Cloudera Runtime

Cloudera Data Platform Security

Date published: 2022-07-01
Date modified: 2022-07-01

https://docs.cloudera.com/

Legal Notice

© Cloudera Inc. 2024. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

Executive summary

This document provides a detailed security reference architecture for Cloudera Private Cloud Base and is accompanied by a series of blog posts. The architecture reflects the four pillars of security engineering best practice, Perimeter, Data, Access and Visibility.

The release of CDP Private Cloud Base has seen a number of significant enhancements to the security architecture including:

- Apache Ranger for security policy management
- Updated Ranger Key Management service

This document is intended as a reference guide only and is not a runbook or configuration manual. Cloudera recommends that you undertake your own security design activity in conjunction with Cloudera Professional Services, and where appropriate conduct third party assurances.

Note also that effective security is not just a combination of physical controls and system architecture but also organizational best practices around password/credential management, joiners-movers-leavers processes, group and privilege management, audit and compliance, separation of duties and physical access controls.

Before diving into the technologies it is worth becoming familiar with the key security principle of a layered approach that facilitates defense in depth. Each layer is defined as follows:

These multiple layers of security are applied in order to ensure the confidentiality, integrity and availability of data to meet the most robust of regulatory requirements. CDP Private Cloud Base offers 3 levels of security that implement these features

Level	Security	Characteristics
0	Non-secure	No security configured. Non-secure clusters should never be used in production environments because they are vulnerable to any and all attacks and exploits.
1	Minimal	Configured for authentication, authorization, and auditing. Authentication is first configured to ensure that users and services can access the cluster only after proving their identities. Next, authorization mechanisms are applied to assign privileges to users and user groups. Auditing procedures keep track of who accesses the cluster (and how).
2	More	Sensitive data is encrypted. Key management systems handle encryption keys. Auditing has been setup for data in the metastore. System metadata is reviewed and updated regularly. Ideally, the cluster has been setup so that lineage for any data object can be traced (data governance).
3	Most	The secure cluster is one in which all data, both data-at-rest and data-in-transit, is encrypted and the key management system is fault-tolerant. Auditing mechanisms comply with industry, government, and regulatory standards (PCI, HIPAA, NIST, for example), and extend from the Cluster to the other systems that integrate with it. Cluster administrators are well-trained, security procedures have been certified by an expert, and the cluster can pass technical review.

For the purposes of this document we are going to focus on the most secure level 3 security.

Architecture

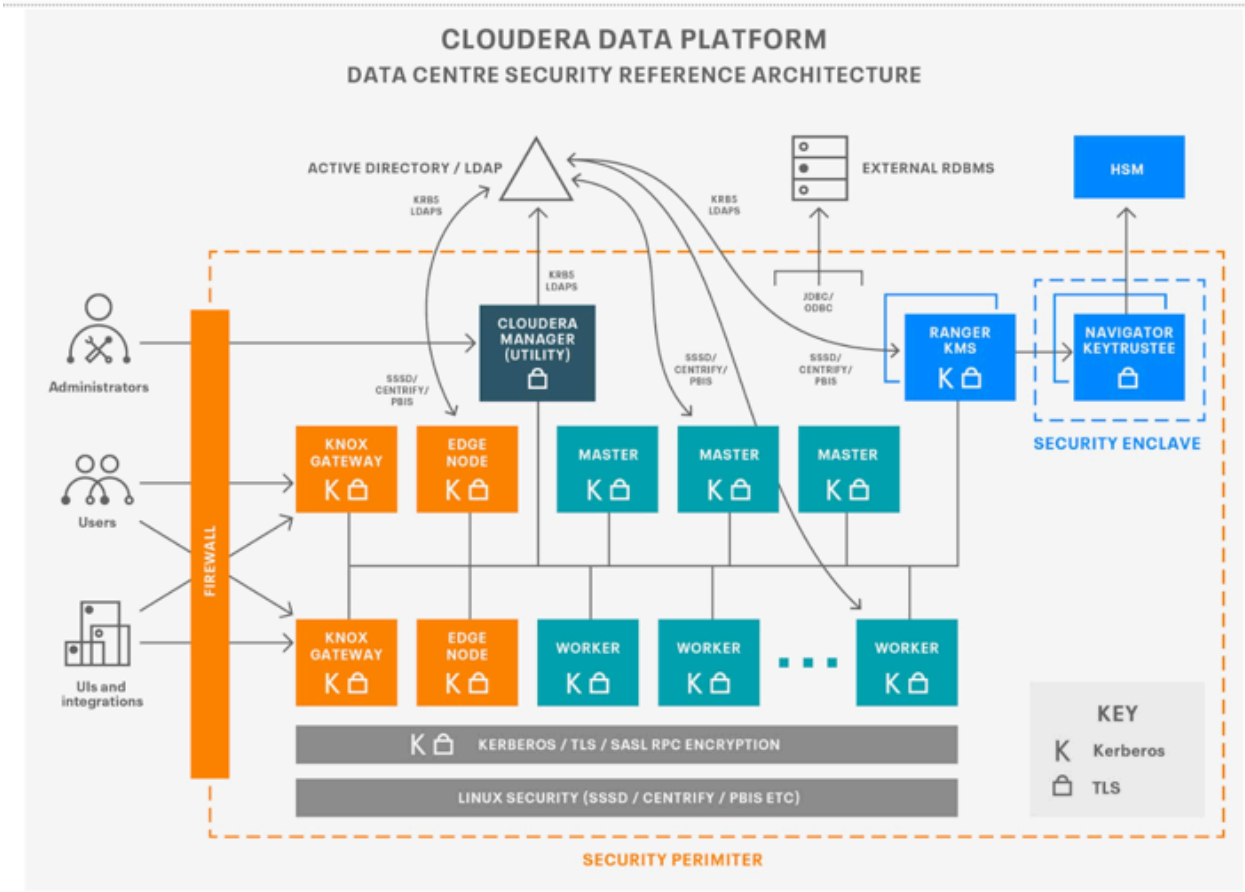
Explanation and diagram of the layers that are applied in order to ensure the confidentiality, integrity and availability of data to meet the most robust of regulatory requirements.

Security architecture improvements

To comply with the regulatory standards of a level 3 implementation, customers will create network topologies that ensure only privileged administrators are able to access core CDP services with applications, analysts and developers limited to appropriate gateway services such as Hue and appropriate management and monitoring web interfaces. The addition of Apache Knox significantly simplifies the provisioning of secure access with users benefiting from robust single sign on. Apache Ranger consolidates security policy management with tag based access controls, robust auditing and integration with existing corporate directories.

Logical Architecture

The cluster architecture can be split across a number of zones as illustrated in the following diagram:



Outside the perimeter are source data and applications, the gateway zones are where administrators and applications will interact with the core cluster zones where the work is performed. These are then supported by the data tier where configuration and key material is maintained. Services in each zone use a combination of kerberos and transport layer security (TLS) to authenticate connections and APIs calls between the respective host roles, this allows authorization policies to be enforced and audit events to be captured. Cloudera Manager will generate credentials either directly against a local KDC or via an intermediary in the corporate directory. Similarly, Cloudera Manager Auto-TLS enables per host certificates to be generated and signed by established certificate authorities. If necessary, authority can be delegated to Cloudera Manager to sign the certificates in order to simplify the implementation. The following sections go into more detail as to how each aspect can be implemented.

Authentication

Overview of authentication in CDP Private Cloud.

Typically a cluster will be integrated with an existing corporate directory, simplifying credentials management and align with well established HR procedures for managing and maintaining both user and service accounts. Kerberos is used to authenticate all service accounts within the cluster with credentials generated in the corporate directory (IDM/AD) and distributed by Cloudera Manager. To ensure that these procedures are secured it's important that all interactions between CM, the Corporate Directory and the cluster hosts are encrypted using TLS security. Signed Certificates are distributed to each cluster host enabling service roles to mutually authenticate. This includes the Cloudera Agent process which will perform an TLS handshake with the Cloudera Manager server in order that configuration changes such as the generation and distribution of Kerberos credentials are undertaken across an encrypted channel. In addition to the CM agent, all the cluster service roles such as Impala Daemons, HDFS worker roles and management roles typically use TLS.

Kerberos

With Kerberos enabled, all cluster roles are able to authenticate each other providing they have a valid kerberos ticket. The authentication tickets are issued by the KDC, typically a local Active Directory Domain Controller, FreeIPA, or MIT Kerberos server with a trust established with the corporate kerberos infrastructure, upon presentation of valid credentials. Cloudera Manager generates and distributes these credentials to each of the service roles using an elevated privilege that is securely maintained within its database. Typically the administration privilege will enable the creation and deletion of kerberos principles within a specific organisation unit (OU) within the corporate directory (see, "Delegating Administration by Using OU Objects"). Good practice is to first enable TLS security between the Cloudera Manager and agents in order to ensure the Kerberos keytab files are transported over an encrypted connection.

Impersonalisation

Within a CDP system there are two methods of impersonation that are supported. The first is a simple "doAs" system, where certain services are trusted to assert the identity of connecting users. For example, Oozie, Hue and Hive are trusted within the CDP environment to act on behalf of the user who connected to them - this is known as impersonation and is configured by the hadoop.proxyuser.* set of parameters.

This allows, for example, a user to connect to Hue or HiveServer2 (and we trust Hue/HiveServer2 to correctly identify that user) and then for Hue/HiveServer2 to act on that user's behalf.

Note: When configuring third-party integrations or add-ons, it is possible that they will require hadoop.proxyuser.* configurations, so that they can also impersonate users that have connected.

The second method is supported by some implementations of Kerberos, is known as Constrained Delegation. For further details on constrained delegation, see "Accessing Secure Cluster from Web Applications".

Related Information

Delegating Administration by Using OU Objects

Accessing Secure Cluster from Web Applications

Encryption

An overview of encryption in CDP Private Cloud.

Encryption in transit is enabled using TLS security with two modes of deployment: manual or Auto TLS. For manual TLS, customers use their own scripts to generate and deploy their own certificates to the cluster hosts and the locations used are then configured in Cloudera Manager to enable their use by the cluster services.

Typically, security artifacts such as certificates will be stored on the local filesystem /opt/cloudera/security. The cloudera-deploy Ansible automation (see, "Cloudera Labs: Cloudera Deploy") uses this method.

Auto TLS enables Cloudera Manager to act as a certificate authority, standalone or delegated by an existing corporate authority. CM can then generate, sign and then deploy the signed certificates around the cluster along with any associated truststores. Once in place, the certificates are then used to encrypt network traffic between the cluster services. There are three primary communication channels, HDFS Transparent Encryption, Data Transfer and Remote Procedure Calls, and communications with the various user Interfaces and APIs.

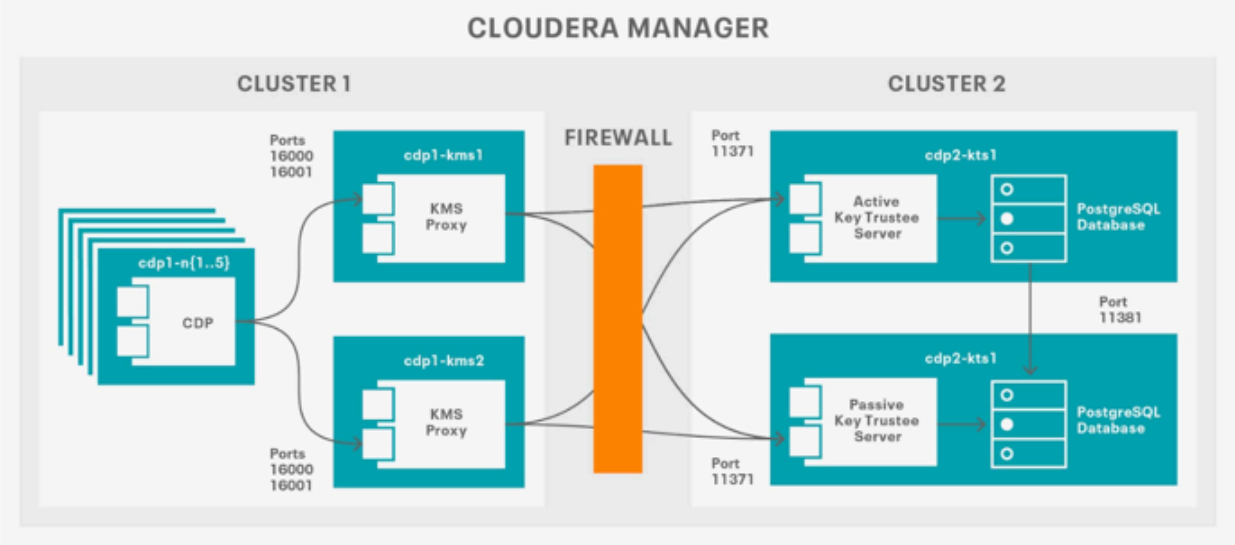
See, “Encryption Overview”.

Encryption at Rest

In order to store sensitive data securely it is vital to ensure that data is encrypted at rest whilst also being available for processing by appropriately privileged users and services that are granted the ability to decrypt. A secure CDP cluster will feature full transparent HDFS Encryption, often with separate encryption zones for the various storage tenants and use cases. We caution against encrypting the entire HDFS, rather those directory hierarchies that require encryption.

As well as HDFS other key local storage locations such as YARN and Impala scratch directories, log files can be similarly encrypted using block encryption. Both HDFS and the local filesystem can be integrated to a secure Key Trustee Service, typically deployed into a separate cluster that assumes responsibility for key management. This ensures separation of duties from the cluster administrators and the security administrators that are responsible for the encryption keys. Furthermore Key Trustee client side encryption provides defence in depth for vulnerabilities such as Apache Log4j2 (see, “CVE-2021-44228”).

Logical Architecture



Ranger KMS Service Overview

The Ranger KMS service consolidates the Ranger KMS service and Key Trustee KMS into a single service that can both support the Ranger KMS DB or the Key Trustee Service as the backing key store for the service. Both implementations support HMS integration with KTS backing support providing full separation of duties and a broader range of HSMs, albeit at the cost of additional complexity and hardware. It is preferable to support RKMS + KTS in order to simplify operations and run at scale. Ranger KMS supports:

- Key Management provides the ability to create, update or delete keys using the Web UI or REST APIs
- Access control provides the ability to manage access control policies within Ranger KMS. The access policies control permissions to generate or manage keys, adding another layer of security for data encrypted in HDFS.
- AuditRanger provides a full audit trace of all actions performed by Ranger KMS.
- All policies are maintained by the Ranger service.

Related Information
Cloudera Labs: Cloudera Deploy
CDP Security Overview
CVE-2021-44228

Key security Services

The Cloudera Security and Governance services form part of the SDX Layer of the cluster that can be deployed and managed with Cloudera Manager. These are described in more detail in the following sections.

Apache Ranger

Apache Ranger is a framework to enable, monitor and manage comprehensive data security across the platform.

Apache Ranger is used for creating and managing policies to access data and other related objects for all services in the CDP stack and features a number of improvements over earlier versions:

- Prior to CDP, Ranger only supported users and groups in policies. In CDP, Ranger has also added the 'role' functionality that existed in Apache Sentry previously. A role is a collection of rules for accessing a given object. Ranger gives you the option to assign these roles to specific groups. Ranger policies can then be set for roles or directly on groups or individual users, and then enforced consistently across all the CDP services.
- One of the key changes introduced in CDP for former CDH users is the replacement of Apache Sentry with Apache Ranger which offers a richer range of plugins for YARN, Kafka, Hive, HDFS and Solr services.HDP customers will benefit from the new Apache Ranger RMS feature which synchronises Hive table-level authorisations with the HDFS file system previously available in Apache Sentry (see, “An Introduction to Ranger RMS”).

More broadly Apache Ranger provides:

- Centralized Administration interface and API
- Standardized authentication method across all components
- Supports multiple authorization methods such as Role based access control and attribute based access control
- Centralized auditing of admin and audit actions

Apache Ranger features a number of components:

- Administration portal UI and API
- Ranger Plugins, lightweight Java plugins for each component designed to pull in policies from the central admin service and stored locally. Each user request evaluated against the policy capturing the request and shipping via a separate audit event to the Ranger audit server.
- User Group Sync, synchronization of users and group memberships from UNIX and LDAP and stored by the portal for policy definition

Customers will typically deploy SSSD or similar such technology in order to resolve at the OS a user's group memberships. The Ranger Audit server will then index those events using the cluster’s Solr Infrastructure service to facilitate analysis and reporting.

Note: Note: Take care with the Hive privilege synchronizer. This runs within Hive and will periodically check permissions every 5 minutes for every Hive object which leads to a larger Hive memory requirements and poor Hive metastore performance where there are many thousands of Hive Objects. The feature is only required where customers are using Beeline or using SQL-based statements to manipulate and review grants. Good practice would be to at least slow down the rate of synchronization hive.privilege.synchronizer.interval=3600 (Default is 720) to check every hour or disable completely and use the Ranger UI to manage policies.

Security Zones

Security zones enable you to arrange your Ranger resource and tag based policies into specific groups in order that administration can be delegated. For example in a specific security zone:

- Security zone "finance" includes all content in a "finance" Hive database.
- Users and groups can be designated as administrators in the security zone.
- Users are allowed to set up policies only in security zones in which they are administrators.
- Policies defined in a security zone are applicable only for resources of that zone.
- A zone can be extended to include resources from multiple services such as HDFS, Hive, HBase, Kafka, etc., allowing administrators of a zone to set up policies for resources owned by their organization across multiple services.

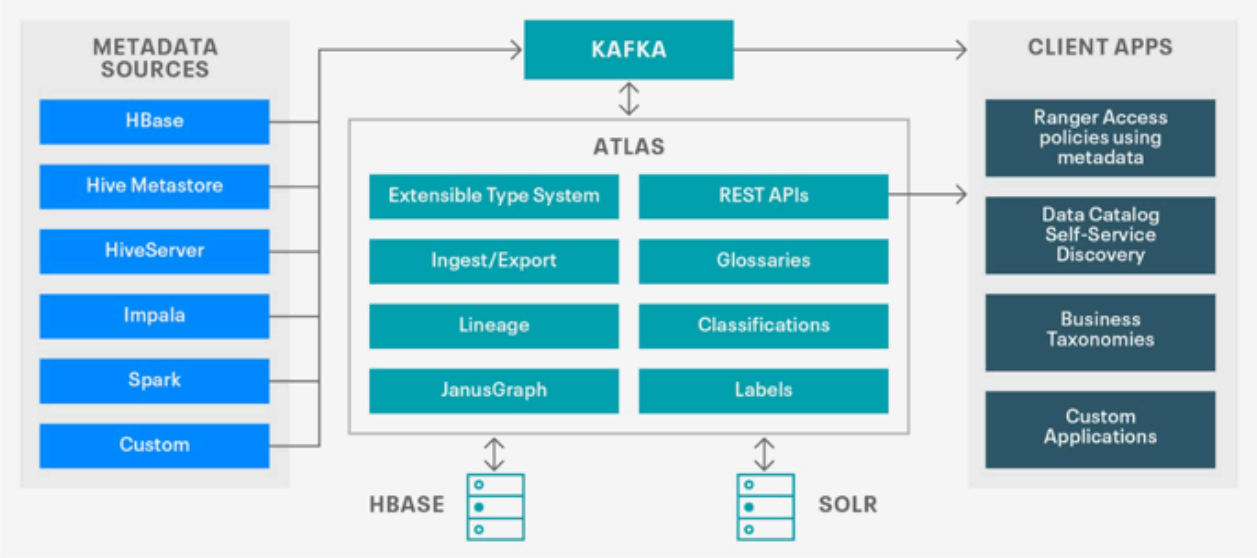
Related Information

An Introduction to Ranger RMS

Apache Atlas

Atlas is a scalable and extensible set of core foundational governance services – enabling enterprises to effectively and efficiently meet their compliance requirements within CDP and allows integration with the whole enterprise data ecosystem.

Organizations can build a catalog of their data assets, classify and govern these assets and provide collaboration capabilities around these data assets for data scientists, analysts and the data governance team. Logically, Apache Atlas is laid out as follows:



Apache Knox

Apache Knox simplifies access to the cluster Interfaces by providing Single Sign-on for CDP Web UIs and APIs by acting as a proxy for all remote access events. Many of these APIs are useful for monitoring and issuing on the fly configuration changes.

As a stateless reverse proxy framework, Knox can be deployed as multiple instances that route requests to CDP's REST APIs. It scales linearly by adding more Knox nodes as the load increases. A load balancer can route requests to multiple Knox instances.

Knox also intercepts REST/HTTP calls and provides authentication, authorization, audit, URL rewriting, web vulnerability removal and other security services through a series of extensible interceptor pipelines.

Each CDP cluster that is protected by Knox has its set of REST APIs represented by a single cluster specific application context path. This allows the Knox Gateway to both protect multiple clusters and present the REST API consumer with a single endpoint for access to all of the services required, across the multiple clusters.

In CDP certain providers (sso, pam, admin, manager) and topologies (cdp-proxy, cdp-proxy-api) are already pre-configured and are mostly integrated into the configuration UI of Cloudera Manager. Furthermore CDP ships with a useful pre-configured home page for users to navigate to those services.

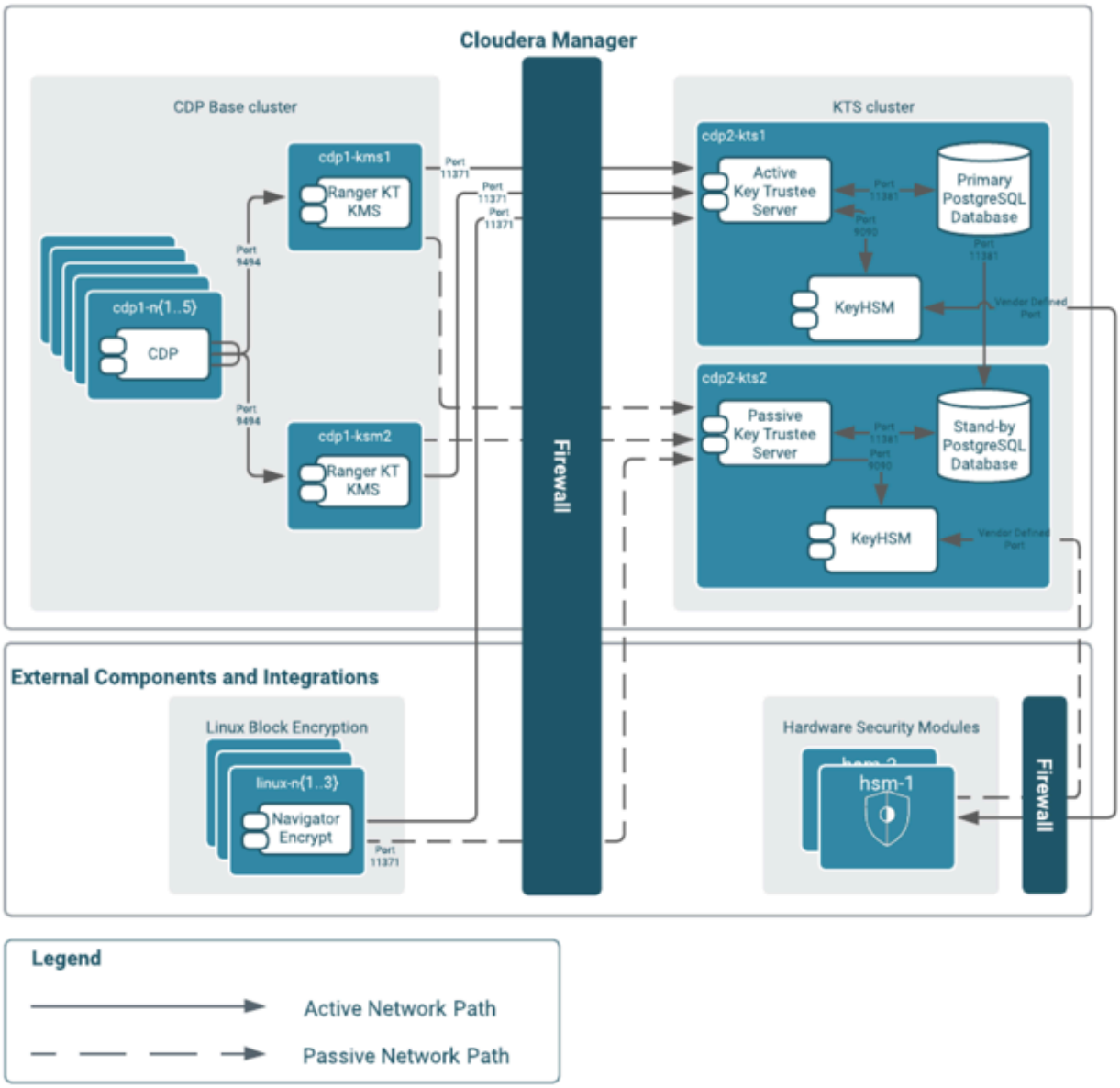
The cluster definition is defined within the topology deployment descriptor and provides the Knox Gateway with the layout of the cluster for purposes of routing and translation between user facing URLs and cluster internals.

Simply by writing a topology deployment descriptor to the topologies directory of the Knox installation, a new CDP cluster definition is processed, the policy enforcement providers are configured and the application context path is made available for use by API consumers. CDP Private Cloud Base comes with a preconfigured topology for all of the various cluster services which customers can extend for their environments.

Encryption logical architecture

Integrity of the data is a crucial pillar to meet the requirements of discerning investors that are looking to hold organizations to the highest ethical, social and governance (ESG) standards. Cloudera Data Platform delivers robust encryption at rest, using KeyTrustee Server to manage key material with the option of integrating with an HSM (Hardware Security Module) to act as your Key Backing store.

This is advantageous as it facilitates integration with pre-existing security processes for maintaining cryptographic material and separation of duties with the security department responsible for maintaining the HSM. In the context of our climate accounting application this ensures that the organization's existing regulatory framework is applied and the potential to achieve equivalent standards or reporting as financing. The diagram below illustrates how this is implemented logically.



The architecture supports the encryption of both HDFS and the local filesystem. Local filesystem directories might include Impala and Yarn scratch directories which are used as working directories when a job might spill to disk due to the task exceeding the memory capacity of worker node, audit directories containing user IDs, and in some cases log directories that contain clauses with PII data depending upon existing enterprise security policies.

We generally recommend that customers avoid encrypting the entire HDFS namespace as this will lead to operational challenges and create encryption zones within the hdfs directory structure. Customers will often create separate cryptographic zones for cluster tenants, however there will often be cases where multiple tenants may wish to share the same datasets and it is more practical to create a single cryptographic zone and utilize Apache Ranger authorisation policies to provide the fine grained access controls. Consideration should be given more broadly to the underlying regulatory environment in arriving at a zoning policy. So for example all data that is classified as Commercially Sensitive might be encrypted under a single zone. Data classified as Secret in another. An illustration of this might be any additional columns that contain commercially sensitive financial information.

Encryption at rest installation

For full installation details using the Cloudera Manager wizard, see “Setting Up Data at Rest Encryption for HDFS”.

Identity Mapping

Much of this blog post focuses upon authorizing groups of users to perform administrative and analytical tasks against different datasets. In order to provide effective security the cluster must first authenticate both users and applications typically using kerberos or in some cases LDAP or SAML. Once an identity has been established that user in question will then be authorized in order to perform some action. In order to do this, the cluster will use and identify mapping service such as SSSD, Centrify or Powerbroker which will integrate with the corporate directory such as Microsoft Active Directory or Red Identity Manager which provides established tools and processes for mapping users to groups which are then used for authorisations.

A new feature introduced in CDP are Ranger roles, previously Ranger only had the concept of groups. Administrators can create roles and then allocate permissions to those roles to perform an action such as to read a table in a database. These roles can then be linked to groups that are maintained in the corporate directory. The advantage is that typically these groups pre-exist and can be easily linked to the new ranger roles which in turn can be amended whilst the linked group remains unchanged. The role itself is a collection of policies whereas the group is the collection of users.

Related Information

Setting Up Data at Rest Encryption for HDFS

Network design

In most situations it will be appropriate to place the CDP cluster within a VLAN of its own and whitelist a number of ports that will be used for inbound connections. This allows for a simplified network configuration for intra-cluster communications, whilst protecting the cluster from external access.

East West traffic on a typical cluster can be extremely high, therefore it's important that a leaf spine network design be used without inferring firewalls between cluster nodes and racks in order not to compromise application performance.

In theory, this level of control might not be necessary given that all ports are protected by one of the authentication mechanisms specified in Kerberos and Identity, however a typical deployment would involve a VLAN.

Isolating Clusters within the same VLAN

Note:

Note: TSB 2021-544: Microsoft AD November 2021 Security Update

Following the November 2021 Security Update for Active Directory (AD) from Microsoft, customers will experience issues with re-generating kerberos credentials due to a change in the way service principal name (SPN) alias uniqueness is determined. This will impact customer upgrades to CDP. For the latest update on this issue, see the corresponding Knowledge article, "Cloudera Customer Advisory-544: Microsoft AD November 2021 Security Update".

Due consideration should be given to how to isolate two clusters that have network access to each other and are housed in the same Kerberos domain. By default, Hadoop does not differentiate between clusters who present the same prefix of a three part Kerberos principal. For example, consider the following service principal names:

- hdfs/myprodhost1.example.com@EXAMPLEREALM.COM
- hdfs/mydevhost1.example.com@EXAMPLEREALM.COM

Using an out of the box configuration, the cluster will extract the first part of the SPN (before the / character) and use that as the service shortname, meaning that there is no way of differentiating between the hdfs user on a production cluster and that on a development cluster.

In order to counteract this, Cloudera recommends using Auth-to-Local rules to isolate any clusters that have network connectivity and are in the same realm, or are trusted by the cluster. For further information on how to configure Auth-to-Local Rules

Note: Using Custom Kerberos Principals (see, “Customizing Kerberos principals”) is generally not recommended, as support is limited due to environment specific testing required.

Edge and Gateway Roles

In a standard configuration it is recommended that Apache Knox be used as a Gateway and Load Balancer for all interactions into the cluster, with the exception of Cloudera Manager and SSH access. SSH access should be restricted to administrators only, other than Edge Nodes, to which end users may be permitted non-root access.

Related Information

Cloudera Customer Advisory-544: Microsoft AD November 2021 Security Update

Customizing Kerberos principals

Basic hive setup

In our scenario there are multiple tenants on a shared cluster. It is assumed that there are a number of encryption zones within the hdfs namespace supporting a variety of user cases.

This will include Travel data with policy restrictions according to EU GDPR regulation requiring masking and encryption of policy data as well as internal public cloud usage data which summarizes energy consumption and can be reported as part of the sustainability initiative. Elements of the sustainability data are considered as corporately sensitive, therefore access is limited to certain tables.

Encryption at rest

The company has a policy of having all Commercially Sensitive data, encrypted at rest, and therefore according to the data classification, the Climate Accounting data will be stored in the Finance encryption zone on the cluster. For the purposes of regulatory reporting the provenance of the data must be assured to the highest corporate standards. Therefore, encryption at rest is mandatory due to regulatory requirements.

Key security documentation

- “How-to: Security”
- “Encryption Zones and Keys”

Commands

In our example we use finance as the <keyname> and create a specific encryption zone for the needs of the finance department on our shared corporate clusters on hdfs at /warehouse/finance/:

1. Create zone /warehouse/finance/
   a. Create an encryption key for your zone as keyadmin for the user/group (regardless of the application that will be using the encryption zone):

```
sudo -u hbase hadoop key create <key_name>
```

2. Create a new empty directory and make it an encryption zone using the key created above:

```
sudo -u hdfs hadoop fs -mkdir /warehouse/finance/
sudo -u hdfs hdfs crypto -createZone -keyName finance -path /warehouse/fin
ance/
```

3. You can verify creation of the new encryption zone by running the -listZones command. You should see the encryption zone along with its key listed as follows:

```
sudo -u hdfs hdfs crypto -listZones
```

```
/warehouse/finance/ finance
```

In this case we create a general encryption zone for all financially sensitive information that will be stored on the cluster. Other encryption zones exist for other data classifications.

Related Information

How-to: Security

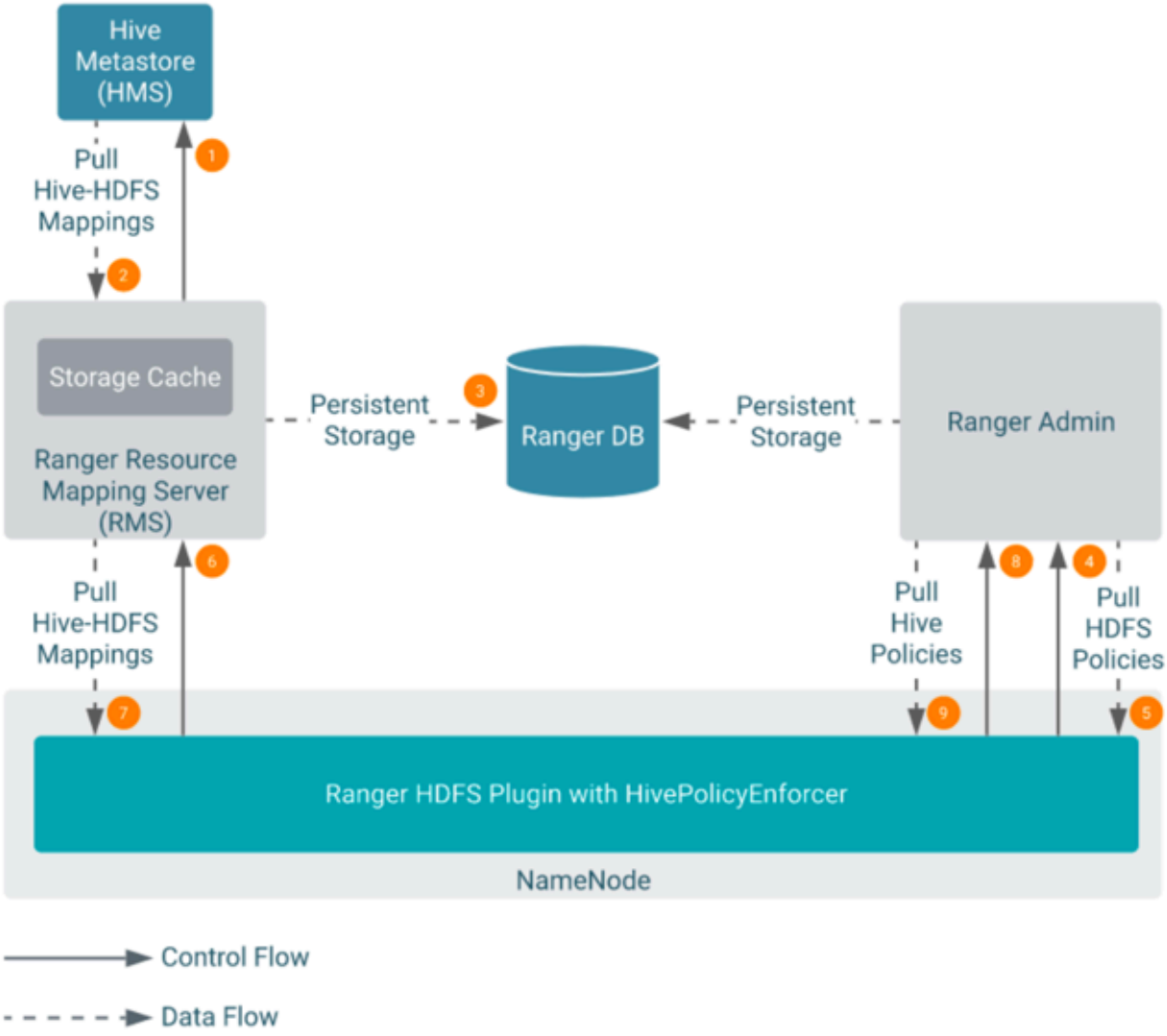
Encryption Zones and Keys

Cluster storage and Ranger mapping service

Ranger Mapping service adds a feature that was available to legacy CDH users that automatically maps Hive Object policies to the underlying storage ACLs, this allows you to authorize access to HDFS directories and files using policies defined for Hive tables, thus simplifying protective access to the underlying data.

Limitations

See, “Ranger Hive-HDFS Policy Sync Overview”.



Related Information

Ranger Hive-HDFS Policy Sync Overview

Create database

In this example we will use a HIVE ACID table as we expect our table to feature a number of slowly changing dimensions such as mileage and data center utility rates. Furthermore we don't expect significant transaction volumes or no of Hive Objects such as partitions on our clusters that could impact performance due to compactions.

We will create a specific database that will use managed tables for both travel expenses and public cloud reporting (see, "Create a default directory for managed tables"). From a security perspective some of the tables will have read access for all employees, with write access limited. The trip table will feature Personal Identifiable Information, and employees in the European Union will be subject to GDPR regulatory requirements and therefore those columns will need to be masked to a restricted number of users that require access.

In the following section we illustrate how we can apply a GDPR compliance policy using Apache Ranger on these objects, first we create the tables.

```
CREATE DATABASE carbonAccounting
```

Each table will be stored in the default managed location for finance data /warehouse/finance/managed/hive.

Our database will be composed of a number of tables that capture scope 1 emissions data such as travel and data center power consumption. We also include an Embedded Emissions Table which can be used to store carbon contribution of assets such as physical hardware.

```
CREATE TABLE trip(employeeID string, date date, departureCity string, arriva
lCity string, routeDistance int, fligh No string, serviceClass string, journe
yType string, co2Emission int)
  STORED AS ORC
  TBLPROPERTIES ('transactional'='true',
  'transactional_properties'='insert_only');

CREATE TABLE officeEmissions(use https://docs.google.com/spreadsheets/d/1
NYU0eQDukjYufN80qzrrSjf2ZUE672wop69qCb3G0Sc/edit?usp=sharing for columns)
  STORED AS ORC
  TBLPROPERTIES ('transactional'='true',
  'transactional_properties'='insert_only');

CREATE TABLE public_cloud_reporting(Kilowatt int, DataCentre string, Utili
zation int)
  STORED AS ORC
  TBLPROPERTIES ('transactional'='true',
  'transactional_properties'='insert_only')
Description "captures power draw of respective data center note public clo
ud";

CREATE TABLE carbonEmissions(type int, emission int, value int)
  STORED AS ORC
  TBLPROPERTIES ('transactional'='true',
  'transactional_properties'='insert_only');
```

The emission data can be calculated by means of a function that can call reference data that might be delivered to the cluster using "Cloudera Flow Management tools" (CFM) such as Apache NiFi. Typically this reference data will be subject to commercial agreements and therefore access will need to be restricted.

Related Information

Creating a default directory for managed tables

Cloudera Flow Management tools

Create policies

Create basic permissions for the DB with two roles: Full Access (Finance) and Read Only/Limited Access (Cloudera Staff) with Ranger Policies.

Best practise is to apply policies to roles, for example:

- Create Role ClimateAuditor (User role for data) ClimateAdmin (Admin Role)
- Admin roles have full control of the ClimateDatabases, can create new tables, add columns etc
- User roles have read write access to data
- Roles can be updated with additional policies and linked to both pre-existing and new groups.

Typically Administrators will link these roles to pre-existing Groups that exist in the corporate directory to enable existing HR processes to maintain memberships, for example a “Finance” group.

Create New Groups and link ClimateAccounting (e.g Composed of individual members such as Architects and Engineers), new groups may be project specific.

Ranger RMS function ensures that as well as the Hive Object, the underlying files are similarly protected with equivalent file ACLs.

Sync ACLs summarize Ranger RMS features and how to configure.

“Migrating from Sentry to Ranger”>“Ranger policies allowing create privilege for Hadoop_SQL databases”

Data encrichment

Whilst travel data will be extracted directly from the finance and accounting system, the enrichment process will see that data blended with external sources such as aircraft carbon emission data, in order to arrive at a more accurate true emissions analytic. Apache NiFi is an excellent technology for these types of enrichment pipelines enabling them to be loosely coupled whilst retaining regulatory compliance.

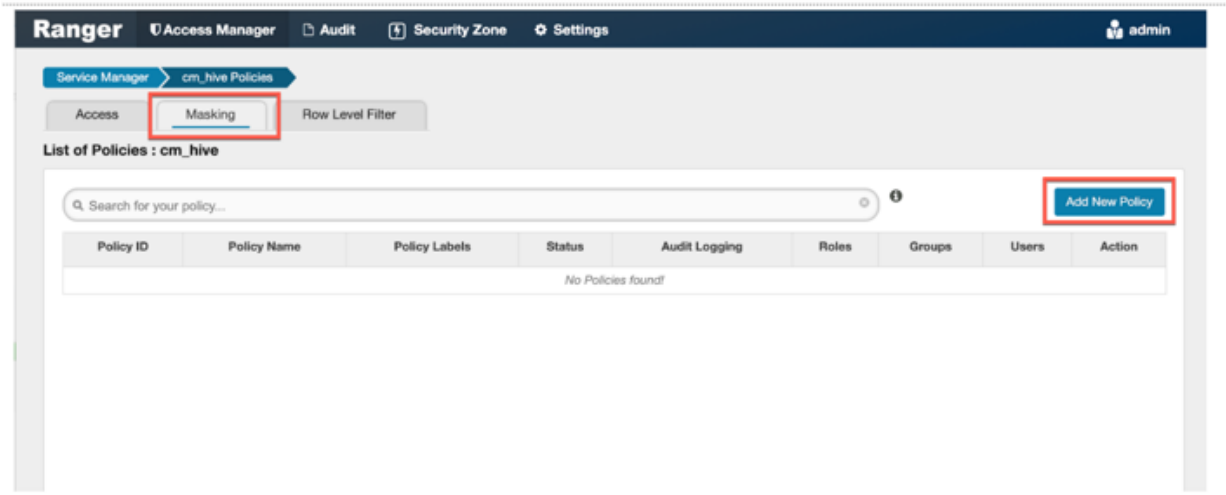
Data masking

Ingest from a governed data source requiring masking of certain columns, for example an employee's ID which would likely be incorporated into any journey information. (Note we capture the EmployeeID as the organization plans to evaluate a policy of budgeting CO2 emissions for individuals as part of its ongoing sustainability efforts.)

Read and analytical access via Impala (JDBC/Tableau etc) illustrate permissions and connection strings.

To apply a mask we need to create a Ranger policy as follows:

1. Navigate to Apache Ranger Access Manager > Service Manager and select Add New Policy as follows:



2. We then create the policy adding the Name (GDPR), Hive Database (carbonAccounting), Table (trip), Hive column (employID), Audit Logging (yes), validity.
3. We then apply to the roles in this case Finance and GDPR.
4. Select Masking Type, in this case, Redact.

Related Information

Ranger policies allowing create privilege for Hadoop_SQL databases

Summary

In this reference architecture we have illustrated how to create a secure CDP Private Cloud Base cluster that is fully encrypted with the principle of least privileges applied to authorized users that are able to perform tasks associated with their allocated roles.

Combined with perimeter security and auditing, administrators will be able use these principles in order to comply with the most stringent of regulatory requirements.

For further information have a look at the CDP Security documentation, see “Cloudera Runtime Security and Governance”.

Related Information

CDP Security Overview