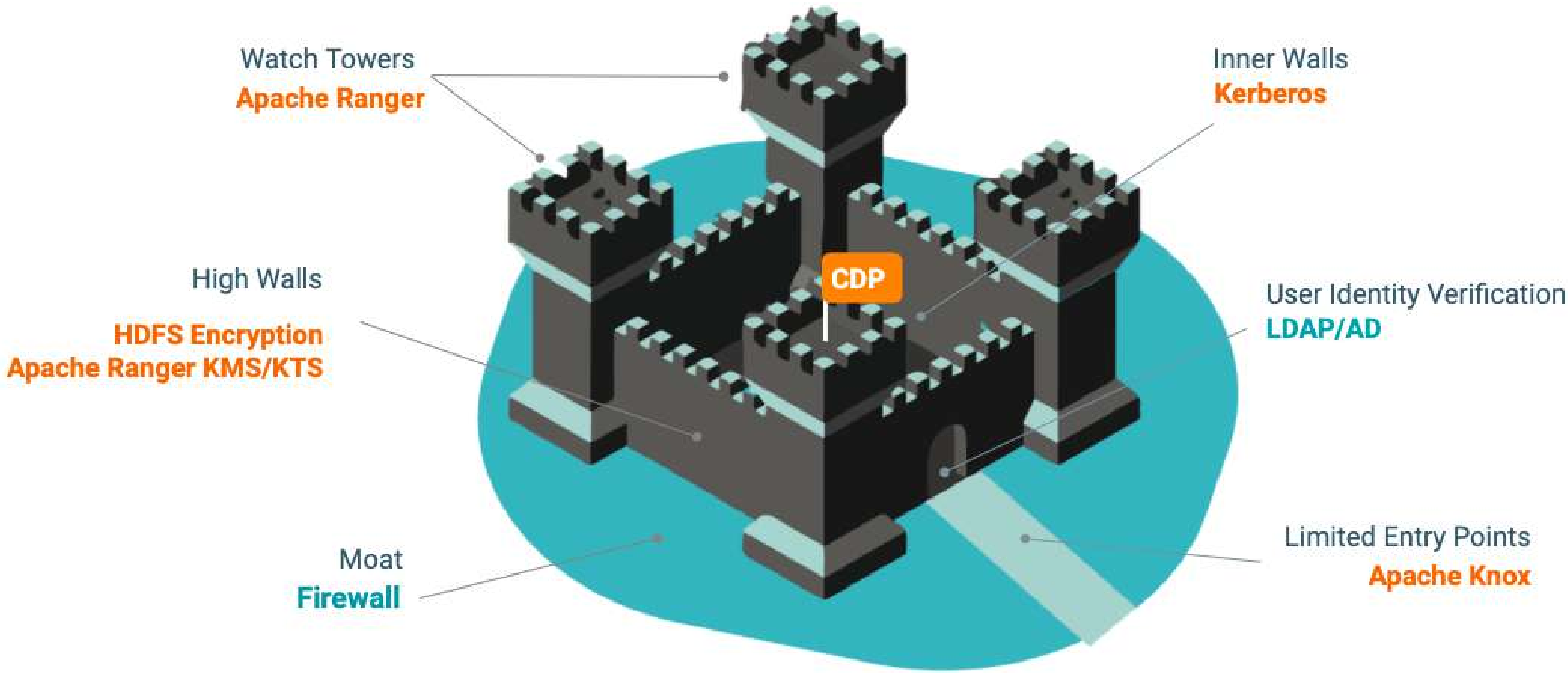
- **The instructor will run a video on: Why SDX?**
- **This video will introduce the need for a Shared Data Experience**
- **Click `here` for Video**

Consistent Security and Governance

- **Data Catalog:* a comprehensive catalog of all data sets, spanning on-premises, cloud object stores, structured, unstructured, and semi-structured**
- **Schema: automatic capture and storage of any and all schema and metadata definitions as they are used and created by platform workloads**
- **Replication: deliver data as well as data policies there where the enterprise needs to work, with complete consistency and security**
- **Security: role-based access control applied consistently across the platform. Includes full stack encryption and key management**
- **Governance: enterprise-grade auditing, lineage, and governance capabilities applied across the platform with rich extensibility for partner integrations**

* Not available in CDP Private Cloud or Private Cloud Base yet

CDP Security Landscape



Key aspects of SDX

- **Cloudera Shared Data Experience (SDX) is a framework used to simplify deployment of both on-premise and cloud-based applications allowing workloads running in different clusters to securely and flexibly share data**
- **Metadata:**
  - Build trusted, reusable data assets and efficient deployments by capturing not only technical and structural (schema) information but also operational, social, and business context
- **Security:**
  - Eliminate risk with complete data access security, delivering powerful access control policies that are defined once and applied consistently across all deployments
- **Governance:**
  - Prove compliance and reduce risk by characterizing data attributes and tracking its complete lineage across all analytics as well as cloud and data center deployments

Chapter Topics

Security

- Data Governance with SDX
- **Hadoop Security Concepts**
- Hadoop Authentication Using Kerberos
- Hadoop Authorization
- Hadoop Encryption
- Securing a Hadoop Cluster
- Apache Ranger
- Apache Atlas
- Backup and Recovery
- Essential Points

Important Security Terms

- **Security**
  - Computer security is a very broad topic
  - Access control and encryption are the areas most relevant to Hadoop
- **Authentication**
  - Confirming the identity of a participant
  - Typically done by checking credentials (such as username/password)
- **Authorization**
  - Determining whether a participant is allowed to perform an action
  - Typically done by checking an access control list (ACL)
- **Encryption**
  - Ensuring only verified users can access data
  - OS filesystem-level, HDFS-level, and network-level options

Chapter Topics

Security

- Data Governance with SDX
- Hadoop Security Concepts
- **Hadoop Authentication Using Kerberos**
- Hadoop Authorization
- Hadoop Encryption
- Securing a Hadoop Cluster
- Apache Ranger
- Apache Atlas
- Backup and Recovery
- Essential Points

What is Kerberos?

- Kerberos is a widely used protocol for network authentication
- Not part of CDP
- Included in Linux distributions
- Part of Microsoft Active Directory

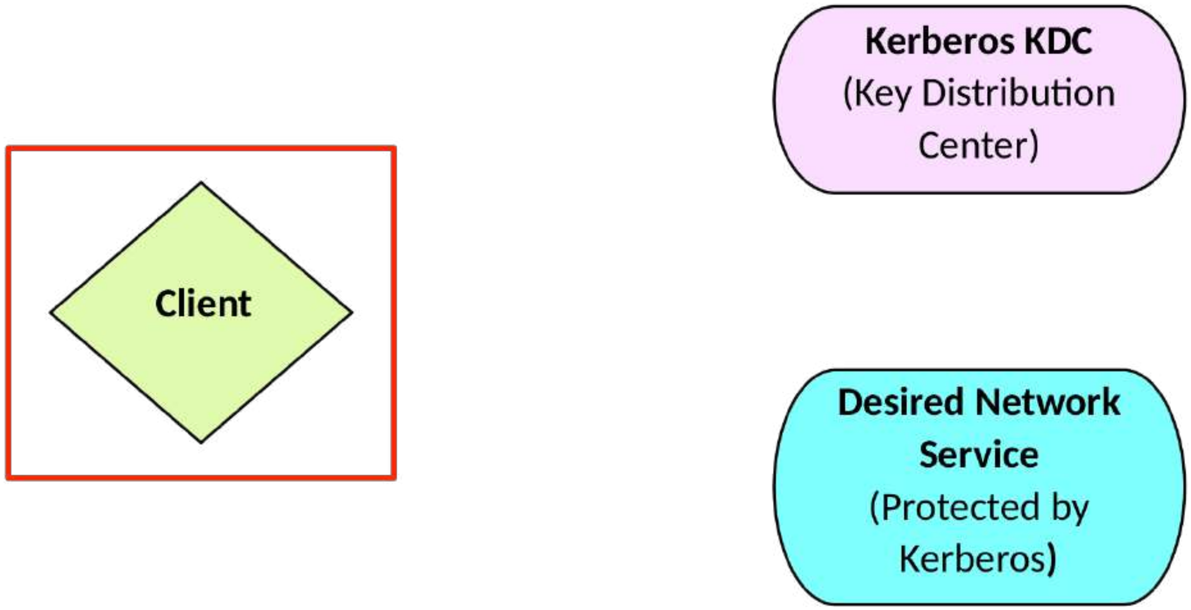
Kerberos and Hadoop

- **By default, Hadoop uses Linux usernames and passwords and group membership for authentication**
  - Example: HDFS file ownership and permissions
  - Easily subverted
  - Unreliable audit trail
- **Hadoop can use Kerberos to provide stronger authentication**
  - Hadoop daemons can use Kerberos to authenticate all remote procedure calls (RPCs)
- **Cloudera Manager simplifies configuring Kerberos on a cluster**

Kerberos Exchange Participants (1)

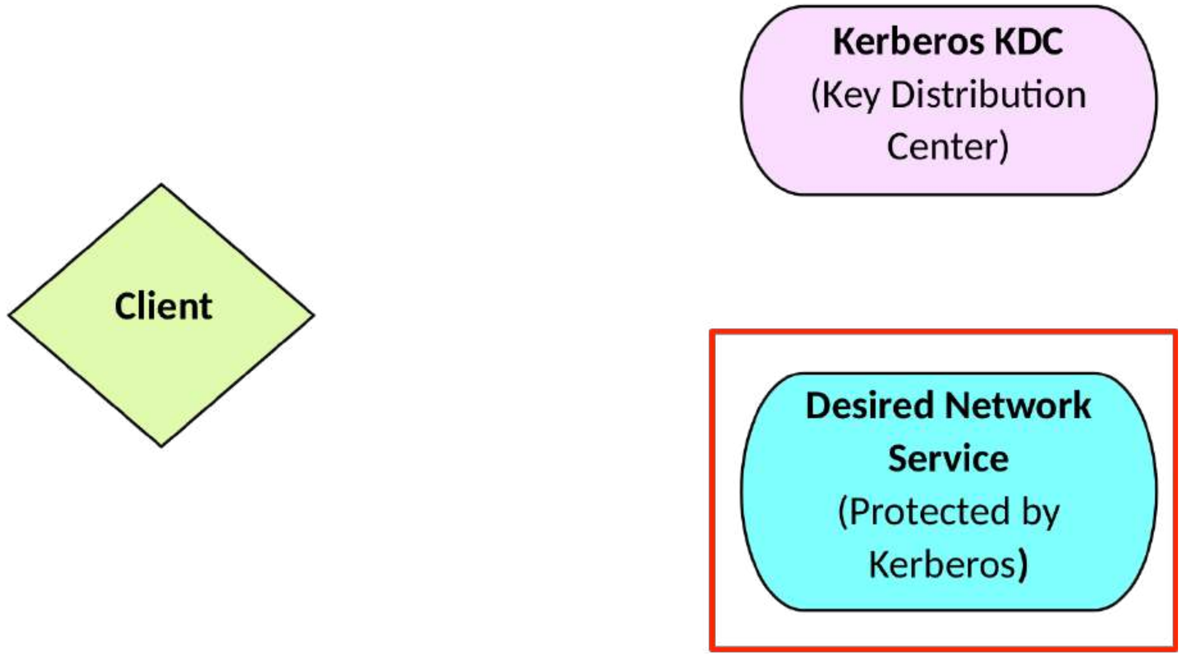
- **Kerberos involves messages exchanged among three parties**
  - The client
  - The server providing a desired network service
  - The Kerberos Key Distribution Center (KDC)

Kerberos Exchange Participants (2)



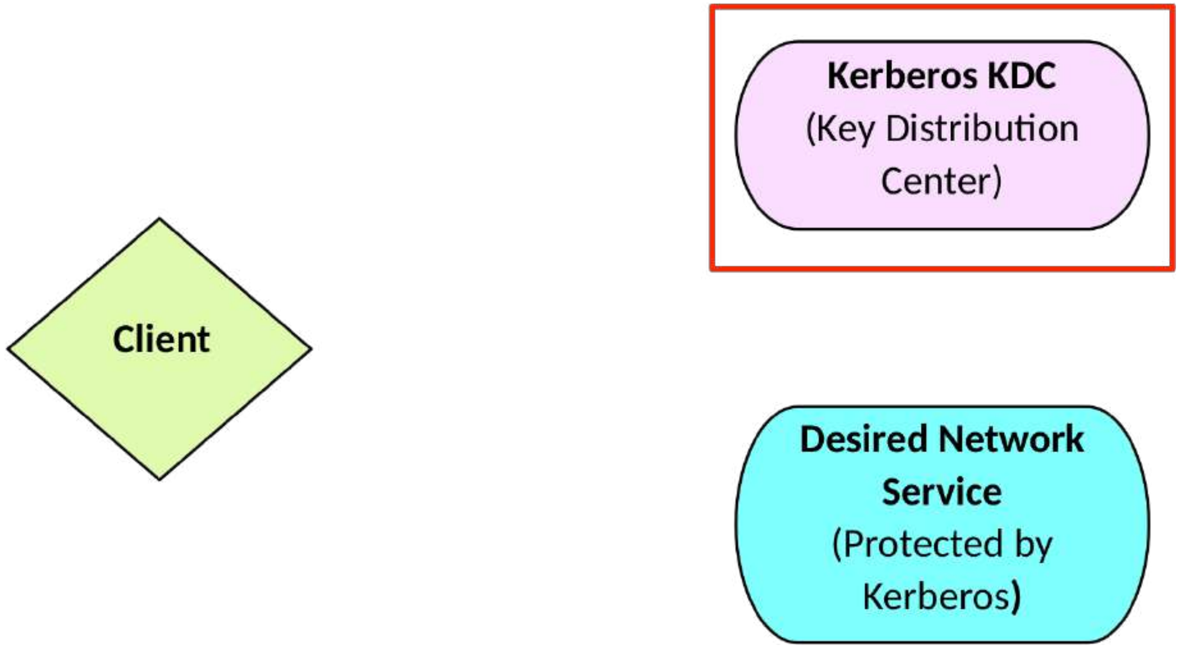
- **The client is software that desires access to a Hadoop service**
  - Such as a Spark application or the `hdfs dfs` command

Kerberos Exchange Participants (3)



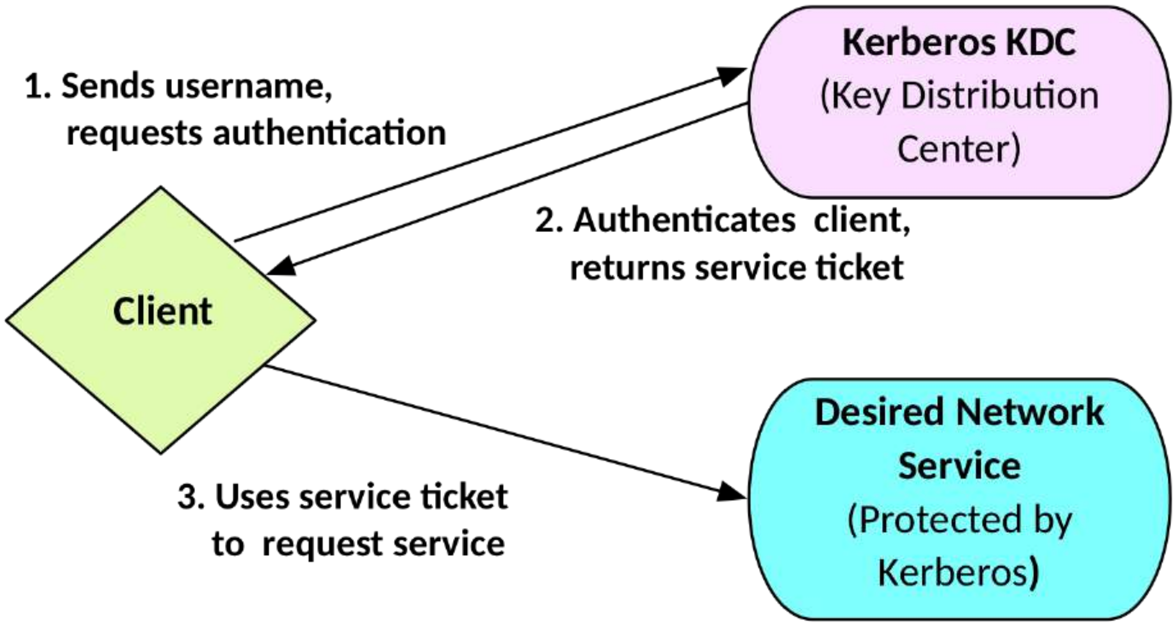
- **This is the service the client wishes to access**
  - For Hadoop, this will be a service daemon (such as the NameNode)

Kerberos Exchange Participants (4)



- **The Kerberos server (KDC) authenticates clients**

Kerberos Concepts (1)



- **Client requests authentication for a user principal**
- **Kerberos authenticates the user and returns a *service ticket***
- **Client connects to a service and passes the service ticket**
  - Services protected by Kerberos do not directly authenticate the client

Kerberos Concepts (2)

- **Authentication status is cached**
  - You do not need to submit credentials explicitly with each request
- **Passwords are not sent across the network**
  - Instead, passwords are used to compute encryption keys
  - The Kerberos protocol uses encryption extensively
- **Timestamps are an essential part of Kerberos**
  - Make sure you synchronize system clocks (NTP)
- **It is important that DNS reverse lookups work correctly**
  - Also be sure to use fully qualified domain names

Kerberos Terminology

- **Realm**
  - A group of services and users that a KDC deployment can authenticate
  - Similar in concept to a domain
  - By convention, the uppercase version of the DNS zone is used as the realm name
    - Example: `EXAMPLE.COM`
- **Principal**
  - A unique identity which can be authenticated by Kerberos
  - Can identify either a service or a user
  - Every user in a secure cluster will have a Kerberos principal
  - Example principals
    - User principal: `kjohnson@EXAMPLE.COM`
    - Service principal: `hdfs/node1.example.com@EXAMPLE.COM`

Kerberos Keytab Files

- **Kerberos uses *keytab* files**
  - Stores Kerberos principals and associated keys
  - Allows non-interactive access to services protected by Kerberos
- **Keytab files *must* be encrypted at rest and over the network**
  - Allows anyone to authenticate as the user or service *without credentials*

Chapter Topics

Security

- Data Governance with SDX
- Hadoop Security Concepts
- Hadoop Authentication Using Kerberos
- **Hadoop Authorization**
- Hadoop Encryption
- Securing a Hadoop Cluster
- Apache Ranger
- Apache Atlas
- Backup and Recovery
- Essential Points

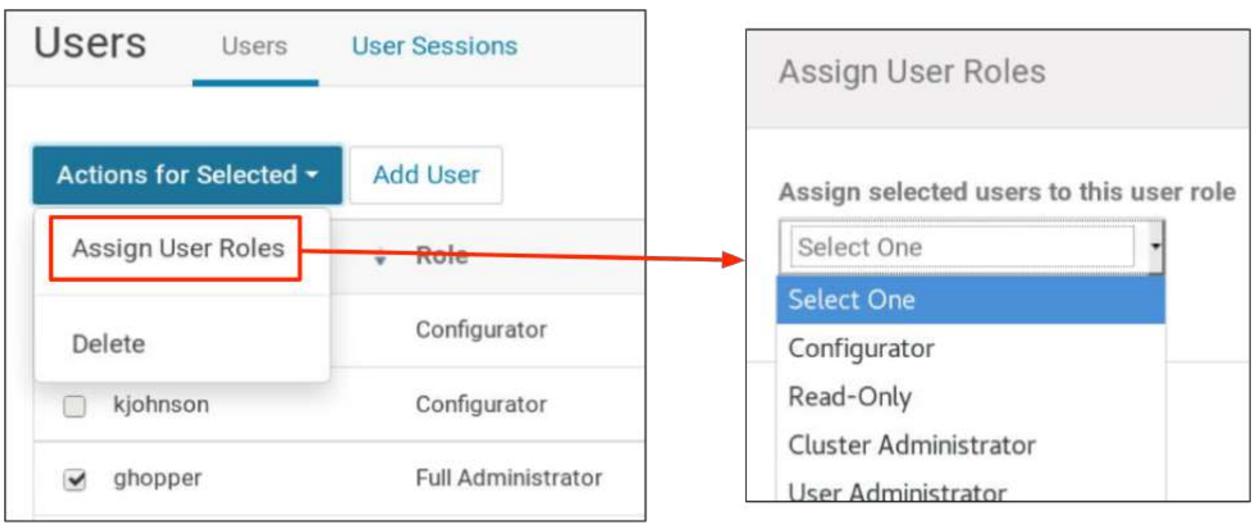
Hadoop Authorization

- **Access Control Lists (ACLs)**
  - YARN resource pools (queues)
    - Pools use ACLs to control who can run applications and use resources
  - YARN, Oozie, HBase, and Zookeeper use ACLs for administration
  - HDFS uses ACLs to control access to web UIs
- **HDFS file ownership and permissions**
  - Uses basic Linux-type file permissions or extended security with ACLs
- **Cloudera Manager**
  - User accounts are assigned roles that control access to administrative functions
- **Apache Ranger**
  - Manage policies for access to files, folders, databases, tables, and columns

Cloudera Manager

- **CM users are assigned roles that limit access, such as**
  - **Configurator**—configure services and dashboards
  - **Read-Only**—view but not change data in CM
  - **Cluster Administrator**—manage cluster
  - **User Administrator**—manage users and roles
  - **Full Administrator**—perform any action
  - **Operator**—start/stop services, commission/ decommission roles and hosts

Administration > Users and Roles > Roles



Chapter Topics

Security

- Data Governance with SDX
- Hadoop Security Concepts
- Hadoop Authentication Using Kerberos
- Hadoop Authorization
- **Hadoop Encryption**
- Securing a Hadoop Cluster
- Apache Ranger
- Apache Atlas
- Backup and Recovery
- Essential Points

Encryption Overview

- **Ensures that only authorized users can access, modify, or delete a dataset**
- **Uses digital keys to encode various components—text, files, databases, passwords, applications, or network packets**
- **Cloudera provides encryption mechanisms to protect data**
  - Data at rest
  - Data in transit

Filesystem Level Encryption

- **Operates at the Linux volume level**
  - Capable of encrypting cluster data inside and outside of HDFS
  - No change to application code required
- **Provides a transparent layer between the application and filesystem**
  - Reduces performance impact of encryption

Ranger Key Management Service

- **Ranger KMS provides a scalable cryptographic key management service for HDFS “data at rest” encryption**
- **Extends the native Hadoop KMS functionality by allowing system administrators to store keys in a secure database**
- **Administration of the key management server through the Ranger admin portal**
- **There are three main functions within the Ranger KMS**
  - Key management
  - Access control policies
  - Audit
- **Ranger KMS along with HDFS encryption are recommended for use in all environments**

Key Management—Navigator Key Trustee

- **Navigator Key Trustee**
  - The default HDFS encryption uses a local keystore
  - Navigator Key Trustee is a "virtual safe-deposit box" keystore server for managing encryption keys, certificates, and passwords
  - Encryption keys stored separately from encrypted data
  - Provides a high availability mode
- **In conjunction with the Ranger KMS, Navigator Key Trustee Server can serve as a backing key store for HDFS transparent encryption**

HDFS Level Encryption—"Data at Rest"

- **Provides transparent end-to-end encryption of data read from and written to HDFS**
  - No change to application code required
  - Data encrypted and decrypted only by the HDFS client
  - Keys are managed by the Key Management Server (KMS)
    - HDFS does not store or have access to unencrypted data or keys
- **Operates at the HDFS folder level**
  - Create *encryption zones* associated with a directory

Hadoop Network Level Encryption—"Data in Motion"

- **Transport Layer Security (TLS)**
  - Provides communication security over the network to prevent snooping
- **Configure TLS encryption for CDP services (HDFS, YARN, and so on)**
  - Example: Encrypt data transferred between different DataNodes and between DataNodes and clients
- **Configure TLS between the Cloudera Manager Server and Agents**
  - Level 1 (good): Encrypt communication between browser and Cloudera Manager, and between agents and Cloudera Manager Server
  - Level 2 (better): Add verification of Cloudera Manager Server certificate
  - Level 3 (best): Authentication of agents to the Cloudera Manager Server using certificates

Chapter Topics

Security

- Data Governance with SDX
- Hadoop Security Concepts
- Hadoop Authentication Using Kerberos
- Hadoop Authorization
- Hadoop Encryption
- **Securing a Hadoop Cluster**
- Apache Ranger
- Apache Atlas
- Backup and Recovery
- Essential Points

Configuring Hadoop Security

- **Hadoop security configuration is a specialized topic**
- **Specifics depend on**
  - Version of Hadoop and related components
  - Type of Kerberos server used (Active Directory or MIT)
  - Operating system and distribution
- **This course does not cover security configuration**
- **For more information**
  - *Cloudera Security Training* course
  - Cloudera Security documentation: `https://www.cloudera.com/documentation/enterprise/latest/topics/sg_edh_overview.html`

Securing Related Services

- **There are many ecosystem tools that interact with Hadoop**
- **Most require minor configuration changes for a secure cluster**
  - For example, specifying Kerberos principals or keytab file paths
- **Exact configuration details vary with each tool**
  - See tool documentation for details
- **Some require no configuration changes at all**
  - Such as Sqoop

Active Directory Integration

- **Microsoft Active Directory (AD) is a enterprise directory service**
  - Used to manage user accounts for a Microsoft Windows network
- **You can use Active Directory to simplify setting up Kerberos principals for Hadoop users**
- **Instructions in the CDP security documentation**

Hadoop Security Design Considerations

- **Isolating a cluster enhances security**
  - The cluster ideally should be on its own network
  - Limit access to those with a legitimate need
- **Setting up multiple clusters is a common solution**
  - One cluster may contain protected data, another cluster does not

Chapter Topics

Security

- Data Governance with SDX
- Hadoop Security Concepts
- Hadoop Authentication Using Kerberos
- Hadoop Authorization
- Hadoop Encryption
- Securing a Hadoop Cluster
- **Apache Ranger**
- Apache Atlas
- Backup and Recovery
- Essential Points

Apache Ranger

Apache Ranger is an open source application to define, administer, and manage security policies

- **Helps manage policies for access to files, folders, databases, tables, or columns**
  - You can set policies for individual users or groups
  - Policies are enforced consistently across CDP
- **Provides a centralized audit location**
  - Tracks all access requests in real time

Apache Ranger

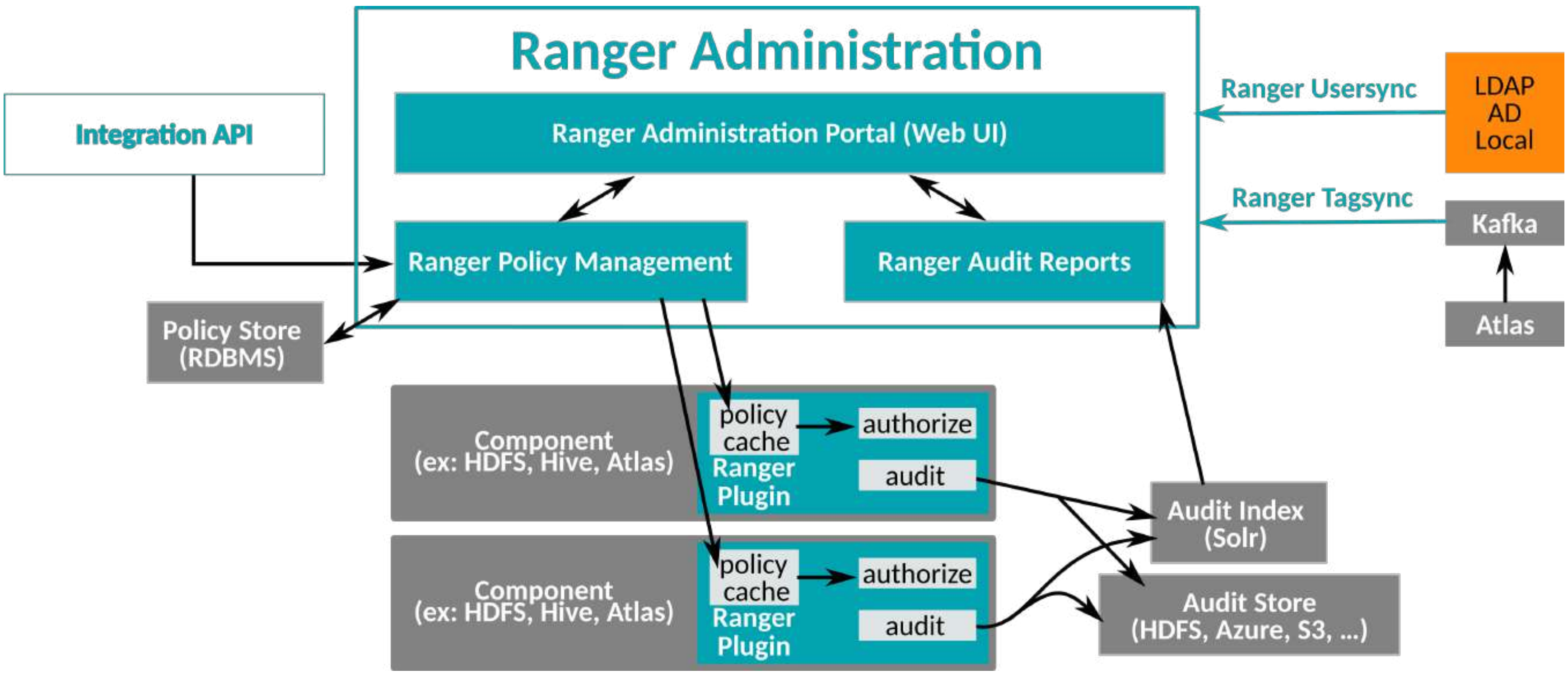
- Delivers a “single pane of glass” for the security administrator
- Centralizes administration of security policies
- Ensures consistent coverage across the entire CDP cluster
- Admins can administer permissions for specific LDAP-based groups or individual users
- Ranger is pluggable and can be easily extended to any data source using a service-based definition

What Ranger Does

- **Apache Ranger has a decentralized architecture with the following internal components**
  - Ranger policy portal and server
    - The central interface for security administration
    - Create and update policies stored in a policy database
    - Plugins within each component poll these policies at regular intervals
  - Ranger plugins
    - Lightweight Java programs
    - Embed within processes of each cluster component
    - Plugins pull policies from a central server and store them locally in a file
    - When a user request comes through the component, these plugins intercept the request and evaluate it against the security policy
  - User group sync
    - User synchronization utility to pull users and groups from Unix, LDAP AD
    - Stored within Ranger portal and used for policy definition

Apache Ranger Architecture Overview

- Users create and update policies using the Ranger Administrative Portal
- Plugins within each component poll policies at regular intervals
- Ranger collects audit data stored by the plugins

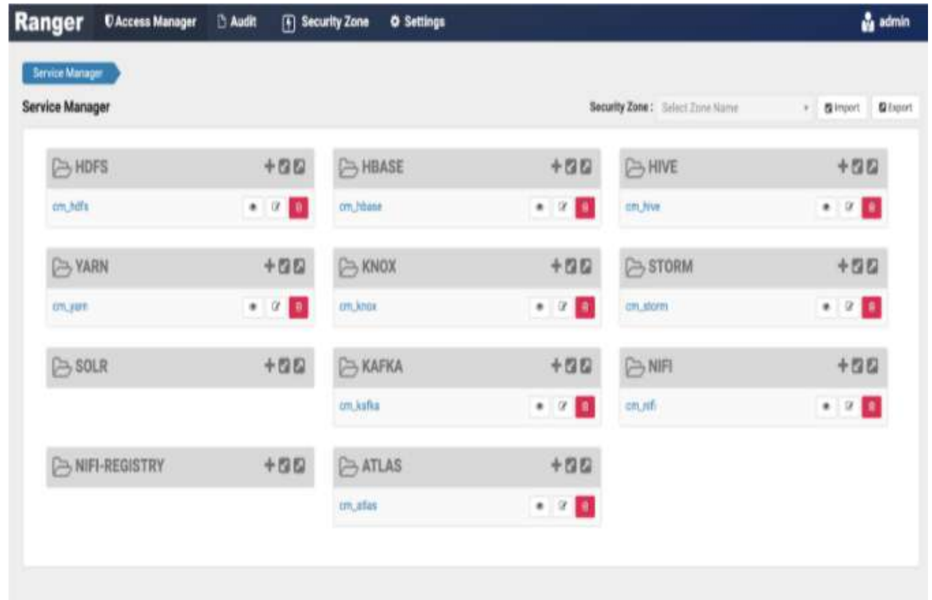


Ranger Basics

- Once a user has been authenticated, their access rights must be determined
- Authorization defines user access rights to resources
- For example, a user may be allowed to create a policy and view reports, but not allowed to edit users and groups
- Use Ranger to set up and manage access to CDP services and underlying data
- Ranger has two types of policies: resource-based and tag-based

Ranger Component Plugins

- Comprehensive coverage across CDP ecosystem components
- Plugins for components resident with component
- Plugin for authorizing other services outside CDP can be built (e.g. Presto, Kylin, Sqoop, WASB, Isilon/OneFS)

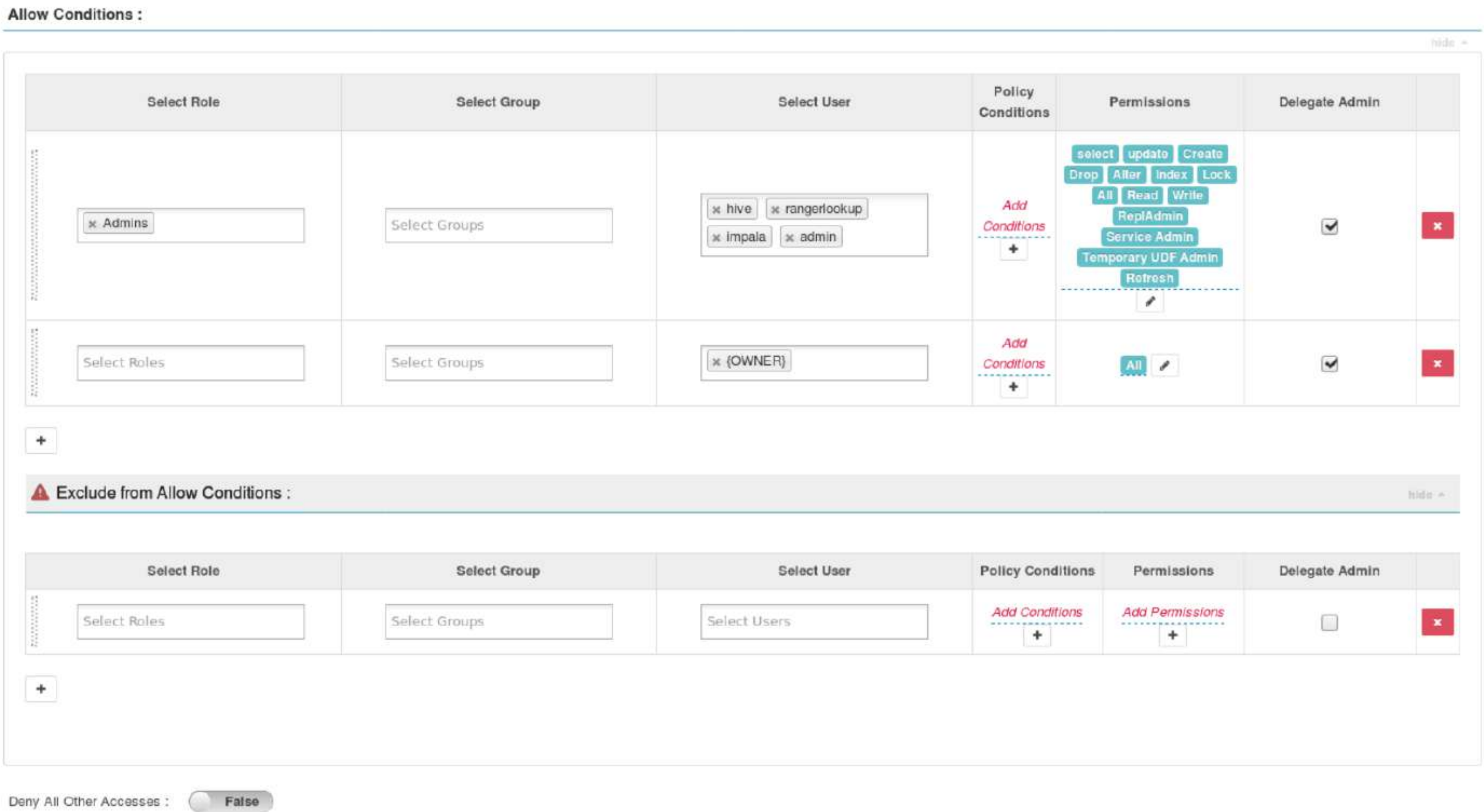


Policies in Apache Ranger

- **Policies provide the rules to allow or deny access to users**
  - All policies can be applied to a role, group, or particular user
- **Resource-based policies are associated with a particular service**
  - Identify who can use the resource
    - In general
    - To perform specific actions
    - To access specific assets
  - Create or edit through the plugin for the service
- **Attribute or tag based policies**
  - Restrict access using classifications and other attributes

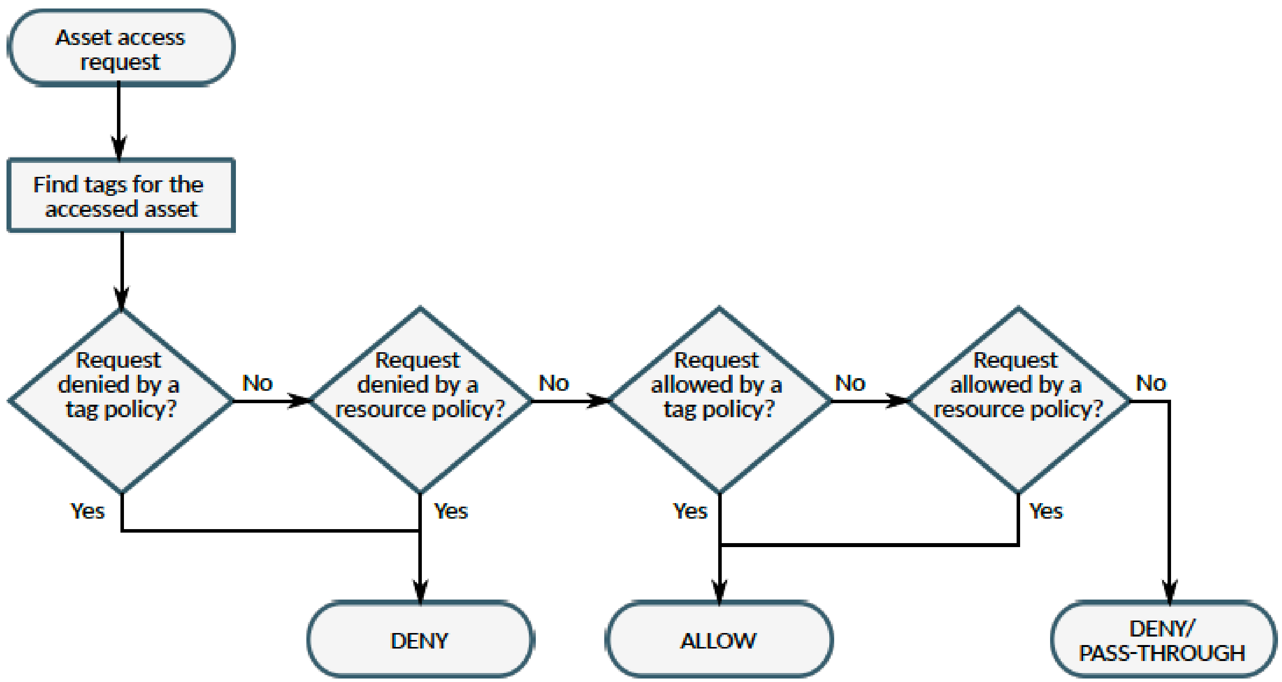
Allow and Deny Conditions

- **Policies are defined using allow and deny conditions**
  - Exceptions to those conditions also can be included



Policy Evaluation Flow

- **Deny conditions are checked first, then allow conditions**
  - This is opposite of the order they are presented on the policy page in Ranger



* Source: Tag Based Policies

Chapter Topics

Security

- Data Governance with SDX
- Hadoop Security Concepts
- Hadoop Authentication Using Kerberos
- Hadoop Authorization
- Hadoop Encryption
- Securing a Hadoop Cluster
- Apache Ranger
- **Apache Atlas**
- Backup and Recovery
- Essential Points

Regulatory Compliance

- **Regulations require implementing a way to keep track of data**
  - What data do you have?
  - Where is particular data held?

> **… the controller shall implement appropriate technical and organisational measures to ensure and to be able to demonstrate that processing is performed in accordance with this Regulation.**
> **—GDPR, Art. 24, Para. 1**

- **Using CDP to help organize your data allows you to find data quickly and easily**

Regulations

Compliance with regulations often requires careful work with data governance

- **General Data Protection Regulation (GDPR) for EU residents**
  - Residents have right to know how personal data is processed and why it is being taken
  - Residents have right to have all collected data erased
  - Residents have right to move their data from one company to a competitor
  - Companies must have data protection officer and protect data
  - Breach of data must be reported to the EU within 72 hours
- **Health Insurance Portability Accountability Act (HIPAA) for US residents**
  - Establishes administrative, physical, and technical safeguards for electronic Protected Health Information (ePHI)
  - Protects the privacy of individually identifiable health information
  - Promotes standardization, efficiency, and consistency
  - Requires notification following a breach of unsecured PHI

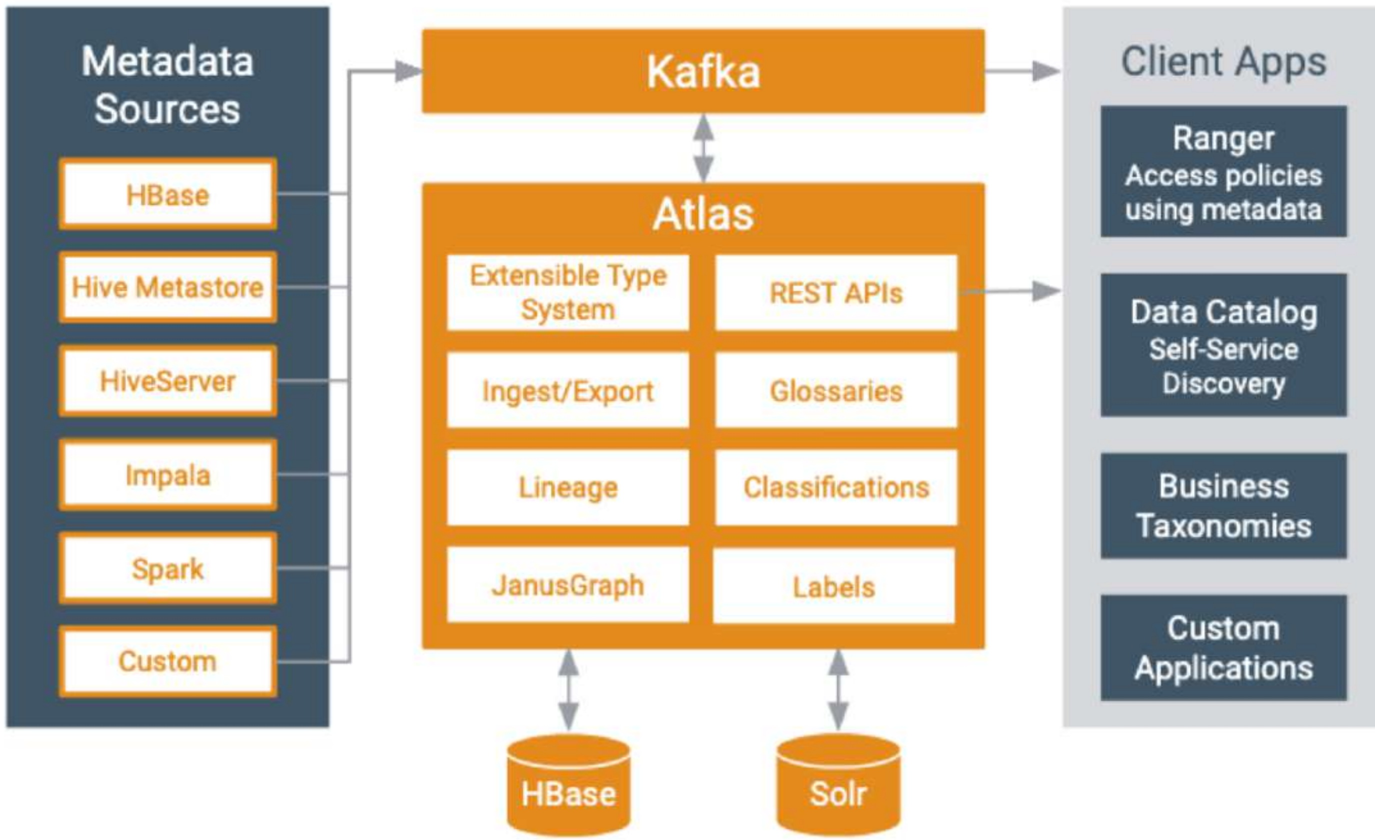
Atlas Overview

- **Atlas is a flexible system designed to exchange metadata with other tools and processes within and outside of the CDP**
- **Apache Atlas is developed around two guiding principles**
  - *Metadata Truth in Hadoop*: Atlas provides true visibility in CDP
  - *Developed in the Open*: Engineers from Aetna, Merck, SAS, Schlumberger, and Target are working together
- **Address compliance requirements through a scalable set of core governance services:**
  - *Data Lineage*: Captures lineage across Hadoop components at platform level
  - *Agile Data Modeling*: Type system allows custom metadata structures in a hierarchy taxonomy
  - *REST API*: Modern, flexible access to Atlas services, CDP components, UI and external tools
  - *Metadata Exchange*: Leverage existing metadata / models by importing it from current tools. Export metadata to downstream systems

Atlas Overview

- **A catalog for metadata of assets in an enterprise**
- **Dynamically create asset types with complex attributes and relationships**
- **Uses graph database to store asset type definitions and instances**
- **Over 100 out-of-box asset types, to cover following components**
  - HDFS, Hive, HBase, Kafka, Sqoop, Storm, NiFi, Spark, AWS S3, AVRO
- **Allows modeling of assets with complex attributes and relationship**
- **Enables capturing of data lineage**
- **Classifies assets for the needs of the enterprise.**
  - Types of classifications: (PII,PHI,PCI,PRIVATE,PUBLIC,CONFIDENTIAL)
- **Ranger integration - Classification-based access control**

Apache Atlas Architecture

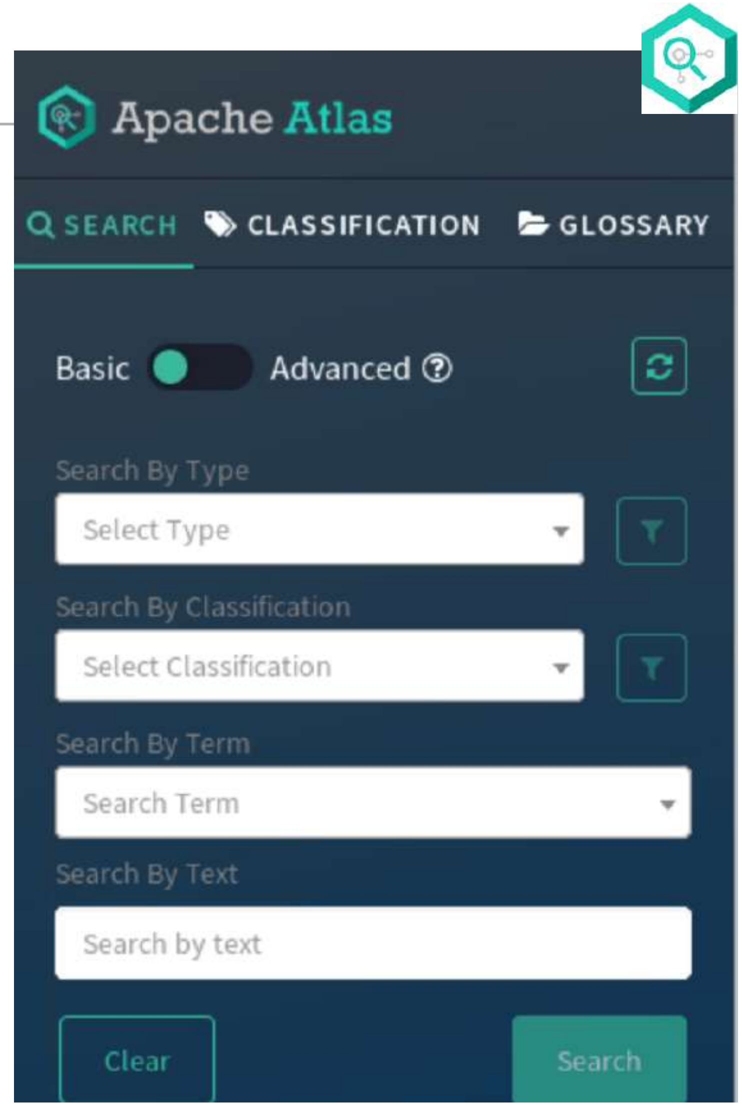


What is Apache Atlas?

- **Allows modeling of assets with complex attributes and relationship**
- **Enables capturing of data lineage**
- **Classifies assets for the needs of the enterprise.**
  - Types of classifications: (PII,PHI,PCI,PRIVATE,PUBLIC,CONFIDENTIAL)
- **Enables search for assets based on various criteria**
  - Search by classification and classification attributes
  - Search by asset attributes

Apache Atlas Basic Search

- **Apache Atlas provides easy search using different filters**
- **Enables search for assets based on various criteria**
  - Search by classification and classification attributes
  - Search by asset attributes



Chapter Topics

Security

- Data Governance with SDX
- Hadoop Security Concepts
- Hadoop Authentication Using Kerberos
- Hadoop Authorization
- Hadoop Encryption
- Securing a Hadoop Cluster
- Apache Ranger
- Apache Atlas
- **Backup and Recovery**
- Essential Points

Implement Backup Disaster Recovery

- **Backup and Disaster Recovery is a standard set of operations for many databases**
- **You must have an effective backup-and-restore strategy to ensure that you can recover data in case of data loss or failures**
- **Planning:**
  - Review disaster Recovery and resiliency requirements
  - Review data sources and volume projections
  - Implementation and test plan
  - Success criteria
- **Implementation**
  - BDR setup and configuration
  - Replication policies
  - Run BDR to copy data and metadata from the source to the target
  - Test the entire process

Backing Up Databases needed for CDP cluster

- **Cloudera recommends that you schedule regular backups of the databases that Cloudera Manager uses to store configuration, monitoring, and reporting data**
  - *Cloudera Manager* - Contains all the information about services you have configured and their role assignments, all configuration history, commands, users, and running processes (small database (less than 100 MB) AND the most important to back up)
  - *Reports Manager* - Tracks disk utilization and processing activities over time (medium-sized)
  - *Hive Metastore Server* - Contains Hive metadata (relatively small)
  - *Hue Server* - Contains user account information, job submissions, and Hive queries (relatively small).

Video: How CDP is Secure by Design

- **The instructor will run a video on: How CDP is Secure by Design**
- **This video will introduce design concepts used to secure CDP**
- **Click `here` for Video**

Chapter Topics

Security

- Data Governance with SDX
- Hadoop Security Concepts
- Hadoop Authentication Using Kerberos
- Hadoop Authorization
- Hadoop Encryption
- Securing a Hadoop Cluster
- Apache Ranger
- Apache Atlas
- Backup and Recovery
- **Essential Points**

Essential Points

- **Kerberos is the primary technology for enabling authentication security on the cluster**
  - Cloudera recommends using Cloudera Manager to enable Kerberos
- **Encryption can be enabled at the filesystem, HDFS, and network levels**
- **Transport Layer Security (TLS) provides security for “data in motion” over the network**
  - Such as communication between NameNodes and Datanode and between Cloudera Manager and agents
- **Utilize Atlas and Ranger to enhance cluster security**
- **Ensure a Backup and Recovery process is in place**

Quiz

1. **Which of the following is a definition of *authenticating*?**

   Checking an access control list

   Confirming the identity of the participant

   Determining whether a participant is allowed to perform an action

   Ensuring only verified users can access data

2. **Which is true about Kerberos?**

   It is a widely used protocol for network authentication

   It is not part of CDP

   It is part of Microsoft Active Directory

   All of the above

3. **Which of the following is *not* true about encryption?**

   Encryption ensures that only authorized users can access, modify, or delete a dataset

Cloudera provides encryption mechanisms to protect data at rest

Encryption uses digital keys to encode various component

Encrpytion requires use of Active Directory

4. Which is true about Apache Ranger?

It helps manage policies for access to files, folders, databases, tables or columns

It provides a central audit location

It tracks all access requests in real time

All of the above

Private Cloud / Public Cloud

Chapter 17

Course Chapters

Private Cloud / Public Cloud

After completing this chapter, you will be able to

- **Understand Private Cloud Capabilities**
- **Understand Public Cloud Capabilities**
- **Explore the uses of WXM**
- **Understanding the use of Auto-scaling**

Chapter Topics

Private Cloud / Public Cloud

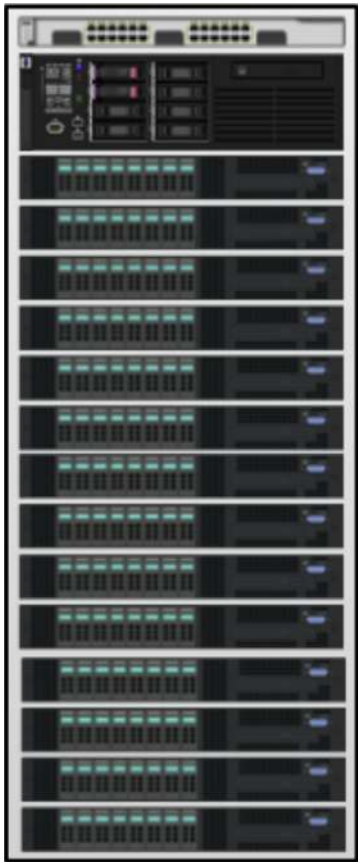
- **CDP Overview**
- Private Cloud Capabilities
- Public Cloud Capabilities
- What is Kubernetes?
- Workload XM Overview
- Auto-scaling
- Essential Points

The World's First Enterprise Data Cloud

- **Finally, a platform for both IT and the business, Cloudera Data Platform is:**
  - On-premises and public cloud
  - Multi-cloud and multi-function
  - Simple to use and secure by design
  - Manual or automated
  - Open and extensible
  - For data engineers and data scientists

Limitations of the Traditional Cluster Architecture

- **Colocation of storage and compute**
  - Can’t scale them independently
- **Optimized for large files**
  - Leads to the "small files" problem
- **Shared resource model for multitenancy**
  - Leads to "noisy neighbor" problem
- **Rigid mapping of services to nodes**
  - Distributes resources inefficiently



Key Aspects of the Cloud-Native Architecture

- **Fast networks enable separation of storage from compute**
  - This allows administrators to scale them independently
- **Object stores are the preferred way to store data**
  - This eliminates the "small files" problem
- **Containers decouple an application from the environment where it runs**
  - They provide isolation needed to solve the "noisy neighbor" problem
  - They also enable more efficient distribution of resources

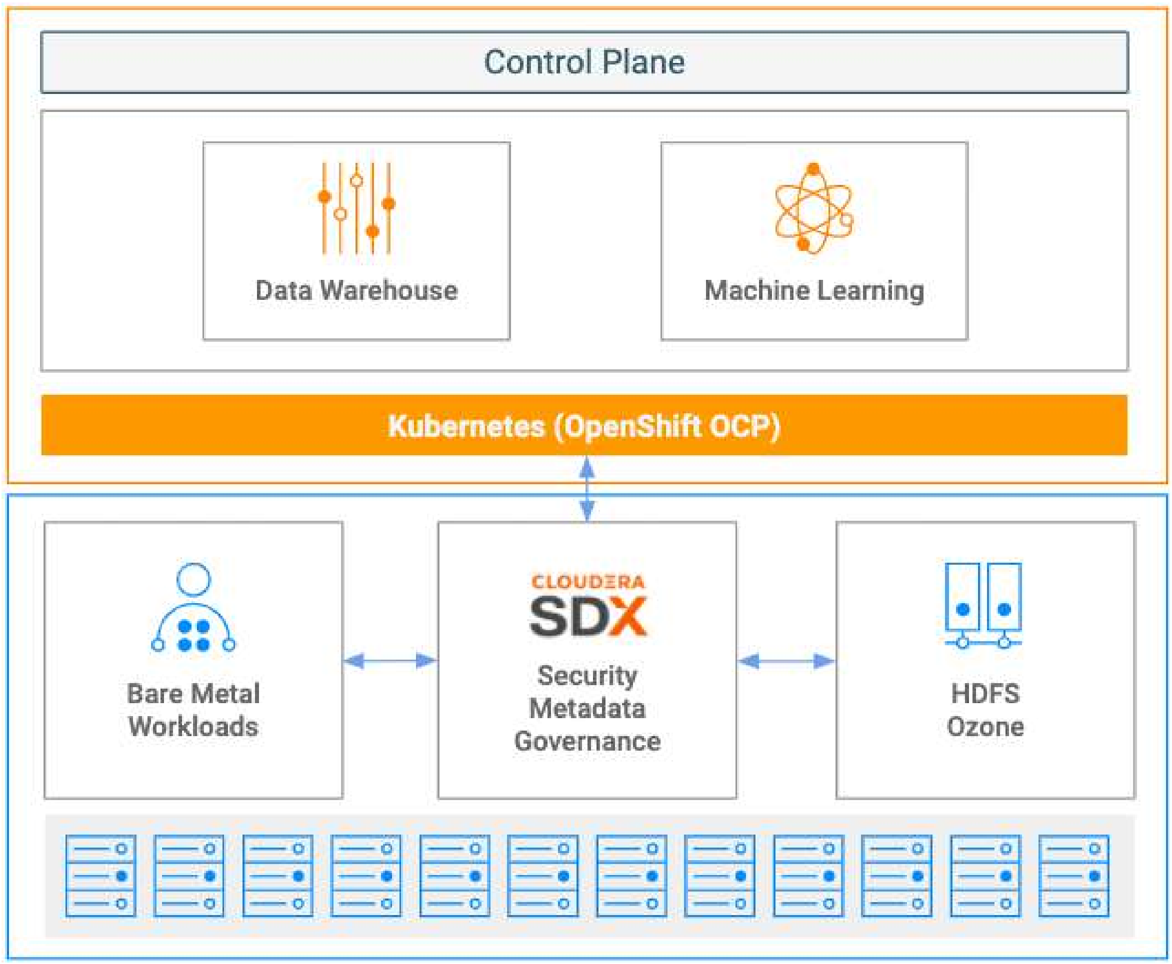
Comparing CDP Public and Private Cloud

Chapter Topics

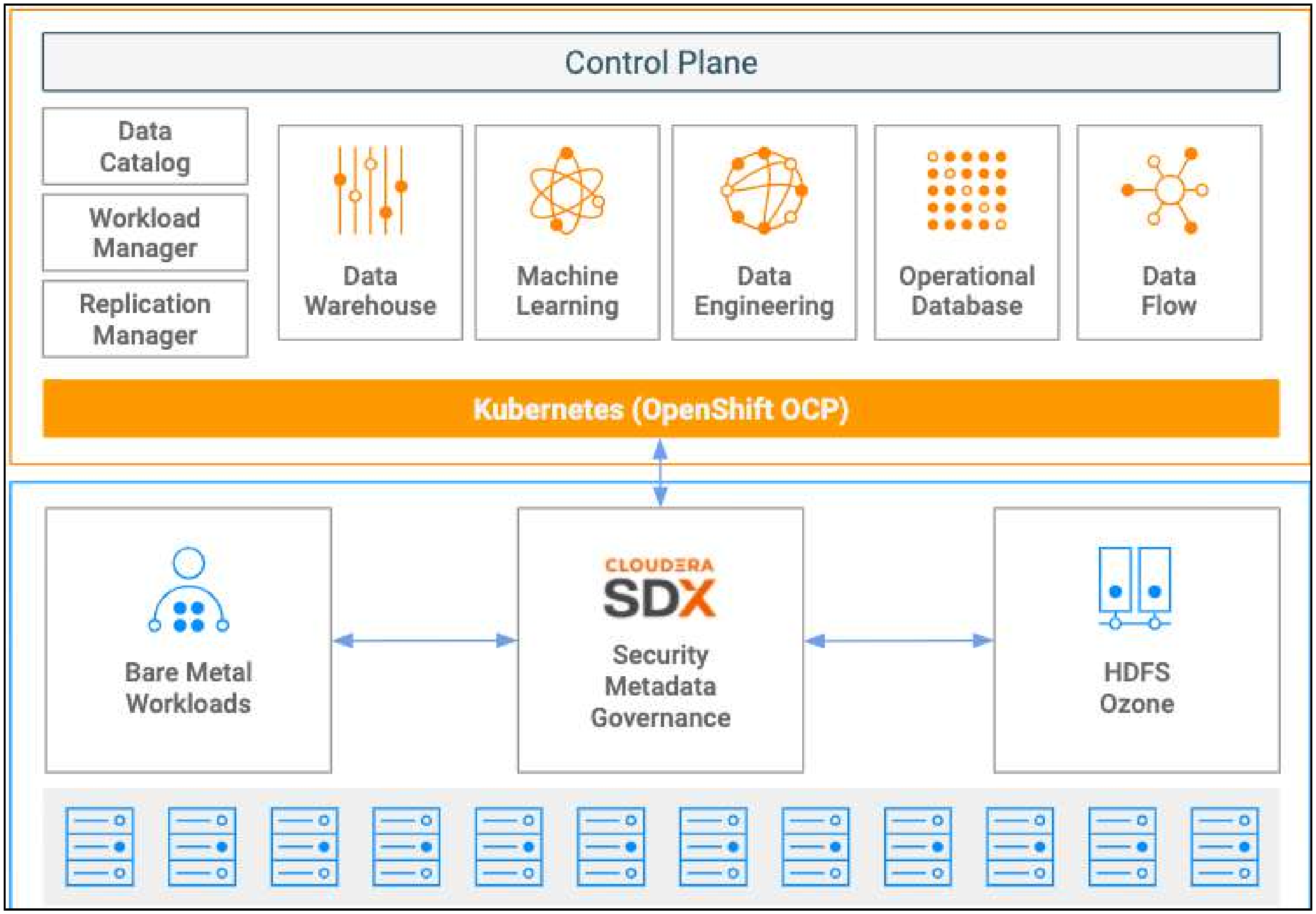
Private Cloud / Public Cloud

- CDP Overview
- **Private Cloud Capabilities**
- Public Cloud Capabilities
- What is Kubernetes?
- Workload XM Overview
- Auto-scaling
- Essential Points

CDP Private Cloud: Initial Release



CDP Private Cloud: Future State



Video: Demo of Private Cloud

- **The instructor will run a video on: Demo of Private Cloud**
- **Click `here` for Video**

Chapter Topics

Private Cloud / Public Cloud

- CDP Overview
- Private Cloud Capabilities
- **Public Cloud Capabilities**
- What is Kubernetes?
- Workload XM Overview
- Auto-scaling
- Essential Points

CDP Public Cloud

- **Create and manage secure data lakes, self-service analytics, and machine learning services without installing and managing the data platform software**
- **CDP Public Cloud services are managed by Cloudera**
- **Your data will always remain under your control in your VPC**
- **CDP runs on AWS and Azure, with Google Cloud Platform coming soon**

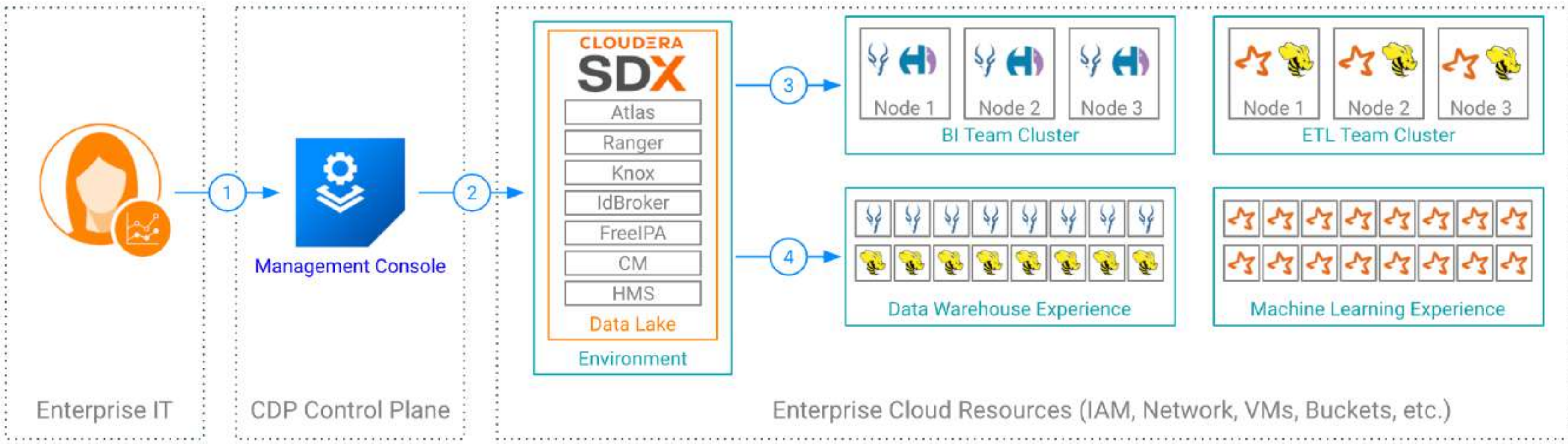
CDP Public Cloud

- **CDP Public Cloud lets you:**
  - Control cloud costs by automatically spinning up workloads when needed and suspending their operation when complete
  - Isolate and control workloads based on user type, workload type, and workload priority
  - Combat proliferating silos and centrally control customer and operational data across multi-cloud and hybrid environments
- **Check the documentation for complete details**

CDP Public Cloud Services

- **Data Hub**
  - Simplify building mission-critical data-driven applications with security, governance, scale, and control across the entire data lifecycle
- **Data Warehouse**
  - Unleash hybrid and multi-cloud data warehouse service for all modern, self-service, and advanced analytics use cases, at scale
- **Machine Learning**
  - Accelerate development at scale, anywhere, with self-service machine learning workspaces and the underlying compute clusters

Typical User Flow



Step 1
User connects to CDP with their enterprise identity

Step 2
They create an environment and data lake for their enterprise

Step 3
They create data hub clusters for traditional workloads

Step 4
They create access points for containerized analytic experiences

Video: Demomstration of Public Cloud

- **The instructor will run a video on: Private Cloud**
- **Click `here` for Video**

Chapter Topics

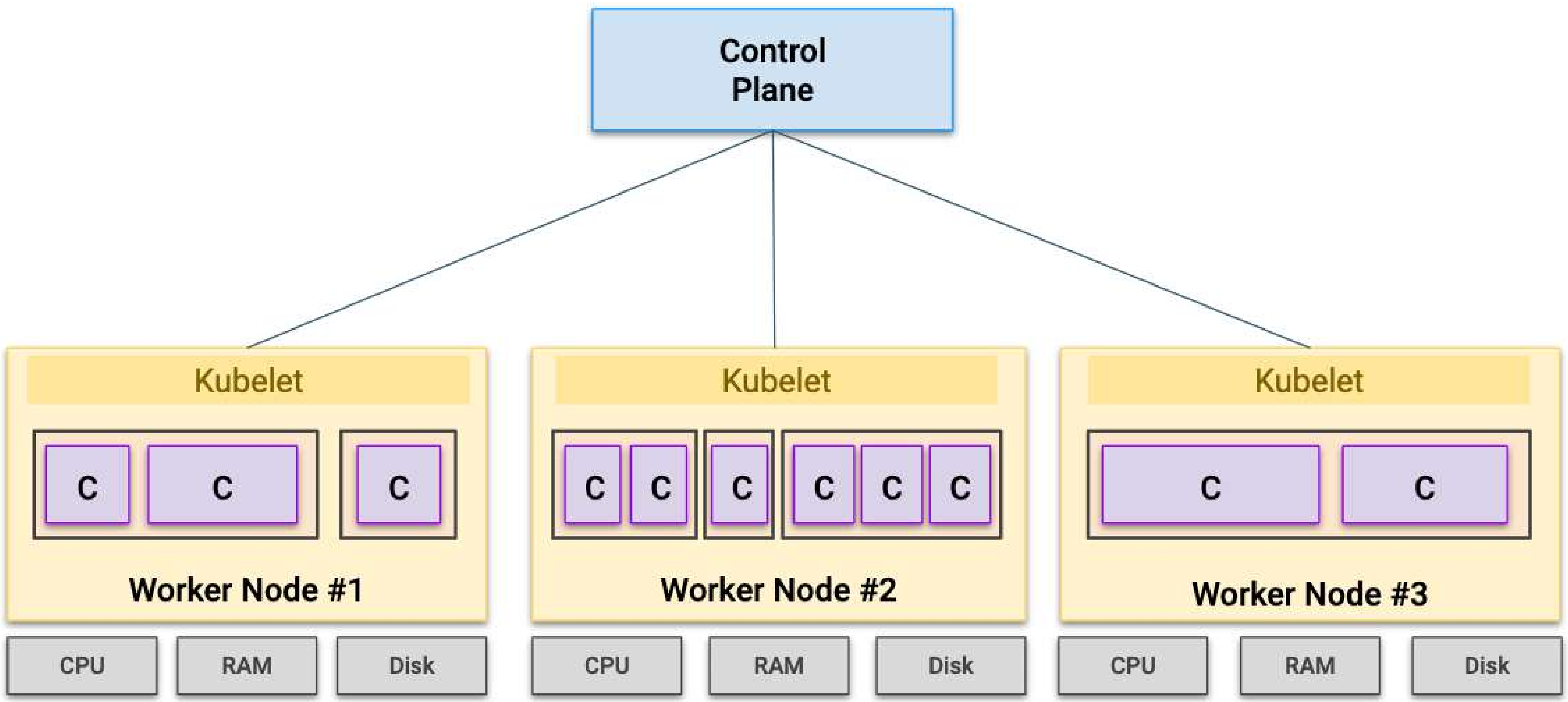
Private Cloud / Public Cloud

- CDP Overview
- Private Cloud Capabilities
- Public Cloud Capabilities
- **What is Kubernetes?**
- Workload XM Overview
- Auto-scaling
- Essential Points

What is Kubernetes

- Often abbreviated as k8s
- Software system used to deploy, scale, and manage containerized applications
- Originally developed at Google, now open source
- Supported by all major cloud providers and available in commercial products
- Collection of machines running Kubernetes software is called a "cluster"

Kubernetes Overview



* pods = black boxes and containers = C

Chapter Topics

Private Cloud / Public Cloud

- CDP Overview
- Private Cloud Capabilities
- Public Cloud Capabilities
- What is Kubernetes?
- **Workload XM Overview**
- Auto-scaling
- Essential Points

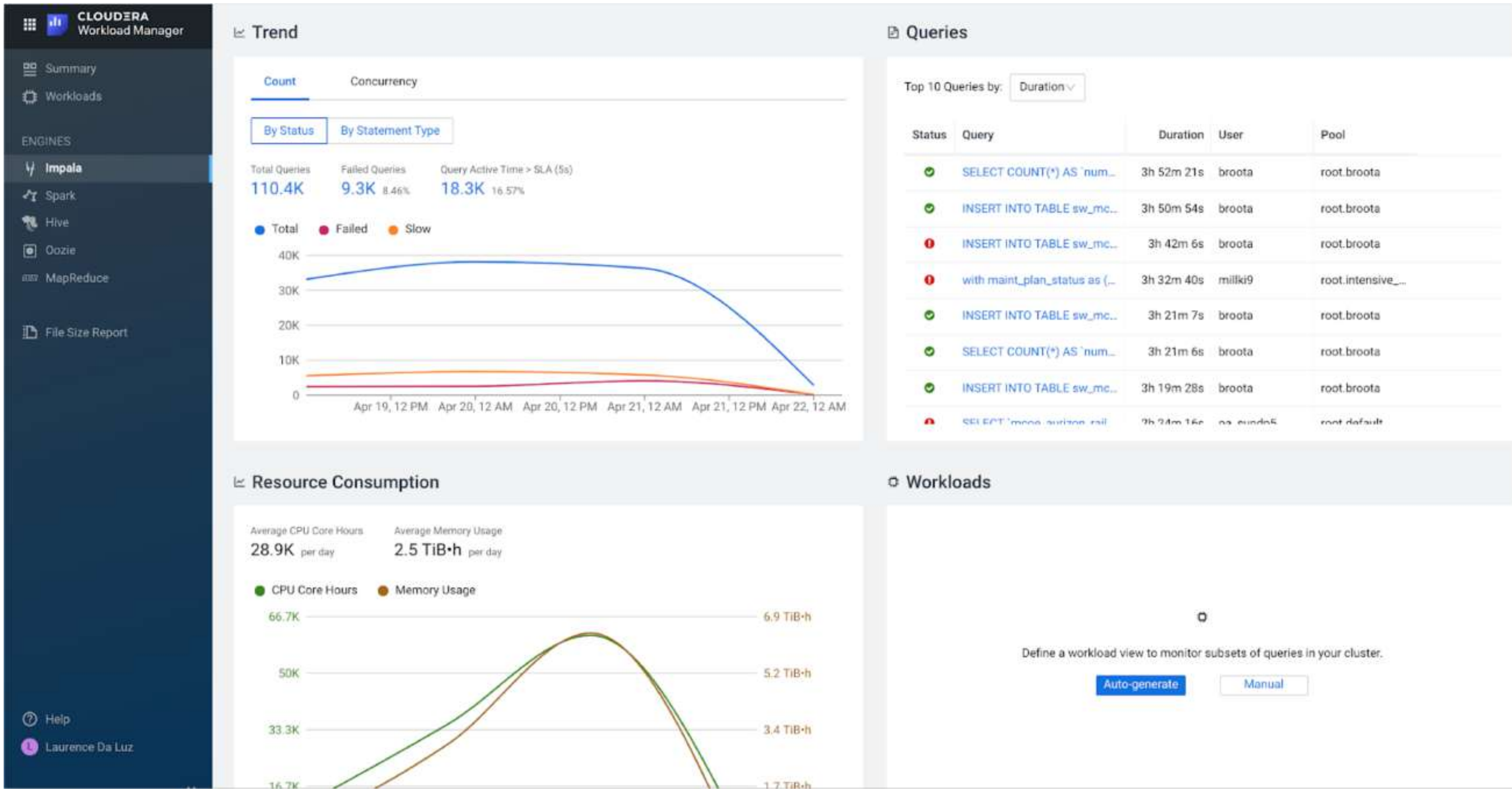
What is Workload XM (WXM)?

- WXM operates primarily as a Cloudera cloud managed service
- WXM can also be setup and configured on-prem with cloud-based components
- Once configured WXM receives a constant flow of data (Spark logs, Hive logs, Impala logs, etc.) from a connected cluster
- WXM analyzes each execution and calculates a history of metrics for each distinct job or query
- Each metric is statistically analyzed to identify outliers for key metrics
- Statistical outliers from multiple executions of the same job or query are then flagged as potential issues to be reviewed

How WXM Helps

- **Some issues are job or query specific**
  - Complex query
  - Querying on a partitioned table but not using a partition spec
- **Some issues are data specific**
  - Data held in too many files
  - Data skew
- **Some issues arise from configured limitations**
  - YARN queue limitations
  - Impala memory limitations
- **Some issues are cluster specific**
  - Cluster is busier today as compared to yesterday during the execution of today's job
  - Some cluster issues arise from contention across services such as heavy Solr indexing slowing HDFS reads or writes

Workload XM



CLOUDERA
Workload Manager
Summary
Workloads
ENGINES
Impala
Spark
Hive
Oozie
MapReduce
File Size Report
Help
Laurence Da Luz
Trend
Count
Concurrency
By Status
By Statement Type
Total Queries
110.4K
Failed Queries
9.3K 8.46%
Query Active Time > SLA (5s)
18.3K 16.57%
Total
Failed
Slow
Queries
Top 10 Queries by:
Duration
Status
Query
Duration
User
Pool
SELECT COUNT(*) AS `num...
3h 52m 21s
broota
root.broota
INSERT INTO TABLE sw_mc...
3h 50m 54s
3h 42m 6s
with maint_plan_status as (...
3h 32m 40s
milki9
root.intensive_...
3h 21m 7s
3h 21m 6s
3h 19m 28s
Resource Consumption
Average CPU Core Hours
28.9K per day
Average Memory Usage
2.5 TiB•h per day
CPU Core Hours
Memory Usage
Workloads
Define a workload view to monitor subsets of queries in your cluster.
Auto-generate
Manual

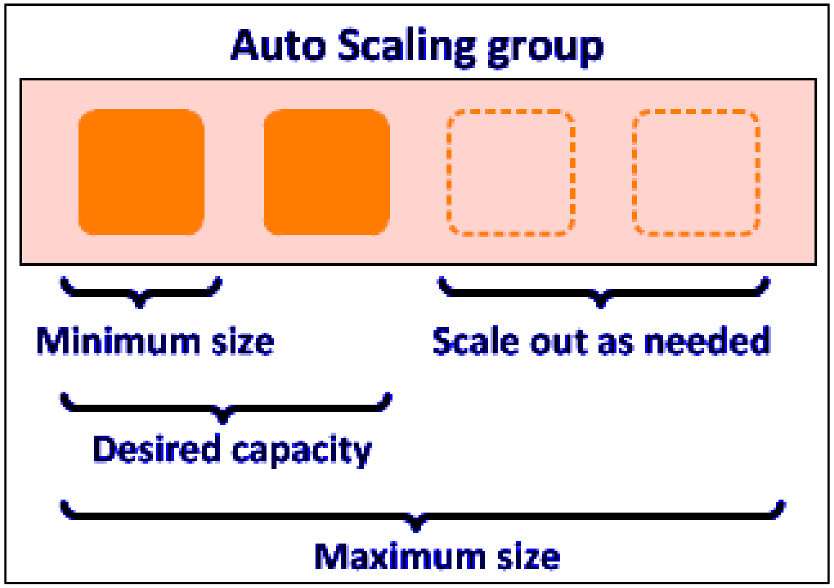
Chapter Topics

Private Cloud / Public Cloud

- CDP Overview
- Private Cloud Capabilities
- Public Cloud Capabilities
- What is Kubernetes?
- Workload XM Overview
- **Auto-scaling**
- Essential Points

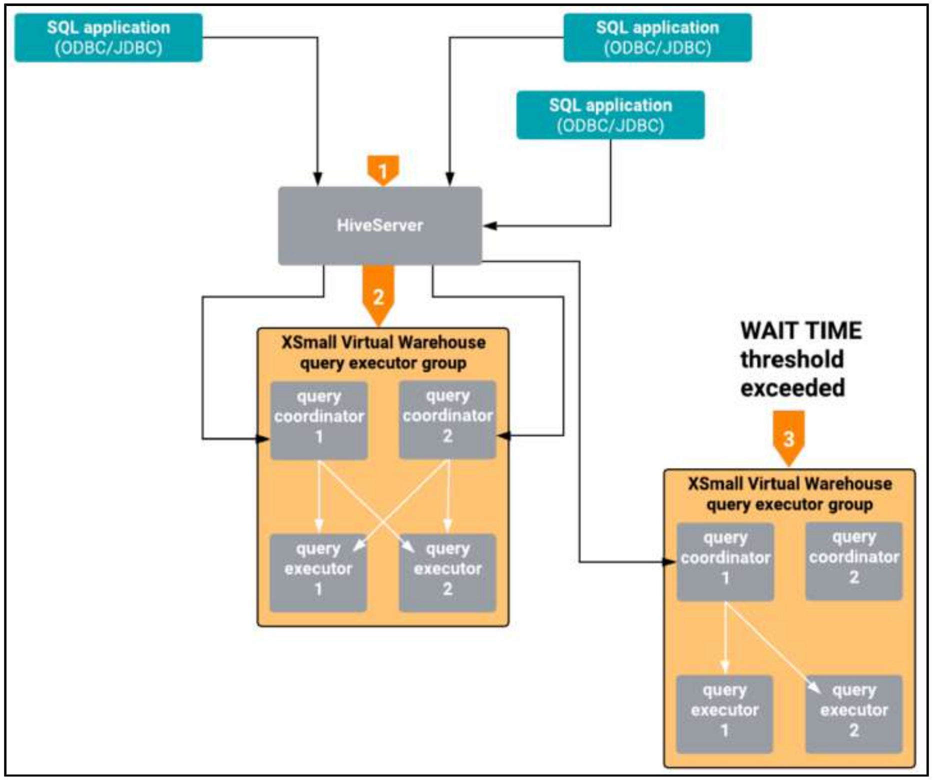
What is Auto-scaling?

- **Auto-scaling enables both scaling up and scaling down of Virtual Warehouse instances so they can meet your varying workload demands and free up resources on the OpenShift cluster for use by other workloads**



Auto-scaling process

- Hive Virtual Warehouse auto-scaling manages resources based on query load
- Depending on whether WAIT TIME has been set to manage auto-scaling, additional query executor groups are added when the auto-scaling threshold has been exceeded.



Chapter Topics

Private Cloud / Public Cloud

- CDP Overview
- Private Cloud Capabilities
- Public Cloud Capabilities
- What is Kubernetes?
- Workload XM Overview
- Auto-scaling
- **Essential Points**

Essential Points

- **Private Cloud clusters can enhance capabilities**
- **Public Cloud clusters can enhance capabilities**
- **K8s is a software system used to deploy, scale, and manage containerized applications**
- **WXM analyzes each execution and calculates a history of metrics for each distinct job or query**
- **Auto-scaling enables both scaling up and scaling down**

Conclusion

Chapter 18

Course Chapters

Course Objectives

During this course, you have learned

- **About the topology of a typical Cloudera cluster and the role the major components play in the cluster**
- **How to install Cloudera Manager and CDP**
- **How to use Cloudera Manager to create, configure, deploy, and monitor a cluster**
- **What tools Cloudera provides to ingest data from outside sources into a cluster**
- **How to configure cluster components for optimal performance**
- **What routine tasks are necessary to maintain a cluster, including updating to a new version of CDP**
- **About detecting, troubleshooting, and repairing problems**
- **Key Cloudera security features**

Class Evaluation

- **Please take a few minutes to complete the class evaluation**
  - Your instructor will show you how to access the online form

Which Course to Take Next?

Cloudera offers a range of training courses for you and your team

- **For administrators**
  - *Cloudera Security Training*
- **For developers**
  - *Developer Training for Spark and Hadoop*
  - *Cloudera Search Training*
  - *Cloudera Training for Apache HBase*
- **For data analysts and data scientists**
  - *Cloudera Data Analyst Training*
  - *Data Science at Scale using Spark and Hadoop*
- **For architects, managers, CIOs, and CTOs**
  - *Cloudera Essentials for Apache Hadoop*

Thank You!

- **Thank you for attending this course**
- **If you have any further questions or comments, please feel free to contact us**
  - Full contact details are on our Web site at `http://www.cloudera.com/`

Appendix: Cloudera Manager API

Chapter 19

Course Chapters

Appendix: Cloudera Manager API

After completing this chapter, you will be able to

- **Understand the Cloudera Manager API**
- **Understand the installation of CM API**
- **API Code Examples**

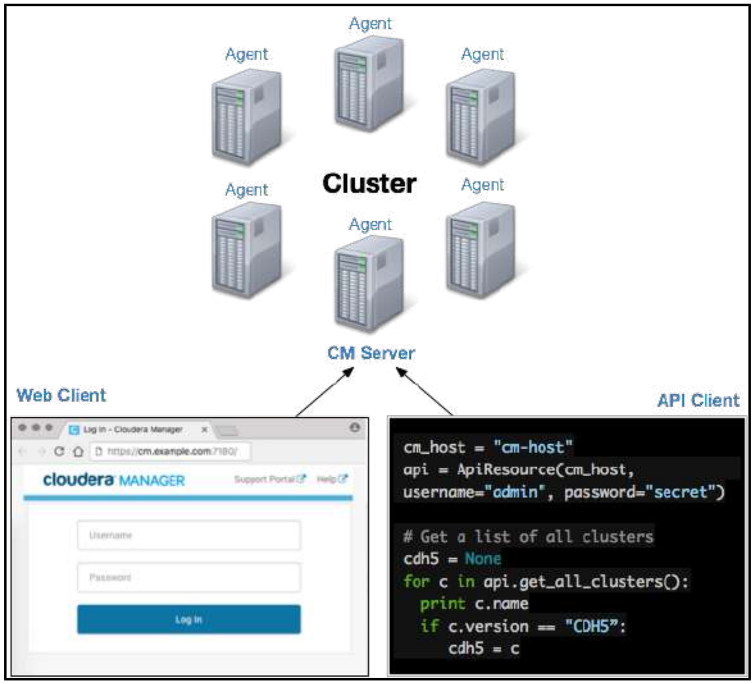
Chapter Topics

Appendix: Cloudera Manager API

- **Cloudera Manager API**
- Installation and Setup
- Code Examples
- Essential Points

Basic Cloudera Manager Concepts

- **Cloudera Manager**
  - Server
  - Agent
- **Client access**
  - Web UI
  - API



Supported Implementations

- **REST**
- **Language-specific client libraries**
  - Java
  - Python
- **Which to Choose?**

Features and Limitations

- **Features**
  - Manage multiple clusters
  - Deploy, expand or shrink clusters
  - Start, stop, or restart services or roles
  - Monitor and configure various cluster services and activities
  - Monitor user jobs
- **Limitations**
  - Not all features of CM are supported by API
  - Client bindings currently available in Python and Java only

Chapter Topics

Appendix: Cloudera Manager API

- Cloudera Manager API
- **Installation and Setup**
- Code Examples
- Essential Points

Installation: REST

- **System requirements**
  - HTTP client (typically curl)
- **Example call**

```
$ curl -u cloudera:cloudera 'http://cmhost.example.com:7180/
api/v17/clusters'
```

- **Helpful curl options to add when debugging**
  - `-i` (include headers)
  - `-v` (verbose output)

Using REST with TLS

- **Should "just work" if certificate signed by well-known CA**

```
$ curl -u 'cloudera:cloudera' 'https://
cmhost.example.com:7180/api/v17/clusters'
```

- **If not, you can disable certificate validation**

```
$ curl -u 'cloudera:cloudera' -k 'http://
cmhost.example.com:7180/api/v17/clusters'
```

- **Better yet, specify location of CA certificate to trust**

```
$ curl -u 'cloudera:cloudera' --cacert my_awesome_ca.pem
 'http://cmhost.example.com:7180/api/v17/clusters'
```

Installation: Python

- **Install via `pip`, first install pip**

```
$ sudo yum install python-pip
```

```
$ sudo pip install cm-api
```

- **CM API version defaults to latest at time of installation**

Chapter Topics

Appendix: Cloudera Manager API

- Cloudera Manager API
- Installation and Setup
- **Code Examples**
- Essential Points

Basic Example (REST)

- **Returns a list of clusters managed by a CM instance**
  - URL includes required credentials and API version

```
$ curl -u 'cloudera:cloudera' \
   'http://cmhost.example.com:7180/api/v17/clusters'
```

- **Some operations may require you to specify**
  - HTTP method (use −X option in curl)
  - Content-type header (use −H option in curl)
  - Input data (use −d option in curl)

Basic Example (Python)

- **This Python code invokes the same REST operation**

cdp-admin-chapi-basic-python.png

```
#!/usr/bin/env python

from cm_api.api_client import ApiResource

cm_host = "cmhost.example.com"
api = ApiResource(cm_host, username="cloudera", password="cloudera", version=17)

for c in api.get_all_clusters():
  print "Name=%s" % c.name
```

Python: Stopping or Starting Cluster (1)

```python
#!/usr/bin/env python

from cm_api.api_client import ApiResource

cm_host = "cmhost.example.com"
api = ApiResource(cm_host, username="cloudera", password="cloudera", version=17)

c = api.get_cluster('Cloudera Quickstart')
c.stop()   # ... or c.start()
```

Python: Stopping or Starting Cluster (2)

```python
#!/usr/bin/env python

from cm_api.api_client import ApiResource

cm_host = "localhost"
api = ApiResource(cm_host, username="cloudera", password="cloudera", version=17)

c = api.get_cluster('Cloudera Quickstart')
service = c.get_service('hbase')
cmd = service.stop()    # ... or cmd.start()
cmd = cmd.wait()
print "Successfully started? %s" % cmd.success
```

Add a New Service to Cluster

```python
#!/usr/bin/env python

from cm_api.api_client import ApiResource

cm_host = "localhost"
api = ApiResource(cm_host, username="cloudera", password="cloudera", version=17)
c = api.get_cluster('Cloudera Quickstart')

service = cluster.create_service("FLUME-1", "FLUME")        # Creating the Service

cmd = service.create_role("FLUME-AGENT-1","AGENT", hostid) # Adding the Service Roles

print "Service and role add successfully? %s" % cmd.success
```

Add a New Node to Cluster

```python
#!/usr/bin/env python

from cm_api.api_client import ApiResource

cm_host = "localhost"
api = ApiResource(cm_host, username="cloudera", password="cloudera", version=17)
c = api.get_cluster('Cloudera Quickstart')

hostlist = ["HostID"]  # list of new hostids
cmd = c.add_host(hostlist)

cmd = cmd.wait()
print "Host added successfully? %s" % cmd.success
```

Deleting a Node From Cluster

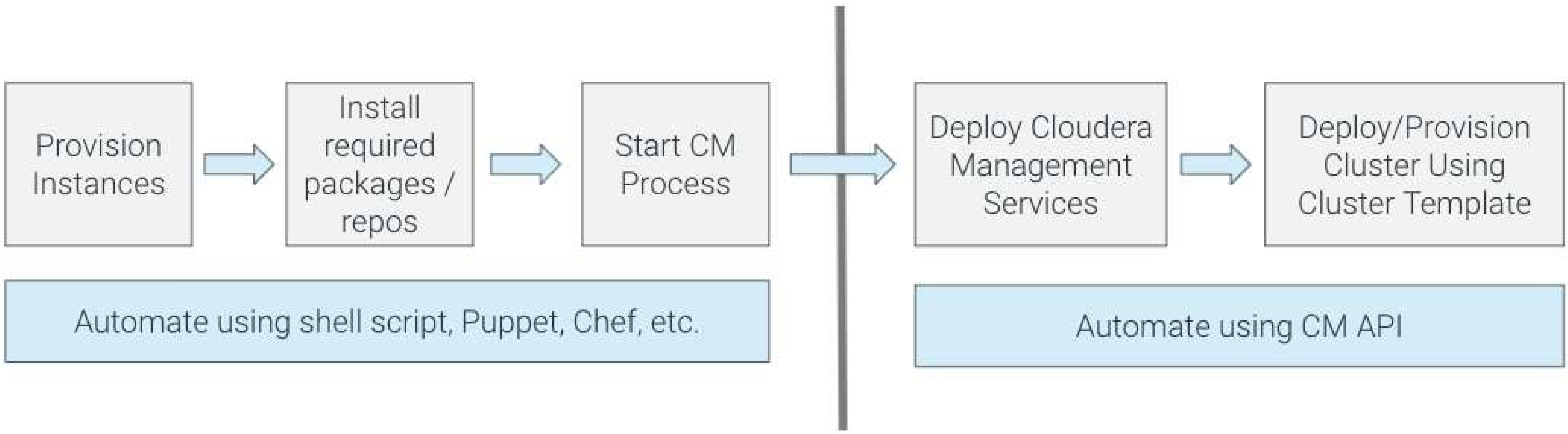
```python
#!/usr/bin/env python

from cm_api.api_client import ApiResource

cm_host = "localhost"
api = ApiResource(cm_host, username="cloudera", password="cloudera", version=17)
cm = ClouderaManager(api)
hosts = ["hostname1"]          # list of host to be removed
cm.host_decommission(hosts)

# Remove the roles prior to removal of host
cmd = api.delete_host("hostname1")
cmd = cmd.wait()
print "Host removal successfully? %s" % cmd.success
```

Cluster Deployment: Concept



Cluster Deployment: API Calls (1)

```python
#!/usr/bin/env python

import json
from cm_api.api_client import ApiResource
from cm_api.endpoints.types import ApiClusterTemplate
from cm_api.endpoints.cms import ClouderaManager
from cm_api.endpoints.services import ApiServiceSetupInfo

cm_host = "localhost"
api = ApiResource(cm_host, username="cloudera", password="cloudera", version=17)
cm = api.get_cloudera_manager()

# Add Cloudera Management Services
mgmt_setup_info= ApiServiceSetupInfo(name="MGMT",type="MGMT",config={})
mgmt_svc = cm.create_mgmt_service(mgmt_setup_info)
```

Cluster Deployment: API (2)

```
# Create/Assign Cloudera Management Services Roles
mgmt_svc.create_role("mgmt-SERVICEMONITOR", "SERVICEMONITOR", cm_host)

# Add other service roles
# mgmt_svc.create_role("mgmt-REPORTMANAGER","REPORTMANAGER", cm_host) and so on

# Update the service configs
with open("cls_config.json", 'r') as f:
clstr_config = json.load(f)

# Deploy the cluster - Using cluster template import
template = ApiClusterTemplate(api).from_json_dict(clstr_config, api)
cmd = cm.import_cluster_template(template)
```

Cluster Deployment using Templates

- **Get Existing Cluster Deployment Template ->** `http://localhost:7180/api/v17/cm/deployment`

```
{
  "cdhVersion":"",
  "displayName":"",
  "cmVersion":"",
  "repositories":"",
  "products": [{ "product":"", "version":"" }],

  "services": { "refName":"", "serviceType":"", "serviceConfigs":[{}],
                "roleConfigGroups": [{"refName":"", "roleType":"",
                "configs":[{}], "base":true }]},

  "hostTemplates":[{ "refName":"", "cardinality":1, "roleConfigGroupsRefNames":[] }],
  "instantiator": {"clusterName":"", "hosts":[
                    { "hostName":"", "hostTemplateRefName":"" }]}
}
```

Table of Important API Calls

Upgrade CDH in Cluster	`cluster.upgrade_cdh(deploy_client_config=True, start_all_services=True, cdh_parcel_version=PARCEL_VERSION, cdh_package_version=None, rolling_restart=False, slave_batch_size=None, sleep_seconds=None, slave_fail_count_threshold=None)`
Get Impala Queries	`# get impala Service using cluster.get_all_services call` `impala.get_impala_queries(start, end, filter_string, limit, i*limit)`
Get YARN Applications	`# get yarn Service using cluster.get_all_services call` `yarn.get_yarn_applications(start_time, end_time,filter_str='', limit=100, offset=0)`
Kill YARN Application	`yarn.kill_yarn_application(application_id)`
Upgrade Hive metastore	`# get Hive service using cluster.get_all_services call` `hive.upgrade_hive_metastore`
Update License	`# get cloudera_manager using ApiResource` `cm.update_license(license_file)`

Chapter Topics

Appendix: Cloudera Manager API

- Cloudera Manager API
- Installation and Setup
- Code Examples
- **Essential Points**

Essential Points

- **Cloudera Manager offers access through an API**
- **Capabilities provided through the API include**
  - Checking status of nodes and services
  - Starting, stopping, and restarting services
  - Adding or removing nodes and services
  - Provisioning a new cluster
- **API gives you an alternative to manual processes**

Appendix: Ozone Overview

Chapter 20

Course Chapters

Appendix: Ozone Overview

After completing this chapter, you will be able to

- **Understand the use of Ozone**
- **Working with Ozone**

Chapter Topics

Appendix: Ozone Overview

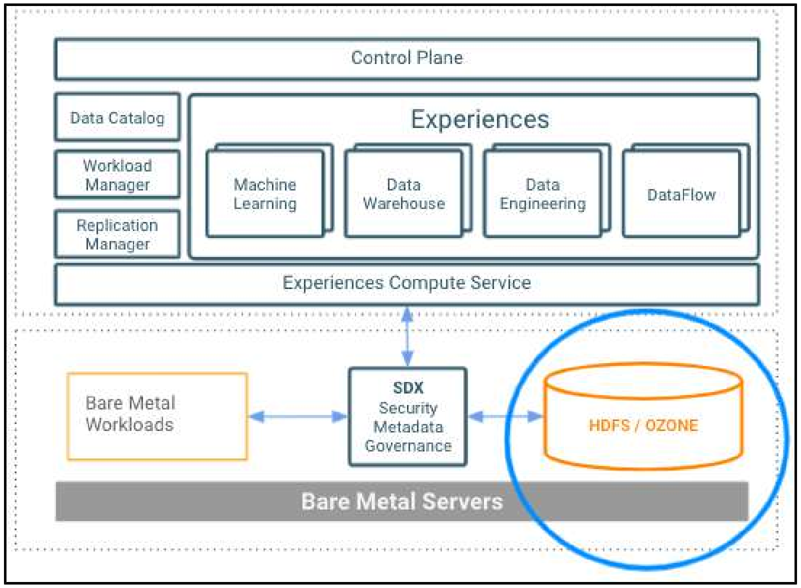
- **Ozone Overview**
- Working with Ozone
- Essential Points

Ozone Basics

- **Distributed key-value store**
- **Efficiently manages both small and large files**
- **Designed to work well with the existing Apache Hadoop ecosystem**
- **Open-source**
- **Scales to thousands of nodes and billions of objects in a single cluster**
- **Cloudera Manager**
  - Small files
  - Many files
  - Block Reports

Modern Big Data Object Storage

- Ozone is a distributed Key Value Object Store that provide 20x the scalability of traditional HDFS
- Increase storage node configurations by 350% reducing storage costs by 50% and cost per TB
- Ozone is designed and optimized for Big Data workloads providing the scale of a modern object store
- Native support for S3 API that enables cloud native architectures



Ozone Architecture

- **Layered architecture**
  - Key/Value Objects
  - Blocks
  - Containers
  - Buckets
  - Volumes
- **Separating HDFS Namenode's monolithic architecture**
  - Managing Namespace (Ozone Manager (OM))
  - Block Storage (Storage Container Management (SCM))
- **Simplified architecture for High Availability**
  - No more Zookeeper, Failover Controllers and Journal Nodes needed

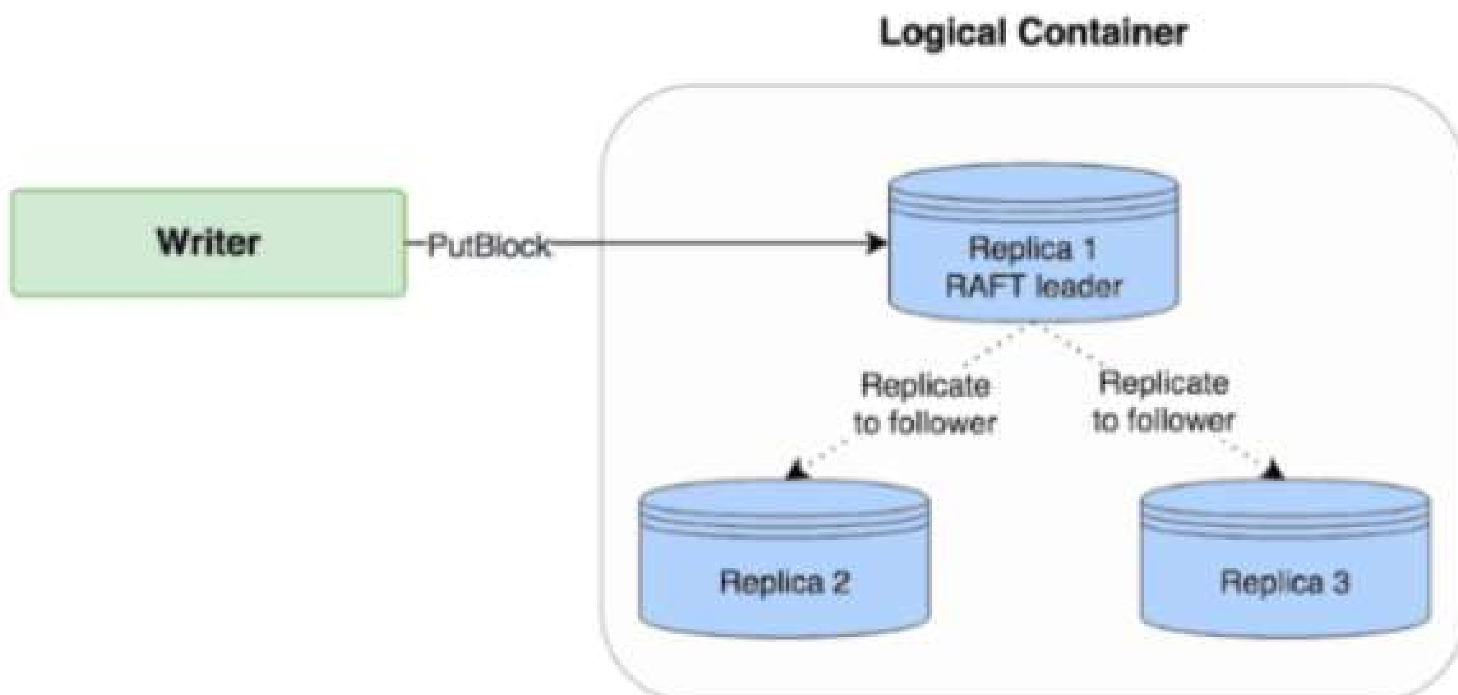
Key/Values, Blocks and Containers

- **Lowest Layer of Ozone**
  - Key/Value objects can be stored, read and deleted
  - Multiple Blocks get stored into a container

BlockName 1	... BlockData ...
BlockName 2	... BlockData ...
BlockName N	... BlockData ...

Containers

- **Lowest Layer of Ozone**
- **Default container size: 5 GB**
- **Containers get replicated across multiple DataNodes using RAFT protocol**

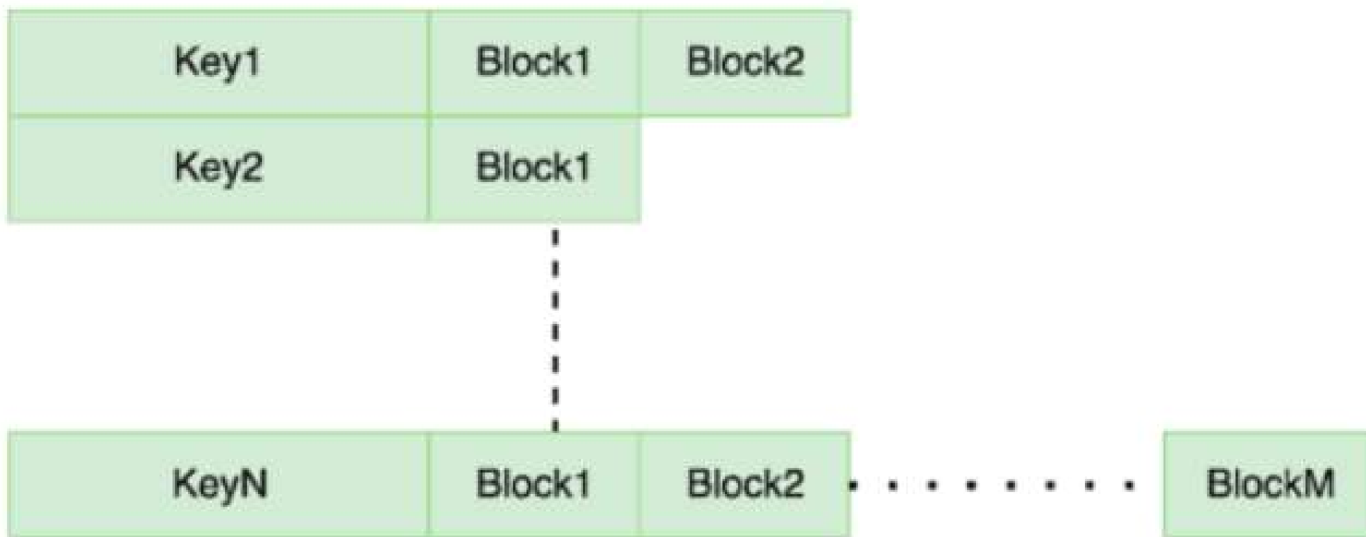


Storage Container Manager (SCM)

- Very important component of ozone
- Manages container locations and replication
- SCM offers block and container-based services to Ozone Manager
- Redundant and distributed architecture (3 or more SCMs)
- Detects and manages Over/Under-replication
- Containers + RAFT + SCM = HDDS (Hadoop Distributed DataStore)

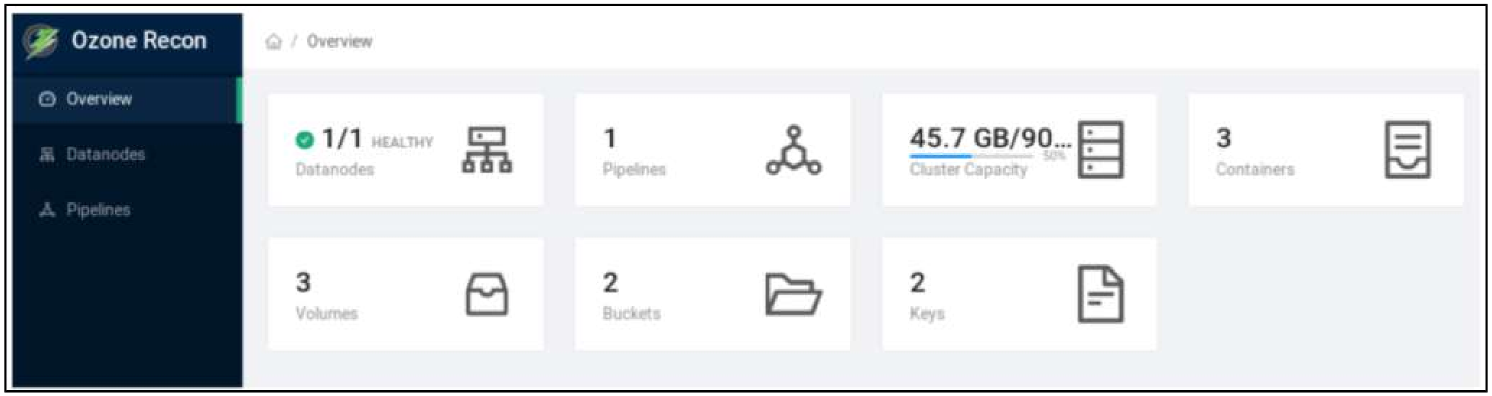
Ozone Manager (OM)

- Mapping service from key names to the corresponding set of blocks
- OM stores its own state in a RocksDB database



Recon Server

- Read-only interface to see a few simple metrics
- Accessible through Recon Web UI link inside Ozone service page of Cloudera Manager



Chapter Topics

Appendix: Ozone Overview

- Ozone Overview
- **Working with Ozone**
- Essential Points

Installation

- Cloudera Manager simplifies Ozone Deployment inside a CDP cluster
- Add Ozone as new service to cluster
- Assign host roles as required
- Restart/Refresh cluster

Upgrade from HDFS to Ozone

- **Plan and estimate block distribution**

```
$ ozone upgrade plan
```

- **Execute the upgrade from HDFS to Ozone**

```
$ ozone upgrade execute
```

Working with Ozone

- **Volumes and Buckets**
- **Command Line tools**
- **Spark integration**
- **Hive/Impala integration**
- **S3 gateway**

Volumes

- Only administrator can create, modify, delete containers
- Usually each user and application gets assigned a volume
- Volumes can have quotas
- List all volumes

```
$ ozone sh volume list
```

- Create volume

```
$ ozone sh volume create --user=training training
```

- Delete non-empty volume

```
$ ozone sh volume delete training
```

Buckets

- Users can create one or more buckets in their volumes
- List all buckets

```
$ ozone sh bucket list training
```

- Create bucket

```
$ ozone sh bucket create training/mybucket
```

- Delete non-empty bucket

```
$ ozone sh bucket delete training/bucket
```

Objects (1)

- **Key/Value pairs can be stored into buckets**
  - Key: Folder and Filename
  - Value: File content
- **Objects are immutable**
- **List bucket content**

```
$ ozone fs -ls o3fs://mybucket.training/
```

- **Create folder**

```
$ ozone fs -mkdir o3fs://mybucket.training/myfolder
```

Objects (2)

- **Upload file**

```
$ ozone fs -put myfile.txt o3fs://mybucket.training/myfolder
```

- **Delete file**

```
$ ozone fs -rm o3fs://mybucket.training/myfolder/myfile.txt
```

S3 Buckets (1)

- Hadoop Ozone comes with an S3 gateway to provide AWS S3 compatibility
- When creating an S3 bucket through the gateway, Ozone dynamically generates a unique volume
- List S3 compatible buckets

```
$ aws s3api --endpoint http://localhost:9878/ list-buckets
```

- Create S3 compatible buckets

```
$ aws s3api --endpoint http://localhost:9878/ create-bucket
 --bucket=mys3bucket
```

S3 Buckets (2)

- **Obtain Ozone volume and bucket name**

```
$ ozone s3 path mys3bucket
```

- **Delete S3 compatible bucket**

```
$ aws s3api --endpoint http://localhost:9878/ delete-bucket
 --bucket=mys3bucket
```

Spark Integration

- **Spark can use Ozone objects transparently like any other Hadoop storage engine**

```
$ pyspark

...

In [1]: spark.read.text("o3fs://mybucket.training/myfolder/
myfile.txt".count()

Out [1]: 3
```

Hive/Impala Integration

- **Hive and Impala can use Ozone objects transparently like any other Hadoop storage engine**

```
$ impala-shell # Or beeline

> CREATE EXTERNAL TABLE … LOCATION “o3fs://mybucket.\
myvolume/myfolder/”;
```

Chapter Topics

Appendix: Ozone Overview

- Ozone Overview
- Working with Ozone
- **Essential Points**

Essential Points

- **Hadoop Ozone is a next-generation storage engine**
- **It implements a distributed and redundant object store**
- **Transparent integrated into the Hadoop ecosystem**
- **Simple architecture**
- **Easy to administer**