DEBO CLI

User Manual

Version 0.1.0

deboteamwork@gmail.com

Contents

Introduction

Debo is a lightweight, open-source Hadoop cluster management tool offering both CLI and GUI control. This manual covers the CLI interface, tailored for administrators and users familiar with terminals, scripting, and automation.

Component Support

Debo is a comprehensive tool built to manage a wide range of components within the Hadoop ecosystem and related big data infrastructure. Each supported component can be managed individually through a consistent interface that allows users to perform standard lifecycle operations such as installation, configuration, and monitoring.

The following components are currently supported by Debo:

Big Data & Distributed Processing

- HDFS – Hadoop Distributed File System
- YARN – Yet Another Resource Negotiator
- MapReduce (via YARN and Tez) – Handled through component interactions

Data Processing Engines

- Spark
- Flink
- Tez
- Pig
- Presto
- Livy

Databases & Storage Engines

- HBase
- Phoenix
- Hive

Messaging & Coordination

- Kafka
- Zookeeper

Security & Governance

- Ranger
- Atlas

Search & Visualization

- Solr
- Zeppelin

Each of these components can be controlled using Debo's unified set of actions:

Supported Actions	Description
INSTALL	Installs the selected component
UNINSTALL	Removes the selected component
START	Starts the service
STOP	Stops the service
RESTART	Restarts the service
CONFIGURE	Applies or updates configuration for the component
VERSION_SWITCH	Switches to a different supported version of the component
REPORT	Displays the status or operational report of the component

By supporting both core infrastructure (e.g., HDFS, Kafka, Zookeeper) and upper-layer services (e.g., Hive, Ranger, Zeppelin), Debo provides full-stack control for data platform operations. Components can be managed independently or in combination with their dependencies (using the --with-dependencies flag), giving users flexibility and control over complex deployments.

Prerequisites

The `debo` CLI tool requires a GNU/Linux environment with the following components:

1. **Build Tools**: GCC compiler and GNU Make must be installed to compile from source.
2. **Core Libraries**:
   - `libxml2` (for XML configuration parsing)
   - `libgssapi_krb5` (for Kerberos authentication)
   - Standard C libraries (`libresolv`, `pthreads`)
3. **Installation**:
   - Root/sudo access is required for system-wide installation to `/usr/local/bin` (customizable via `PREFIX`/`DESTDIR` in the Makefile)
   - Kerberos environment must be configured with `KRB5_KTNAME` set to your keytab path (e.g., `/etc/debo.keytab`)

Install dependencies using your system package manager:

Debian/Ubuntu:

```
bash
sudo apt install build-essential libssh-dev libxml2-dev libkrb5-dev
```

RedHat/CentOS:

```
bash
sudo yum install gcc make libssh-devel libxml2-devel krb5-devel
```

The provided Makefile will automatically verify dependencies during compilation. Failed checks will display explicit error messages guiding remediation. Runtime execution requires proper Kerberos ticket configuration for authenticated operations.

Security: Kerberos Authentication

Debo uses **Kerberos** authentication—an industry-standard, secure authentication system well-suited for distributed systems communicating over untrusted networks. This mechanism provides strong client-server authentication, preventing impersonation attacks and credential sniffing.

Availability

Debo leverages **Kerberos Version 5** via **GSSAPI**, which must be available on the system. While Kerberos itself is not distributed with Debo, packages such as `krb5-user`, `libkrb5-dev`, and `libgssapi-krb5-2` are typically available through your Linux distribution's package manager (e.g., `apt` on Debian/Ubuntu or `yum`/`dnf` on RHEL/CentOS).

To ensure full compatibility, we recommend using the MIT Kerberos implementation. Some vendor-supplied alternatives may be non-interoperable or incomplete.

Configuration and Installation

Before using Debo, ensure the system is properly configured for Kerberos:

- A valid **Kerberos Key Distribution Center (KDC)** must be running and accessible.
- Create a **Kerberos principal** for the Debo service, e.g.:

  ```
  addprinc -randkey debo/hostname@YOUR.REALM
  ```

- Export the keytab file for the server to a secure path:

  ```
  ktadd -k /etc/debo.keytab debo/hostname@YOUR.REALM
  ```

- Make sure the keytab file is **readable by the user running Debo**, and set the environment variable:

  ```
  export KRB5_KTNAME=/etc/debo.keytab
  ```

- Set the **Kerberos service name** used by Debo to match the server principal:

  ```
  export KRB5_SRVNAME=debo
  ```

The default service name is debo .

Operation

Once configured, Debo will authenticate clients using the **GSSAPI** mechanism. This applies to remote commands issued between the Debo server and Debo agents. The client must obtain a valid **Kerberos ticket** before initiating operations:

```
kinit yourusername@YOUR.REALM
```

Debo then transparently uses GSSAPI to authenticate with the server.

The client principal is expected to match the system user running the Debo command, and the service principal must follow the form:

```
debo/hostname@REALM
```

To verify or debug authentication issues, environment variables like KRB5_TRACE=/dev/stdout can be useful.

Dependencies

Debo's build system automatically checks for the presence of required libraries:

- libgssapi_krb5 for GSSAPI-based Kerberos support
- libkrb5-dev for development headers and linking
- Ensure the pkg-config utility can locate these libraries

If GSSAPI support is not found, the build process will fail with an informative error message.

Version supported

Debo is designed to support a wide range of versions across various components of the Hadoop ecosystem, enabling users to manage and deploy compatible services with confidence. The tool currently supports multiple versions of core technologies, including **Hadoop** (2.10.2 to 3.4.1), **HBase** (2.4.18 to 2.6.2), **Hive** (4.0.1), **Flink** (1.17.2 to 1.20.1), **Tez** (0.9.2 to 0.10.4), **Ranger** (0.6.3 to 2.6.0), **Phoenix** (4.16.1 to 5.2.1), **Kafka** (3.7.2 to 3.9.0), and **Zookeeper** (3.7.2 to 3.9.3). It also includes support for **Spark** (3.4.4, 3.5.5), **Pig** (0.16.0, 0.17.0), **Livy** (0.7.1), **Solr** (9.8.1), and **Zeppelin** (0.8.2 to 0.12.0). While some tools like **Presto** currently do not have predefined versions, Debo's modular structure allows for future expansion. Version awareness ensures that configuration, installation, and compatibility checks are tailored precisely to the selected software stack.

Dependency Management

Debo features an intelligent but **user-controlled dependency management system**, allowing users to decide whether actions on a component should also apply to its dependent services. Each component in the Hadoop ecosystem managed by Debo may have one or more dependencies—services that must be installed, started, or configured before the target component can function properly. However, Debo **does not apply actions to dependencies automatically** unless explicitly instructed by the user.

To enable this behavior, users must specify the `--with-dependencies` option (or its shorthand, if available) when issuing actions such as `INSTALL`, `START`, `STOP`, `RESTART`, or `CONFIGURE`. When this option is used, Debo resolves the dependency tree of the target component and applies the action to all its required dependencies in the correct order.

For example:

- Running `debo install HBASE --with-dependencies` ensures that **HDFS** and **Zookeeper** are installed before HBase.
- `debo start HIVE --with-dependencies` will start **HDFS** and **Tez** (and implicitly YARN), followed by Hive.
- If `--with-dependencies` is omitted, only the specified component will be affected.

This explicit control gives users flexibility—enabling them to perform isolated operations when needed, or manage full stacks when appropriate.

The following are examples of dependency relationships:

Component	Dependencies
HBASE	HDFS, Zookeeper
HIVE	HDFS, Tez
PHOENIX	HBase
STORM	Zookeeper
SPARK	HDFS
TEZ	HDFS
LIVY	Spark

Component	Dependencies
RANGER	Solr, HDFS
ATLAS	HBase, Kafka, Solr
PIG	HDFS
SOLR	Zookeeper
YARN	HDFS
FLINK	HDFS, YARN

Components like **HDFS**, **Zookeeper**, **Presto**, and **Zeppelin** have no declared dependencies and can be managed independently.

This design ensures that Debo operations remain predictable and transparent, empowering users to control both component behavior and the extent of dependency involvement.

Version Switching

Debo supports switching component versions via the VERSION_SWITCH action, allowing administrators to move between any two explicitly supported versions listed for a component — in **any direction** (upgrade or downgrade).

How It Works

- A version switch is **allowed only if**:
  - The current version is listed in the component's version array.
  - The target version is also listed in the same array.
- Version order does **not** matter: you can switch from a newer to an older version or vice versa, as long as both are listed.

Command Syntax

debo version-switch <COMPONENT> --to <TARGET_VERSION>

For example, if a component like HBase is currently running version 2.5.11, and you wish to downgrade to 2.4.18, the following is valid:

debo version-switch HBASE --to 2.4.18

This is permitted because both 2.5.11 and 2.4.18 are in the supported version list.

Supported Versions Per Component

Below is a list of supported versions per component. Any version listed can be switched **to/from** any other listed version:

Component	Supported Versions
Hadoop	2.10.2, 3.2.4, 3.3.5, 3.3.6, 3.4.0, 3.4.1
Presto	None (version switching not available)
Pig	0.16.0, 0.17.0
HBase	2.4.18, 2.5.11, 2.6.1, 2.6.2
Hive	4.0.1

Component	Supported Versions
Flink	1.17.2, 1.18.1, 1.19.0, 1.19.1, 1.19.2, 1.20.0, 1.20.1
Livy	0.7.1
Tez	0.9.2, 0.10.1, 0.10.2, 0.10.3, 0.10.4
Ranger	0.6.3, 0.7.1, 1.0.0, 1.1.0, 1.2.0, 2.0.0 → 2.6.0
Phoenix	4.16.1, 5.1.2, 5.1.3, 5.2.0, 5.2.1
Solr	9.8.1
Spark	3.4.4, 3.5.5
Zeppelin	0.8.2, 0.9.0, 0.10.0, 0.10.1, 0.11.0, 0.11.1, 0.11.2, 0.12.0
Kafka	3.7.2, 3.8.0, 3.8.1, 3.9.0
Zookeeper	3.7.2, 3.8.4, 3.9.3

📝 **Note**: Some components like Presto currently have no versions defined and therefore do not support switching. As more versions become available in future releases, switching will be enabled accordingly.

Architecture

The **Debo system** is built on a simple yet powerful architecture optimized for efficient control and flexibility. Whether you're managing a single machine or a full Hadoop cluster, Debo provides a lightweight and modular **client–server–agent model**.

Note: This user manual focuses specifically on using Debo via the **Command-Line Interface (CLI)**. While Debo also supports a Graphical User Interface (GUI), that mode of operation is documented separately.

CLI-Based Interaction

In the CLI mode, users interact directly with the **Debo server** by entering commands in the terminal. These commands instruct the server to perform various tasks such as:

- Starting or stopping services
- Installing components
- Checking status
- Collecting resource metrics
- Managing configuration

The server processes each command and either executes it locally (in standalone mode) or communicates with agents running on remote nodes (in a multi-node setup). The results are printed directly in the terminal for immediate visibility and feedback.

Standalone and Distributed Modes

Debo supports both **standalone** and **distributed** configurations:

- In **standalone mode**, the server handles all operations on the local machine—**no agent processes are required**.
- In **multi-node mode**, the server communicates with **Debo agents** running on other machines to coordinate operations across the cluster.

This flexibility allows Debo to be effective in everything from development and testing to full production environments.

Agent-Based Execution

In a distributed setup, each node in the cluster runs a **Debo agent**, implemented as a **process-based service**. These agents are **passive listeners**—they do not initiate communication with the server but instead wait for incoming instructions.

When the Debo server sends a command to an agent:

1. The agent **receives the request**.
2. It immediately **forks a new child process** to handle the task independently.
3. The child process executes the command (e.g., managing a service, gathering metrics).
4. The result is returned to the server.
5. The main agent process continues listening for further requests.

This **forking model** enables agents to:

- **Handle multiple requests in parallel**
- Avoid blocking the main agent process
- Scale well under high load or concurrent tasks

Each command is isolated within its own process for robustness, fault isolation, and reliability.

Efficient Communication Model

Communication in Debo follows a **request-response pattern**:

- **Agents do not send unsolicited messages.**
- **The server initiates all communication.**

This predictable flow helps ensure:

- Reduced network traffic
- Improved security boundaries
- Easier debugging and system monitoring

Summary

To summarize, Debo's CLI architecture offers:

- A powerful terminal-based control interface
- Seamless operation in both **standalone** and **multi-node** modes
- **Process-based agents** that fork per request for concurrency
- A **passive, server-driven communication model**

This structure provides a stable, scalable foundation for managing Hadoop components from the command line, with the flexibility to support more advanced interfaces in the future.

Installing Debo from Source

To get started with **Debo**, you need to clone the source code from the official GitHub repository and build the appropriate components using the provided Makefile.

Clone the Repository

Open your terminal and run the following command:

```
git clone https://github.com/Debo-et/debo-teamwork.git
```

This will download the entire Debo project into a local directory named debo-teamwork.

Understand the Source Structure

The cloned project directory contains **two main subdirectories**:

- server/ — contains the **Debo server** source code.
- agent/ — contains the **Debo agent** source code.

Each directory includes its own Makefile for compiling the respective component.

Install Required Dependencies

Before building either component, make sure the following system libraries and development packages are installed:

- libssh
- libxml2
- libkrb5-dev (for GSSAPI / Kerberos support)

On Debian-based systems, install them using:

```
sudo apt install libssh-dev libxml2-dev libkrb5-dev build-essential
```

! If these dependencies are missing, the Makefile will fail with a clear error message before compilation.

Build the Debo Server

To build the **Debo server**:

```
cd debo-teamwork/server
make
```

This will generate an executable (typically named debo) in the current directory.

If you'd like to install it system-wide (e.g., to /usr/local/bin), run:

```
sudo make install
```

 Developer Tip:
If you're testing or developing, you can skip make install and run the compiled executable **directly from the build directory**, e.g.:

```
./debo --hdfs
```

Build the Debo Agent

To build the **Debo agent**:

```
cd debo-teamwork/agent
```

```
make
```

This creates the deboAgent binary in the current directory.

You can install it system-wide with:

```
sudo make install
```

 Just like the server, developers may choose to **run deboAgent directly** from the build directory without installing it globally.

Clean, Rebuild, or Uninstall

To clean the build artifacts:

```
make clean
```

To uninstall a previously installed binary:

```
sudo make uninstall
```

🔧 Makefile Behavior (Reference)

Both Makefiles:

- Automatically check for required libraries before compiling
- Fail gracefully if dependencies are missing
- Use /usr/local/bin as the default install target (customizable via PREFIX)
- Support development workflows by allowing direct use of the compiled binary without installation

Installation Hadoop components

The install function of **Debo** is responsible for automating the installation and initial configuration of a wide range of Apache big data components. It is designed to simplify the setup process by downloading, installing, and configuring each component with sensible defaults, allowing administrators to get up and running quickly.

Supported Components

Debo can install the following Apache components:

- **Flink**
- **HDFS**
- **YARN**
- **HBase**
- **Hive**
- **Kafka**
- **Livy**
- **Phoenix**
- **Storm**
- **Pig**
- **Presto**
- **Atlas**
- **Ranger**
- **Solr**
- **Spark**
- **Tez**
- **Hive Metastore**
- **Zeppelin**
- **ZooKeeper**

📡 **Note**: Debo requires an active internet connection during installation to download necessary binaries and dependencies.

⚠ **Important:** One known issue is the instability of Apache's official binary distribution URL (`https://dlcdn.apache.org`). Occasionally, this URL changes or certain component paths are moved or removed, which may cause installation commands to fail. If you encounter such issues, **please try using the latest commit** of Debo, as the system actively tracks these changes and updates the download URLs accordingly to maintain compatibility.

Default Configuration Handling

Once a component is installed, Debo automatically applies its **default configuration** and stores it in a reference text file. This ensures you can review or restore initial settings in the future.

Local Installation

To install a single component locally:

```
./debo --install --hdfs
```

This command installs **HDFS** and configures it using Debo's default settings.

Installing with Dependencies

Some components depend on others. For example, **HBase** requires both **HDFS** and **ZooKeeper**. You can instruct Debo to install all dependencies automatically:

```
./debo --install --hbase --with-dependency
```

Debo will detect and install **HDFS** and **ZooKeeper** along with **HBase**.

Remote Installation

To install a component on a remote machine, ensure the **Debo Agent** is already installed and running on the target machine. The agent listens on port `1221` by default.

```
./debo --install --hdfs --host=" remote-host " --port="1221"
```

To include dependencies:

```
./debo --install --hbase --with-dependency --host=" remote-host " --port="1221"
```

Full Stack Installation

To install **all supported components**:

- **Locally**:

./debo --install --all

- **Remotely**:

./debo --install --all --host=" remote-host " --port="1221"

This installs and configures the full Hadoop ecosystem managed by Debo.

💡 **Reminder:**
Ensure sufficient system privileges and proper environment setup before installation. For remote installations, the agent must be running with enough permissions to manage system-level operations like service control and configuration file edits.

START Action

Once the Apache components are installed using **Debo**, you can use the START action to launch and initialize them. This ensures that all required services are running in the correct order, with proper dependencies activated where necessary.

Basic Usage

To start a specific component on the **local machine**, run:

```
./debo --start --hdfs
```

This command will start the HDFS service using the configuration previously set by Debo during installation.

Starting with Dependencies

If you want to start a component along with all of its required dependencies, use the `--with-dependency` option:

```
./debo --start --hbase --with-dependency
```

In this example, **HDFS** and **ZooKeeper** will be started before HBase, ensuring all services are available in the correct sequence.

Remote Start

To start services on a **remote machine**, make sure:

- The **Debo Agent** is installed and running on the target system.
- The agent is listening on port **1221** (default).

Use the following command syntax:

```
./debo --start --kafka --host="remote-host" --port="1221"
```

To start a remote component **with dependencies**, use:

```
./debo --start --hive --with-dependency --host="remote-host" --port="1221"
```

Starting All Components

To start **all components** on the local machine:

```
./debo --start --all
```

Or, for a **remote** machine:

```
./debo --start --all --host="remote-host" --port="1221"
```

📝 **Note:** Debo ensures components are started in the correct order based on their dependencies. Services that are already running will be skipped unless explicitly stopped beforehand.

STOP Action

The **STOP** action in **Debo** is used to safely shut down any installed Apache components, either individually or all at once. It ensures that services are stopped in the correct order, especially in cases where components depend on one another (e.g., HBase depends on ZooKeeper and HDFS).

Basic Usage

To stop a single component on the **local machine**, use:

```
./debo --stop --hdfs
```

This command stops the **HDFS** service gracefully using the configuration and state previously handled by Debo.

Stopping with Dependencies

If the component you're stopping relies on other services, or you want to shut down its entire dependency tree, use:

```
./debo --stop --hbase --with-dependency
```

This command will stop **HBase**, **ZooKeeper**, and **HDFS** in the proper order.

Remote Stop

To stop services on a **remote machine**, ensure:

- **Debo Agent** is installed and running on the remote host.
- The agent is listening on the default port **1221**.

A typical remote stop command looks like:

```
./debo --stop --spark --host="remote-host" --port="1221"
```

To stop a remote service **along with its dependencies**, use:

```
./debo --stop --hive --with-dependency --host="remote-host" --port="1221"
```

Stopping All Components

To stop **all components** on the **local machine**, run:

```
./debo --stop --all
```

For stopping all services on a **remote machine**:

```
./debo --stop --all --host="remote-host" --port="1221"
```

🔁 **Debo handles service shutdown sequencing automatically**, ensuring dependent services are not terminated before the services that rely on them have stopped.

RESTART Action

The **RESTART** action in **Debo** allows you to seamlessly stop and then re-start any installed Apache component—either on the local machine or a remote host. This is particularly useful when applying configuration changes or recovering services after failure, without fully uninstalling or manually restarting them.

Basic Usage

To restart a specific component on the **local machine**, run:

```
./debo --restart --hdfs
```

Debo will automatically stop and then re-start the **HDFS** service in the correct order, ensuring minimal downtime.

Restarting with Dependencies

If your component depends on other services, or if you want to restart the full service chain, include the `--with-dependency` flag:

```
./debo --restart --hbase --with-dependency
```

This command restarts **HBase**, **ZooKeeper**, and **HDFS** in the appropriate sequence, ensuring all services return to a consistent state.

Remote Restart

To restart a component on a **remote machine**, confirm that:

- The **Debo Agent** is installed and actively running on the target host.
- The agent is listening on the default port **1221**.

Example command:

```
./debo --restart --spark --host="remote-host" --port="1221"
```

To restart a remote component **with dependencies**, use:

```
./debo --restart --hive --with-dependency --host="remote-host" --port="1221"
```

Restarting All Components

To restart **all services** on the **local system**, run:

```
./debo --restart --all
```

For remote full-cluster restarts:

```
./debo --restart --all --host="remote-host" --port="1221"
```

❗ Debo takes care of stopping and starting each component in the correct dependency order, minimizing risks during cluster-wide restarts.

REPORT Action

The **REPORT** action in **Debo** provides detailed information about the current status, version, configuration, and runtime state of installed components. It is designed to help users monitor and audit the health and setup of their Hadoop ecosystem services.

Unlike other actions, the **REPORT** action is implicit—**no explicit `--report` flag is needed**. When you run a command like `./debo --hdfs`, Debo automatically detects that no action was specified and enters **report mode**.

Basic Usage

To generate a report about a specific component on the **local machine**, use:

```
./debo --hdfs
```

Reporting with Dependencies

If you'd like to view the status of a component **along with its dependencies**, include the `--with-dependency` flag:

```
./debo --hbase --with-dependency
```

This will return a comprehensive report for **HBase**, **ZooKeeper**, and **HDFS**, including the relationships between them.

Reporting All Components

To get a full system-wide report of **all installed components**:

```
./debo --all
```

Debo will generate a consolidated report showing each component’s Runtime status

📝 Use this command regularly to verify cluster health, version consistency, and configuration integrity across all services.

Remote Reporting

You can also generate reports for remote systems by providing the --host and --port parameters (port defaults to **1221**):

```
./debo --hive --with-dependency --host="remote-host" --port="1221"
```

Or for all remote components:

```
./debo --all --host="remote-host" --port="1221"
```

💡 **Tip:** The REPORT action is read-only and safe to run at any time. It is especially useful before performing version switches, updates, or reconfiguration.

VERSWITCH Action

The **VERSWITCH** action in **Debo** allows you to **uninstall the currently installed version** of a component and replace it with a **new specified version**. This feature is ideal for upgrading, downgrading, or testing compatibility with different versions of Apache components.

🔢 **Important:** The list of supported versions for each component is documented in the **Version Supported** section of this manual. Only those versions may be used with the `--verswitch` action.

Basic Usage

To switch the version of a component on the **local machine**, use the following format:

```
./debo --hdfs --verswitch="3.3.6"
```

This command will:

1. Uninstall the currently installed **HDFS** version.
2. Download and install **version 3.3.6** of HDFS.
3. Automatically apply Debo's **default configuration settings** for that version.

⚠ **Note:** Any custom configuration from the previous version will be lost unless backed up and reapplied manually.

Remote Version Switching

To switch a component's version on a **remote machine**, ensure:

- **Debo Agent** is installed and actively running.
- The agent is listening on the default port **1221**.

Command example:

```
./debo --spark --verswitch="3.5.0" --host="remote-host" --port="1221"
```

This will replace the current **Spark** installation with **version 3.5.0** on the remote host, and configure it using Debo's default templates.

⊘ Not Supported with Dependencies or --all

The --verswitch action:

- Must be used **only on individual components**.
- **Cannot** be combined with --all or --with-dependency.

📝 **Tip:** After switching versions, you can use the --configure action to apply any necessary custom parameters. For a full list of available configuration keys, refer to the **Appendix** section of this manual.

CONFIGURE Action

The **CONFIGURE** action in **Debo** provides a convenient way to apply or modify specific configuration parameters for any installed Apache component. This gives users precise control over service behavior without manually editing configuration files.

Unlike other actions such as --install, --start, or --stop, the --configure action does **not** support --all or --with-dependency flags. Configuration changes are applied **individually** to each specified component.

Basic Usage

To configure a specific parameter for a component on the **local machine**, use the following format:

```
./debo --hdfs --configure="dfs.replication" --value="3"
```

This command sets the **dfs.replication** parameter in **HDFS** to **3**.

✓ You can repeat this command to set additional parameters as needed.

Remote Configuration

To apply a configuration to a **remote machine**, ensure:

- **Debo Agent** is installed and running on the remote host.
- The agent is listening on the default port **1221**.

Use this syntax:

```
./debo --yarn --configure="yarn.nodemanager.resource.memory-mb" --value="4096" --host="remote-host" --port="1221"
```

This sets the **memory limit** for YARN NodeManager on the specified remote machine.

📖 Configuration Reference

Each Apache component supports a specific set of configurable parameters.
A complete list of supported configuration keys for each component is provided in the **Appendix** at the end of this manual.

📝 **Note:** Debo validates the specified parameter before applying changes. If an unsupported or unknown parameter is provided, the command will be rejected with a helpful error message.

VERSWITCH Action

The **VERSWITCH** action in **Debo** allows you to **uninstall the currently installed version** of a component and replace it with a **new specified version**. This feature is ideal for upgrading, downgrading, or testing compatibility with different versions of Apache components.

🔢 **Important:** The list of supported versions for each component is documented in the **Version Supported** section of this manual. Only those versions may be used with the `--verswitch` action.

Basic Usage

To switch the version of a component on the **local machine**, use the following format:

```
./debo --hdfs --verswitch="3.3.6"
```

This command will:

1. Uninstall the currently installed **HDFS** version.
2. Download and install **version 3.3.6** of HDFS.
3. Automatically apply Debo's **default configuration settings** for that version.

⚠ **Note:** Any custom configuration from the previous version will be lost unless backed up and reapplied manually.

Remote Version Switching

To switch a component's version on a **remote machine**, ensure:

- **Debo Agent** is installed and actively running.
- The agent is listening on the default port **1221**.

Command example:

```
./debo --spark --verswitch="3.5.0" --host="remote-host" --port="1221"
```

This will replace the current **Spark** installation with **version 3.5.0** on the remote host, and configure it using Debo's default templates.

⊘ Not Supported with Dependencies or --all

The --verswitch action:

- Must be used **only on individual components**.
- **Cannot** be combined with --all or --with-dependency.

📝 **Tip:** After switching versions, you can use the --configure action to apply any necessary custom parameters. For a full list of available configuration keys, refer to the **Appendix** section of this manual.

UNINSTALL Action

The **UNINSTALL** action in **Debo** allows users to completely remove one or more installed Apache components from either the local machine or a remote host. This action ensures all related service files, configurations, logs, and binaries managed by Debo are properly cleaned up.

You can uninstall:

- A **single component**
- A component **with all its dependencies**
- **All components** at once

Basic Usage

To uninstall a specific component from the **local machine**, use:

```
./debo --uninstall --hdfs
```

This will remove the **HDFS** service, including its configuration files, logs, binaries, and installation directory (managed by Debo).

Uninstalling with Dependencies

To uninstall a component **along with all its dependencies**, include the --with-dependency flag:

```
./debo --uninstall --hbase --with-dependency
```

This command removes **HBase**, as well as **ZooKeeper** and **HDFS**—in the proper order.

Uninstalling All Components

To remove **all installed components** from the local system:

```
./debo --uninstall --all
```

This is useful when resetting the entire environment or preparing for a fresh installation.

Remote Uninstallation

To uninstall a component on a **remote host**, make sure:

- The **Debo Agent** is running on the remote machine.
- It is listening on the default port **1221**.

Example command:

```
./debo --uninstall --spark --host="remote-host" --port="1221"
```

To uninstall with dependencies remotely:

```
./debo --uninstall --hive --with-dependency --host="remote-host" --port="1221"
```

To uninstall **everything** remotely:

```
./debo --uninstall --all --host="remote-host" --port="1221"
```

⚠ **Caution:** The uninstall process is irreversible. Ensure you've backed up any important configurations or data before running this command.

💡 You can run a REPORT action before uninstalling to confirm the exact state and dependencies of the components you plan to remove.

- Debo Server running on the control node.

- Debo Agent installed and running on each managed node.

- Ensure network communication (default port 1221) is open.

- Have IP addresses or hostnames of target nodes.

Administrator Checklist

Before deploying or managing services using **Debo**, system administrators should verify the following operational prerequisites to ensure stable and predictable cluster behavior:

- ✅ **Verify Agent Availability**
  Ensure all Debo agents are running on each managed node. Each agent must be active and reachable from the Debo server.
- ✅ **Sufficient Privileges**
  Confirm that all agents are running with the necessary system privileges to perform Hadoop-related operations such as starting services, reading configuration files, and accessing Hadoop file system locations.
- ✅ **Firewall and Network Configuration**
  Maintain proper network and firewall settings to allow uninterrupted communication between the Debo server and agents. By default, Debo agents listen on **port 1221**, which must be open and accessible across all relevant nodes.
- ✅ **Command Help Reference**
  When using the CLI, always refer to the latest flag options and usage patterns by running:

  ```
  debo --help
  ```

  This ensures you're working with up-to-date syntax and features.

Contact & Support

We are committed to making **Debo** a powerful and accessible tool for managing your Hadoop clusters. If you need assistance beyond what is covered in this manual, we are pleased to offer additional support options.

📧 Paid Help & Professional Support

For tailored, one-on-one assistance, advanced troubleshooting, or deployment guidance, we offer professional support services. To inquire about rates and availability, please contact us directly at:
deboteamwork@gmail.com

💖 Support the Project

If you find **Debo** valuable and would like to support its ongoing development, you can contribute by donating to our PayPal:
deboteamwork@gmail.com

Your donation helps cover development time, documentation efforts, and testing resources.

🖥️ Infrastructure Contributions

We also welcome infrastructure support (e.g., cloud resources, test servers, or CI/CD tooling) to help us improve and expand the project. If you're interested in sponsoring infrastructure or providing in-kind support, please reach out to us at the same email address above.

Special Note on Apache Ranger and Apache Atlas Integration

During the integration of **Apache Ranger** and **Apache Atlas** into the Debo CLI system, we identified a recurring issue that affects their automated installation process. The problem stems from dependencies being hosted on a private Hortonworks Nexus repository that is no longer publicly accessible.

Problem Summary

When attempting to build or install Ranger or Atlas, Maven fails to retrieve required metadata from the following private Hortonworks repository:

https://nexus-private.hortonworks.com/nexus/content/groups/public/

This results in errors similar to the following:

```
[WARNING] Could not transfer metadata net.minidev:json-smart/maven-metadata.xml from/to hortonworks.repo ...
transfer failed for https://nexus-private.hortonworks.com/...
```

Why It Matters

Due to this issue, Debo cannot currently perform a complete end-to-end setup of Apache Ranger and Apache Atlas out of the box using public repositories. However, rather than removing support for these components, we have chosen to **retain their implementation** within the Debo CLI tool for the following reasons:

- Users who have access to the appropriate **Hortonworks artifacts** or maintain their own **mirror repositories** can still take advantage of the existing functionality.
- It allows enterprise or advanced users with internal package caches or legacy distributions to continue managing Ranger and Atlas via Debo.
- The integration logic and service lifecycle operations (start, stop, configure, etc.) remain intact and functional once the software is installed.

Looking Ahead

The Debo team is committed to providing a seamless experience for all supported components, including Apache Ranger and Apache Atlas. To address the current limitations caused by the reliance on private Hortonworks repositories, **Debo will introduce its own solutions** in future releases. Planned improvements include:

- **Providing alternative installation paths** that do not depend on the discontinued Hortonworks Nexus repository.
- **Packaging pre-built or verified versions** of Ranger and Atlas, hosted on a **Debo-managed infrastructure**, ensuring reliability and availability.
- **Offering configuration templates and manual integration guides** for users who prefer or need to install these components independently.
- **Detecting and adapting to enterprise environments** where internal mirrors or proprietary package caches are available.

Our goal is to ensure that Ranger and Atlas can be installed and managed as reliably as other components, without requiring privileged access to deprecated or private resources.

We welcome collaboration and feedback from the community to help us refine these solutions and meet real-world needs.

Appendix: Configuration Parameters

This appendix serves as the definitive reference for the core configuration parameters managed and validated by the Debo system. These parameters play a vital role in ensuring proper integration and control over various components within the Hadoop ecosystem.

Each listed configuration entry reflects what Debo checks for correctness or attempts to set during installation, startup, or runtime validation procedures. Users are encouraged to review these parameters when preparing or troubleshooting their Hadoop environment.

Missing a Critical Parameter?

The parameters listed here are essential for Debo's core operations. If you believe a crucial configuration parameter required for your specific Hadoop setup is missing from this appendix, please help us improve the documentation by notifying the Debo team. Contact us directly at:

deboteamwork@gmail.com

Include the name of the parameter, its typical location (e.g., hdfs-site.xml, yarn-site.xml), and a brief description of its purpose. Your feedback is valuable for enhancing Debo.

Flink

jobmanager.rpc.address
jobmanager.rpc.port
jobmanager.heap.size
taskmanager.heap.size
taskmanager.numberOfTaskSlots
parallelism.default
io.tmp.dirs
classloader.resolve-order

// State Backend & Checkpointing
state.backend
state.checkpoints.dir
state.savepoints.dir
checkpoint.interval
execution.checkpointing.interval
state.backend.incremental
state.backend.async
state.backend.rocksdb.ttl-compaction-filter.enabled

// Rest & Web UI
rest.port
rest.address
web.timeout
web.submit.enable
web.upload.dir
web.access-control-allow-origin

// High Availability
high-availability
high-availability.storageDir
high-availability.zookeeper.quorum
high-availability.zookeeper.path.root
high-availability.cluster-id

// Security
security.ssl.enabled
security.ssl.keystore
security.ssl.keystore-password
security.ssl.key-password
security.ssl.truststore
security.ssl.truststore-password
security.kerberos.login.keytab

// Network & Communication
taskmanager.data.port
taskmanager.data.ssl.enabled
blob.server.port
queryable-state.proxy.ports
akka.ask.timeout
akka.framesize

// Memory Management
taskmanager.memory.framework.heap.size
taskmanager.memory.network.min
taskmanager.memory.managed.size
taskmanager.memory.managed.fraction
jobmanager.memory.off-heap.size

// YARN Deployment
yarn.application.name
yarn.application.queue
yarn.containers.vcores
yarn.containers.memory
yarn.ship-files

// Kubernetes Deployment
kubernetes.cluster-id
kubernetes.namespace
kubernetes.service.account
kubernetes.container.image

// Mesos Deployment
mesos.resourcemanager.tasks.cpus
mesos.resourcemanager.tasks.mem

// Metrics & Monitoring
metrics.reporter.prom.class
metrics.reporter.prom.port

metrics.system-resource
// Parameters from log4j-cli.properties (28 entries)
monitorInterval
rootLogger.level
rootLogger.appenderRef.file.ref
appender.file.name
appender.file.type
appender.file.append
appender.file.fileName
appender.file.layout.type
appender.file.layout.pattern
logger.yarn.name
logger.yarn.level
logger.yarn.appenderRef.console.ref
logger.yarncli.name
logger.yarncli.level
logger.yarncli.appenderRef.console.ref
logger.hadoop.name
logger.hadoop.level
logger.hadoop.appenderRef.console.ref
logger.hive.name
logger.hive.level
logger.hive.additivity
logger.hive.appenderRef.file.ref
logger.kubernetes.name
logger.kubernetes.level
logger.kubernetes.appenderRef.console.ref
appender.console.name
appender.console.type
appender.console.layout.type
appender.console.layout.pattern
logger.hadoopnative.name
logger.hadoopnative.level
logger.netty.name
logger.netty.level
// New parameters from log4j-console.properties (35 entries)
monitorInterval
rootLogger.level
rootLogger.appenderRef.console.ref
// Failover & Recovery
jobmanager.execution.failover-strategy
restart-strategy
restart-strategy.fixed-delay.attempts
logger.zookeeper.name
logger.zookeeper.level
logger.shaded_zookeeper.name
logger.shaded_zookeeper.level
appender.main.name
appender.main.type
appender.main.append
appender.main.fileName
appender.main.filePattern
appender.main.layout.type
appender.main.layout.pattern
appender.main.policies.type
appender.main.policies.size.type
appender.main.policies.size.size
appender.main.policies.startup.type
appender.main.strategy.type
appender.main.strategy.max
logger.netty.name
logger.netty.level
monitorInterval
rootLogger.level
rootLogger.appenderRef.console.ref
appender.console.name
appender.console.type
appender.console.layout.type
appender.console.layout.pattern
logger.netty.name
logger.netty.level
logger.zookeeper.name
logger.zookeeper.level
logger.shaded_zookeeper.name
logger.shaded_zookeeper.level
logger.curator.name
logger.curator.level
logger.runtimeutils.name
logger.runtimeutils.level
logger.runtimeleader.name
logger.runtimeleader.level

rootLogger.appenderRef.rolling.ref
appender.console.name
appender.console.type
appender.console.layout.type
appender.console.layout.pattern
appender.console.filter.threshold.type
appender.console.filter.threshold.level
appender.rolling.name
appender.rolling.type
appender.rolling.append
appender.rolling.fileName
appender.rolling.filePattern
appender.rolling.layout.type
appender.rolling.layout.pattern
appender.rolling.policies.type
appender.rolling.policies.size.type
appender.rolling.policies.size.size
appender.rolling.policies.startup.type
appender.rolling.strategy.type
appender.rolling.strategy.max
logger.pekko.name
logger.pekko.level
logger.kafka.name
logger.kafka.level
logger.hadoop.name
logger.hadoop.level
logger.zookeeper.name
logger.zookeeper.level
logger.shaded_zookeeper.name
logger.shaded_zookeeper.level
logger.netty.name
logger.netty.level
monitorInterval
rootLogger.level
rootLogger.appenderRef.file.ref
logger.pekko.name
logger.pekko.level
logger.kafka.name
logger.kafka.level
logger.hadoop.name
logger.hadoop.level

Hbase

hbase.rootdir
hbase.zookeeper.quorum
hbase.hregion.max.filesize
hbase.hstore.blockingStoreFiles
hbase.rpc.timeout
hbase.hregion.majorcompaction
hbase.tmp.dir
hbase.cluster.distributed
hbase.zookeeper.property.clientPort
hbase.regionserver.handler.count
hbase.master.info.port
hbase.regionserver.info.port
hbase.hstore.compactionThreshold
hbase.hstore.blockingWaitTime
hbase.client.write.buffer
hbase.security.authentication
hbase.security.authorization
hbase.superuser
hbase.coprocessor.region.classes
hbase.rest.port
// HBase policy parameters (from hbase-policy.xml)
security.client.protocol.acl
security.admin.protocol.acl
security.master.protocol.acl
security.regionserver.protocol.acl

// New core-site.xml parameters (extended list)
io.native.lib.available
hadoop.http.filter.initializers
hadoop.security.authorization
hadoop.security.authentication
hadoop.security.group.mapping
hadoop.rpc.protection
fs.permissions.umask-mode
io.file.buffer.size
io.bytes.per.checksum
io.compression.codecs
hadoop.security.auth_to_local
hadoop.proxyuser.knox.groups
hadoop.proxyuser.knox.hosts
hbase.defaults.for.version

// Advanced Client Behavior
fs.oci.client.hostname
fs.oci.client.custom.authenticator
fs.viprfs.impl
fs.AbstractFileSystem.viprfs.impl
hadoop.security.dns.interface
hadoop.security.groups.cache.secs
viprfs.security.principal
// Core Audit Destinations
// Connection/Timeout/Retry
hbase.client.operation.timeout
hbase.client.scanner.timeout.period
hbase.client.pause
hbase.client.retries.number
hbase.client.ipc.pool.size

hbase.zookeeper.property.session.timeout
hbase.client.connection.maxidletime

// Security
hbase.security.auth.enable
hbase.rpc.protection
hbase.sasl.clientconfig
hbase.kerberos.regionserver.principal
hbase.regionserver.kerberos.principal

// Caching/Buffering
hbase.client.scanner.caching
hbase.client.keyvalue.maxsize
hbase.client.scanner.max.result.size

// Region/Meta
hbase.client.meta.operation.timeout
hbase.client.localityCheck.interval
hbase.client.prefetch.limit
hbase.meta.replicas.use

// SSL/TLS
hbase.rpc.ssl.enabled
hbase.ssl.enabled
hbase.rest.ssl.enabled
hbase.ssl.keystore.store
hbase.ssl.keystore.password
hbase.ssl.truststore.store
hbase.ssl.truststore.password

hbase.client.scanner.lease.period
hbase.client.primaryCallTimeout.get
hbase.client.primaryCallTimeout.multiget
hbase.client.hedged.read.timeout
hbase.client.hedged.read.threadpool.size
xasecure.audit.is.enabled
xasecure.audit.hdfs.is.enabled
xasecure.audit.hdfs.is.async
xasecure.audit.hdfs.async.max.queue.size

xasecure.audit.hdfs.async.max.flush.interval.ms
xasecure.audit.hdfs.config.encoding

xasecure.audit.hdfs.config.destination.directory

xasecure.audit.hdfs.config.destination.file

xasecure.audit.hdfs.config.destination.flush.interval.seconds

xasecure.audit.hdfs.config.destination.rollover.interval.seconds

xasecure.audit.hdfs.config.destination.open.retry.interval.seconds

xasecure.audit.hdfs.config.local.buffer.directory
xasecure.audit.hdfs.config.local.buffer.file

xasecure.audit.hdfs.config.local.buffer.file.buffer.size.bytes

xasecure.audit.hdfs.config.local.buffer.flush.interval.seconds

xasecure.audit.hdfs.config.local.buffer.rollover.interval.seconds

xasecure.audit.hdfs.config.local.archive.directory

xasecure.audit.hdfs.config.local.archive.max.file.count
xasecure.audit.log4j.is.enabled
xasecure.audit.log4j.is.async
xasecure.audit.log4j.async.max.queue.size

xasecure.audit.log4j.async.max.flush.interval.ms
xasecure.audit.kafka.is.enabled

// Serialization/Compatibility

// Ranger HBase SSL policy manager parameters (from ranger-hbase-policymgr-ssl.xml)
xasecure.policymgr.clientssl.keystore
xasecure.policymgr.clientssl.truststore

xasecure.policymgr.clientssl.keystore.credential.file

xasecure.policymgr.clientssl.truststore.credential.file
ranger.plugin.hbase.service.name
ranger.plugin.hbase.policy.source.impl
ranger.plugin.hbase.policy.rest.url

ranger.plugin.hbase.policy.rest.ssl.config.file

ranger.plugin.hbase.policy.pollIntervalMs
ranger.plugin.hbase.policy.cache.dir

xasecure.hbase.update.xapolicies.on.grant.revoke

ranger.plugin.hbase.policy.rest.client.connection.timeoutMs

ranger.plugin.hbase.policy.rest.client.read.timeoutMs
log4j.rootlogger
log4j.threshold
log4j.appender.stdout
log4j.appender.stdout.layout

log4j.appender.stdout.layout.conversionpattern
log4j.logger.org.apache.hadoop

log4j.logger.org.apache.hadoop.metrics2
log4j.logger.org.apache.hadoop.fs
log4j.org.apache.hadoop.util
log4j.logger.org.apache.hadoop.fs.s3a

log4j.logger.org.apache.hadoop.hbase.oss
hbase.root.logger
hbase.security.logger
hbase.log.dir

xasecure.audit.kafka.async.max.queue.size

xasecure.audit.kafka.async.max.flush.interval.ms
xasecure.audit.kafka.broker_list
xasecure.audit.kafka.topic_name
xasecure.audit.solr.is.enabled
xasecure.audit.solr.async.max.queue.size

xasecure.audit.solr.async.max.flush.interval.ms
xasecure.audit.solr.solr_url
common.name.for.certificate
policy_user
ranger-hbase-plugin-enabled
REPOSITORY_CONFIG_USERNAME

hbase.log.file
log4j.rootLogger
log4j.threshold
log4j.appender.DRFA
// ... (all other log4j properties from calls) ...

// HBase Site Parameters
hbase_log_maxfilesize
hbase_log_maxbackupindex
hbase_security_log_maxfilesize
hbase_security_log_maxbackupindex
hbase.master.port
phoenix.rpc.index.handler.count
// ... (all other missing hbase-site params) ...

// Ranger Plugin Properties

HDFS

dfs.replication

dfs.namenode.name.dir

dfs.datanode.data.dir

fs.defaultFS

hadoop.tmp.dir

dfs.blocksize

dfs.namenode.checkpoint.dir

dfs.permissions.enabled

dfs.client.use.datanode.hostname

dfs.datanode.address

dfs.datanode.http.address

dfs.datanode.ipc.address

dfs.namenode.http-address

dfs.namenode.https-address

dfs.namenode.rpc-address

dfs.hosts.exclude

dfs.datanode.failed.volumes.tolerated

dfs.datanode.max.transfer.threads

io.file.buffer.size

dfs.namenode.acls.enabled

// Storage management (6)

dfs.datanode.du.reserved

dfs.storage.policy.satisfier.mode

dfs.namenode.num.extra.edits.retained

dfs.datanode.data.dir.perm

dfs.namenode.delegation.key.update-interval

dfs.namenode.delegation.token.max-lifetime

// Fault tolerance (7)

dfs.namenode.checkpoint.period

dfs.namenode.num.checkpoints.retained

dfs.client.block.write.replace-datanode-on-failure.policy

dfs.client.block.write.replace-datanode-on-failure.enable

dfs.client.block.write.replace-datanode-on-failure.best-effort

dfs.namenode.replication.min

dfs.heartbeat.interval

// Performance tuning (10)

dfs.client.read.shortcircuit

dfs.domain.socket.path

dfs.client.socket-timeout

dfs.datanode.balance.bandwidthPerSec

dfs.client.max.block.acquire.failures

dfs.namenode.handler.count

dfs.datanode.handler.count

dfs.client.write.packet.size

dfs.replication.interval

dfs.namenode.replication.work.multiplier.per.iteration

// Security configurations (7)

dfs.encrypt.data.transfer

dfs.encrypt.data.transfer.algorithm

dfs.http.policy

dfs.https.port

hadoop.security.authentication

hadoop.security.authorization

hadoop.rpc.protection

// Network/RPC settings (5)

dfs.datanode.hostname

dfs.namenode.secondary.http-address

dfs.namenode.backup.address

dfs.journalnode.rpc-address

dfs.journalnode.http-address

// Cluster management (6)

dfs.hosts

dfs.namenode.safemode.threshold-pct

dfs.ha.automatic-failover.enabled

dfs.namenode.audit.loggers

dfs.client.failover.proxy.provider

dfs.namenode.replication.considerLoad

// Client behavior (5)

dfs.client.retry.policy.enabled

dfs.client.retry.max.attempts

dfs.client.failover.sleep.base.millis

dfs.client.hedged.read.threadpool.size

dfs.client.hedged.read.threshold.millis

// DataNode advanced configs (4)

dfs.datanode.max.locked.memory

dfs.datanode.socket.write.timeout

dfs.image.compress

dfs.image.compression.codec

// Quota management (2)

dfs.namenode.quota.enabled

dfs.namenode.quota.update.interval

// Ranger audit parameters from ranger-hdfs-audit.xml

xasecure.audit.is.enabled

xasecure.audit.hdfs.is.enabled

xasecure.audit.hdfs.is.async

xasecure.audit.hdfs.async.max.queue.size

xasecure.audit.hdfs.async.max.flush.interval.ms

xasecure.audit.hdfs.config.encoding

xasecure.audit.hdfs.config.destination.directory

xasecure.audit.hdfs.config.destination.file

xasecure.audit.hdfs.config.destination.flush.interval.seconds

xasecure.audit.hdfs.config.destination.rollover.interval.seconds

xasecure.audit.hdfs.config.destination.open.retry.interval.seconds

xasecure.audit.hdfs.config.local.buffer.directory

xasecure.audit.hdfs.config.local.buffer.file

xasecure.audit.hdfs.config.local.buffer.file.buffer.size.bytes

xasecure.audit.hdfs.config.local.buffer.flush.interval.seconds

xasecure.audit.hdfs.config.local.buffer.rollover.interval.seconds

xasecure.audit.hdfs.config.local.archive.directory

xasecure.audit.hdfs.config.local.archive.max.file.count

xasecure.audit.log4j.is.enabled

xasecure.audit.log4j.is.async

xasecure.audit.log4j.async.max.queue.size

xasecure.audit.log4j.async.max.flush.interval.ms

xasecure.audit.kafka.is.enabled

xasecure.audit.kafka.async.max.queue.size

xasecure.audit.kafka.async.max.flush.interval.ms

xasecure.audit.kafka.broker_list

xasecure.audit.kafka.topic_name

xasecure.audit.solr.is.enabled

xasecure.audit.solr.async.max.queue.size

xasecure.audit.solr.async.max.flush.interval.ms

xasecure.audit.solr.solr_url

// SSL configuration parameters from ranger-policymgr-ssl.xml

xasecure.policymgr.clientssl.keystore

xasecure.policymgr.clientssl.truststore

xasecure.policymgr.clientssl.keystore.credential.file

xasecure.policymgr.clientssl.truststore.credential.file

// Ranger security parameters from ranger-hdfs-security.xml

ranger.plugin.hdfs.service.name

ranger.plugin.hdfs.policy.source.impl

ranger.plugin.hdfs.policy.rest.url

ranger.plugin.hdfs.policy.rest.ssl.config.file

ranger.plugin.hdfs.policy.pollIntervalMs

ranger.plugin.hdfs.policy.cache.dir

ranger.plugin.hdfs.policy.rest.client.connection.timeoutMs

ranger.plugin.hdfs.policy.rest.client.read.timeoutMs

xasecure.add-hadoop-authorization

log4j.rootLogger

log4j.threshhold

log4j.appender.stdout

log4j.appender.stdout.layout

log4j.appender.stdout.layout.ConversionPattern

log4j.appender.subprocess

log4j.appender.subprocess.layout

log4j.appender.subprocess.layout.ConversionPattern
log4j.logger.org.apache.hadoop.yarn.registry log4j.logger.org.apache.hadoop.service
log4j.logger.org.apache.hadoop.security.UserGroupInformation
log4j.logger.org.apache.hadoop.util.NativeCodeLoader
log4j.logger.org.apache.hadoop.hdfs.server.datanode.BlockPoolSliceScanner

log4j.logger.org.apache.hadoop.hdfs.server.blockmanagement
log4j.logger.org.apache.hadoop.hdfs.server.namenode.FSNamesystem.audit

log4j.logger.org.apache.hadoop.hdfs
log4j.logger.org.apache.hadoop.yarn.server.nodemanager.containermanager.monitor

log4j.logger.org.apache.hadoop.yarn.server.nodemanager.NodeStatusUpdaterImpl

log4j.logger.org.apache.zookeeper

log4j.logger.org.apache.zookeeper.ClientCnxn
log4j.logger.org.apache.hadoop.yarn.server.resourcemanager.security

log4j.logger.org.apache.hadoop.metrics2 log4j.logger.org.apache.hadoop.util.HostsFileReader
log4j.logger.org.apache.hadoop.yarn.event.AsyncDispatcher
log4j.logger.org.apache.hadoop.security.token.delegation
log4j.logger.org.apache.hadoop.yarn.util.AbstractLivelinessMonitor
log4j.logger.org.apache.hadoop.yarn.server.nodemanager.security
log4j.logger.org.apache.hadoop.yarn.server.resourcemanager.RMNMInfo
log4j.logger.org.apache.curator.framework.imps
log4j.logger.org.apache.curator.framework.state.ConnectionStateManager
log4j.logger.org.apache.directory.api.ldap log4j.logger.org.apache.directory.server

// Service-Level Authorization Parameters from hadoop-policy.xml (11 entries)

security.client.protocol.acl

security.client.datanode.protocol.acl

security.datanode.protocol.acl

security.inter.datanode.protocol.acl

security.namenode.protocol.acl

security.inter.tracker.protocol.acl

security.job.submission.protocol.acl

security.task.umbilical.protocol.acl

security.refresh.policy.protocol.acl

security.admin.operations.protocol.acl

security.ha.service.protocol.acl

// RBF-specific parameters (49)

dfs.federation.router.rpc.enable

dfs.federation.router.rpc-address

dfs.federation.router.rpc-bind-host

dfs.federation.router.handler.count

dfs.federation.router.handler.queue.size

dfs.federation.router.reader.count

dfs.federation.router.reader.queue.size

dfs.federation.router.connection.creator.queue-size

dfs.federation.router.connection.pool-size

dfs.federation.router.connection.min-active-ratio

dfs.federation.router.connection.clean.ms

dfs.federation.router.connection.pool.clean.ms

dfs.federation.router.metrics.enable

dfs.federation.router.dn-report.time-out

dfs.federation.router.dn-report.cache-expire

dfs.federation.router.metrics.class

dfs.federation.router.admin.enable

dfs.federation.router.admin-address

dfs.federation.router.admin-bind-host

dfs.federation.router.admin.handler.count

dfs.federation.router.http-address

dfs.federation.router.http-bind-host

dfs.federation.router.https-address

dfs.federation.router.https-bind-host

dfs.federation.router.http.enable

dfs.federation.router.file.resolver.client.class

dfs.federation.router.namenode.resolver.client.class

dfs.federation.router.store.enable

dfs.federation.router.store.serializer

dfs.federation.router.store.driver.class

dfs.federation.router.store.connection.test

dfs.federation.router.cache.ttl

dfs.federation.router.store.membership.expiration

dfs.federation.router.store.membership.expiration.deletion

dfs.federation.router.heartbeat.enable

dfs.federation.router.heartbeat.interval

dfs.federation.router.heartbeat-state.interval

dfs.federation.router.namenode.heartbeat.enable

dfs.federation.router.store.router.expiration

dfs.federation.router.safemode.enable

dfs.federation.router.safemode.extension

dfs.federation.router.safemode.expiration

dfs.federation.router.monitor.localnamenode.enable

dfs.federation.router.mount-table.max-cache-size

dfs.federation.router.mount-table.cache.enable

dfs.federation.router.quota.enable

dfs.federation.router.quota-cache.update.interval

dfs.federation.router.client.thread-size

dfs.federation.router.client.retry.max.attempts

dfs.federation.router.client.reject.overload

dfs.federation.router.client.allow-partial-listing

dfs.federation.router.client.mount-status.time-out

dfs.federation.router.connect.max.retries.on.timeouts

dfs.federation.router.connect.timeout

dfs.federation.router.mount-table.cache.update

dfs.federation.router.mount-table.cache.update.timeout

dfs.federation.router.mount-table.cache.update.client.max.time

dfs.federation.router.secret.manager.class

// New SSL client parameters from ssl-client.xml (7 entries)

ssl.client.truststore.location

ssl.client.truststore.type

ssl.client.truststore.password

ssl.client.truststore.reload.interval

ssl.client.keystore.type

ssl.client.keystore.location

ssl.client.keystore.password

ssl.server.truststore.location

ssl.server.truststore.type

ssl.server.truststore.password

ssl.server.truststore.reload.interval

ssl.server.keystore.type

ssl.server.keystore.location

ssl.server.keystore.password

ssl.server.keystore.keypassword

// YARN parameters from yarn-site.xml

yarn.resourcemanager.hostname

yarn.resourcemanager.resource-tracker.address

yarn.resourcemanager.scheduler.address

yarn.resourcemanager.address

yarn.resourcemanager.admin.address

yarn.resourcemanager.scheduler.class

yarn.scheduler.minimum-allocation-mb

yarn.scheduler.maximum-allocation-mb

yarn.acl.enable

yarn.admin.acl

yarn.nodemanager.address

yarn.nodemanager.resource.memory-mb

yarn.application.classpath

yarn.nodemanager.vmem-pmem-ratio

yarn.nodemanager.container-executor.class

yarn.nodemanager.linux-container-executor.group

yarn.nodemanager.aux-services

yarn.nodemanager.aux-services.mapreduce_shuffle.class

yarn.nodemanager.log-dirs

yarn.nodemanager.local-dirs

yarn.nodemanager.container-monitor.interval-ms

yarn.nodemanager.health-checker.interval-ms

yarn.nodemanager.health-checker.script.timeout-ms

yarn.nodemanager.log.retain-seconds

yarn.log-aggregation-enable

yarn.nodemanager.remote-app-log-dir

yarn.nodemanager.remote-app-log-dir-suffix

yarn.nodemanager.log-aggregation.compression-type

yarn.nodemanager.delete.debug-delay-sec

yarn.log-aggregation.retain-seconds

yarn.nodemanager.admin-env

yarn.nodemanager.disk-health-checker.min-healthy-disks

yarn.resourcemanager.am.max-attempts

yarn.resourcemanager.webapp.address

yarn.resourcemanager.webapp.https.address

yarn.nodemanager.vmem-check-enabled

yarn.log.server.url

yarn.resourcemanager.nodes.exclude-path

manage.include.files

yarn.http.policy

yarn.timeline-service.enabled

yarn.timeline-service.generic-application-history.store-class

yarn.timeline-service.leveldb-timeline-store.path

yarn.timeline-service.webapp.address

yarn.timeline-service.webapp.https.address

yarn.timeline-service.address

yarn.timeline-service.ttl-enable

yarn.timeline-service.ttl-ms

yarn.timeline-service.leveldb-timeline-store.ttl-interval-ms

hadoop.registry.zk.quorum

hadoop.registry.dns.bind-port

hadoop.registry.dns.zone-mask

hadoop.registry.dns.zone-subnet

hadoop.registry.dns.enabled

hadoop.registry.dns.domain-name

yarn.nodemanager.recovery.enabled

yarn.nodemanager.recovery.dir

yarn.client.nodemanager-connect.retry-interval-ms

yarn.client.nodemanager-connect.max-wait-ms

yarn.resourcemanager.recovery.enabled

yarn.resourcemanager.work-preserving-recovery.enabled

yarn.resourcemanager.store.class

yarn.resourcemanager.zk-address

yarn.resourcemanager.zk-state-store.parent-path

yarn.resourcemanager.zk-acl

yarn.resourcemanager.work-preserving-recovery.scheduling-wait-ms

yarn.resourcemanager.connect.retry-interval.ms

yarn.resourcemanager.connect.max-wait.ms

yarn.resourcemanager.zk-retry-interval-ms

yarn.resourcemanager.zk-num-retries

yarn.resourcemanager.zk-timeout-ms

yarn.resourcemanager.state-store.max-completed-applications

yarn.resourcemanager.fs.state-store.retry-policy-spec

yarn.resourcemanager.fs.state-store.uri

yarn.resourcemanager.ha.enabled

yarn.nodemanager.linux-container-executor.resources-handler.class

yarn.nodemanager.linux-container-executor.cgroups.hierarchy

yarn.nodemanager.linux-container-executor.cgroups.mount

yarn.nodemanager.linux-container-executor.cgroups.mount-path

yarn.nodemanager.linux-container-executor.cgroups.strict-resource-usage

yarn.nodemanager.resource.cpu-vcores

yarn.nodemanager.resource.percentage-physical-cpu-limit

yarn.node-labels.fs-store.retry-policy-spec

yarn.nodemanager.disk-health-checker.min-free-space-per-disk-mb

yarn.nodemanager.disk-health-checker.max-disk-utilization-per-disk-percentage

yarn.nodemanager.resource-plugins

yarn.nodemanager.resource-plugins.gpu.allowed-gpu-devices

yarn.nodemanager.resource-plugins.gpu.path-to-discovery-executables

yarn.nodemanager.log-aggregation.roll-monitoring-interval-seconds

yarn.nodemanager.log-aggregation.debug-enabled

yarn.nodemanager.log-aggregation.num-log-files-per-app

yarn.resourcemanager.system-metrics-publisher.enabled

yarn.resourcemanager.system-metrics-publisher.dispatcher.pool-size

yarn.timeline-service.client.max-retries

yarn.timeline-service.client.retry-interval-ms

yarn.timeline-service.state-store-class

yarn.timeline-service.leveldb-state-store.path

yarn.timeline-service.leveldb-timeline-store.path

yarn.timeline-service.leveldb-timeline-store.read-cache-size

yarn.timeline-service.leveldb-timeline-store.start-time-read-cache-size

yarn.timeline-service.leveldb-timeline-store.start-time-write-cache-size

yarn.timeline-service.http-authentication.type

yarn.timeline-service.http-authentication.simple.anonymous.allowed

yarn.resourcemanager.webapp.delegation-token-auth-filter.enabled

yarn.resourcemanager.bind-host

yarn.nodemanager.bind-host

yarn.timeline-service.bind-host

yarn.node-labels.fs-store.root-dir

yarn.scheduler.minimum-allocation-vcores

yarn.scheduler.maximum-allocation-vcores

yarn.node-labels.enabled

yarn.resourcemanager.scheduler.monitor.enable

yarn.timeline-service.recovery.enabled

yarn.authorization-provider

yarn.timeline-service.version

yarn.timeline-service.versions

yarn.system-metricspublisher.enabled

yarn.rm.system-metricspublisher.emit-container-events

yarn.nodemanager.recovery.supervised

yarn.timeline-service.store-class

yarn.timeline-service.entity-group-fs-store.active-dir

yarn.timeline-service.entity-group-fs-store.done-dir

yarn.timeline-service.entity-group-fs-store.group-id-plugin-classes

yarn.timeline-service.entity-group-fs-store.summary-store

yarn.timeline-service.entity-group-fs-store.scan-interval-seconds

yarn.log.server.web-service.url

yarn.timeline-service.entity-group-fs-store.cleaner-interval-seconds

yarn.timeline-service.entity-group-fs-store.retain-seconds

yarn.nodemanager.container-metrics.unregister-delay-ms

yarn.timeline-service.entity-group-fs-store.group-id-plugin-classpath

yarn.resourcemanager.monitor.capacity.preemption.total_preemption_per_round

yarn.resourcemanager.monitor.capacity.preemption.natural_termination_factor

yarn.resourcemanager.monitor.capacity.preemption.monitoring_interval

yarn.nodemanager.linux-container-executor.nonsecure-mode.limit-users

yarn.nodemanager.runtime.linux.allowed-runtimes

yarn.nodemanager.runtime.linux.docker.allowed-container-networks

yarn.nodemanager.runtime.linux.docker.default-container-network

yarn.nodemanager.runtime.linux.docker.privileged-containers.allowed

yarn.nodemanager.runtime.linux.docker.privileged-containers.acl

yarn.nodemanager.runtime.linux.docker.capabilities

yarn.webapp.ui2.enable

yarn.timeline-service.http-cross-origin.enabled

yarn.resourcemanager.webapp.cross-origin.enabled

yarn.nodemanager.webapp.cross-origin.enabled

yarn.nodemanager.resource-plugins.gpu.docker-plugin

yarn.nodemanager.resource-plugins.gpu.docker-plugin.nvidiadocker-v1.endpoint

yarn.webapp.api-service.enable

yarn.service.framework.path

yarn.nodemanager.aux-services.timeline_collector.class

yarn.timeline-service.reader.webapp.address

yarn.timeline-service.reader.webapp.https.address

yarn.timeline-service.hbase-schema.prefix

yarn.timeline-service.hbase.configuration.file

yarn.timeline-service.hbase.coprocessor.jar.hdfs.location

yarn.resourcemanager.monitor.capacity.preemption.intra-queue-preemption.enabled

yarn.scheduler.capacity.ordering-policy.priority-utilization.underutilized-preemption.enabled

yarn.resourcemanager.display.per-user-apps

yarn.service.system-service.dir

yarn.timeline-service.generic-application-history.save-non-am-container-meta-info

hadoop.registry.dns.bind-address

hadoop.http.cross-origin.allowed-origins

yarn.nodemanager.resourcemanager.connect.wait.secs

// Log4j properties parameters (71)

hadoop.root.logger

hadoop.log.dir

hadoop.log.file

log4j.rootLogger

log4j.threshhold

log4j.appender.DRFA

log4j.appender.DRFA.File

log4j.appender.DRFA.DatePattern

log4j.appender.DRFA.layout

log4j.appender.DRFA.layout.ConversionPattern

log4j.appender.console

log4j.appender.console.target

log4j.appender.console.layout

log4j.appender.console.layout.ConversionPattern

hadoop.tasklog.taskid

hadoop.tasklog.iscleanup

hadoop.tasklog.noKeepSplits

hadoop.tasklog.totalLogFileSize

hadoop.tasklog.purgeLogSplits

hadoop.tasklog.logsRetainHours

log4j.appender.TLA

log4j.appender.TLA.taskId

log4j.appender.TLA.isCleanup

log4j.appender.TLA.totalLogFileSize

log4j.appender.TLA.layout

log4j.appender.TLA.layout.ConversionPattern

hadoop.security.logger

hadoop.security.log.maxfilesize

hadoop.security.log.maxbackupindex

log4j.category.SecurityLogger

hadoop.security.log.file

log4j.appender.DRFAS

log4j.appender.DRFAS.File

log4j.appender.DRFAS.layout

log4j.appender.DRFAS.layout.ConversionPattern

log4j.appender.DRFAS.DatePattern

log4j.appender.RFAS

log4j.appender.RFAS.File

log4j.appender.RFAS.layout

log4j.appender.RFAS.layout.ConversionPattern

log4j.appender.RFAS.MaxFileSize

log4j.appender.RFAS.MaxBackupIndex

hdfs.audit.logger

log4j.logger.org.apache.hadoop.hdfs.server.namenode.FSNamesystem.audit

log4j.additivity.org.apache.hadoop.hdfs.server.namenode.FSNamesystem.audit

log4j.appender.DRFAAUDIT

log4j.appender.DRFAAUDIT.File

log4j.appender.DRFAAUDIT.layout

log4j.appender.DRFAAUDIT.layout.ConversionPattern

log4j.appender.DRFAAUDIT.DatePattern

mapred.audit.logger

log4j.logger.org.apache.hadoop.mapred.AuditLogger

log4j.additivity.org.apache.hadoop.mapred.AuditLogger

log4j.appender.MRAUDIT

log4j.appender.MRAUDIT.File

log4j.appender.MRAUDIT.layout

log4j.appender.MRAUDIT.layout.ConversionPattern

log4j.appender.MRAUDIT.DatePattern

log4j.appender.RFA

log4j.appender.RFA.File

log4j.appender.RFA.MaxFileSize

log4j.appender.RFA.MaxBackupIndex

log4j.appender.RFA.layout

log4j.appender.RFA.layout.ConversionPattern

hadoop.metrics.log.level

log4j.logger.org.apache.hadoop.metrics2

log4j.logger.org.jets3t.service.impl.rest.httpclient.RestS3Service

log4j.appender.NullAppender

log4j.appender.EventCounter

log4j.logger.org.apache.hadoop.conf.Configuration.deprecation

log4j.logger.org.apache.commons.beanutils

// New MapRed parameters from mapred-site.xml

mapreduce.task.io.sort.mb

mapreduce.map.sort.spill.percent

mapreduce.task.io.sort.factor

mapreduce.cluster.administrators

mapreduce.reduce.shuffle.parallelcopies

mapreduce.map.speculative

mapreduce.reduce.speculative

mapreduce.job.reduce.slowstart.completedmaps

mapreduce.job.counters.max

mapreduce.reduce.shuffle.merge.percent

mapreduce.reduce.shuffle.input.buffer.percent

mapreduce.output.fileoutputformat.compress.type

mapreduce.reduce.input.buffer.percent

mapreduce.map.output.compress

mapreduce.task.timeout

mapreduce.map.memory.mb

mapreduce.reduce.memory.mb

mapreduce.shuffle.port

mapreduce.jobhistory.intermediate-done-dir

mapreduce.jobhistory.done-dir

mapreduce.jobhistory.address

mapreduce.jobhistory.webapp.address

mapreduce.framework.name

yarn.app.mapreduce.am.staging-dir

yarn.app.mapreduce.am.resource.mb

yarn.app.mapreduce.am.command-opts

yarn.app.mapreduce.am.admin-command-opts

yarn.app.mapreduce.am.log.level

mapreduce.admin.map.child.java.opts

mapreduce.admin.reduce.child.java.opts

mapreduce.application.classpath

mapreduce.am.max-attempts

mapreduce.map.java.opts

mapreduce.reduce.java.opts

mapreduce.map.log.level

mapreduce.reduce.log.level

mapreduce.admin.user.env

mapreduce.output.fileoutputformat.compress

mapreduce.jobhistory.http.policy

mapreduce.job.queuename

REPOSITORY_CONFIG_USERNAME

REPOSITORY_CONFIG_PASSWORD

REPOSITORY_CONFIG_USER_PASSWORD

REPOSITORY_TYPE

POLICY_DOWNLOAD_AUTH_USERS

REPOSITORY_CONFIG_BASE_URL

REPOSITORY_CONFIG_COMMON_NAME_FOR_CERTIFICATE

REPOSITORY_CONFIG_POLICY_MGR_SSL_CERTIFICATE

content.property-file-name

xasecure.audit.destination.db.jdbc.url

REPOSITORY_CONFIG_USERNAME

REPOSITORY_CONFIG_PASSWORD

REPOSITORY_CONFIG_USER_PASSWORD

REPOSITORY_TYPE

POLICY_DOWNLOAD_AUTH_USERS

REPOSITORY_CONFIG_BASE_URL

REPOSITORY_CONFIG_COMMON_NAME_FOR_CERTIFICATE

REPOSITORY_CONFIG_POLICY_MGR_SSL_CERTIFICATE

HIVE

hive.execution.engine // mr/tez/spark

hive.exec.parallel

hive.exec.parallel.thread.number

hive.fetch.task.conversion // query result fetching

hive.exec.mode.local.auto // auto-local mode

// Metastore Configuration

hive.metastore.uris // remote metastore URIs

javax.jdo.option.ConnectionURL// Embedded metastore JDBC URL

javax.jdo.option.ConnectionDriverName

hive.metastore.warehouse.dir

hive.metastore.schema.verification

hive.metastore.thrift.port

hive.metastore.sasl.enabled // Metastore security

// Security & Authorization

hive.security.authorization.enabled

hive.security.authorization.manager // SQLStd/Ranger

hive.server2.authentication // KERBEROS/LDAP/etc

hive.server2.xsrf.filter.enabled

hive.server2.enable.doAs // impersonation

hive.users.in.admin.role

hive.security.authorization.ranger.url // Ranger integration

// Transactions & Concurrency

hive.support.concurrency // enable concurrency

hive.txn.manager // DbTxnManager

hive.compactor.worker.threads

hive.lock.numretries

hive.lock.sleep.between.retries

// Query Optimization

hive.auto.convert.join

hive.optimize.bucketmapjoin

hive.cbo.enable // Cost-based optimization

hive.vectorized.execution.enabled

hive.optimize.ppd // predicate pushdown

hive.optimize.skewjoin

hive.merge.mapfiles // small file merging

// Storage & Serialization

hive.default.fileformat // ORC/Parquet/Text

hive.exec.compress.output

hive.exec.compress.intermediate

hive.orc.compute.splits.num.threads

hive.parquet.compression

// Tez/Spark Engine Configuration

hive.tez.container.size // Tez container sizing

hive.tez.java.opts

hive.execution.spark.client.timeout

hive.spark.client.server.connect.timeout

// LLAP Configuration

hive.llap.io.enabled

hive.llap.daemon.service.hosts

// Dynamic Partitioning

hive.exec.dynamic.partition.mode

hive.exec.max.dynamic.partitions

hive.exec.max.dynamic.partitions.pernode

// Statistics & Metadata

hive.stats.autogather

hive.stats.fetch.column.stats

// HDFS Integration

hive.exec.stagingdir // temp directory

hive.blobstore.use.blobstore.as.scratchdir // S3/Cloud integration

// Server Configuration

hive.server2.thrift.port

hive.server2.idle.operation.timeout

hive.server2.thrift.max.worker.threads

// Legacy & Compatibility

hive.mapred.mode // strict/nonstrict

hive.support.sql11.reserved.keywords

// Global audit parameters

xasecure.audit.is.enabled

// HDFS audit parameters :cite[2]:cite[3]

xasecure.audit.hdfs.is.enabled

xasecure.audit.hdfs.is.async

xasecure.audit.hdfs.async.max.queue.size

xasecure.audit.hdfs.async.max.flush.interval.ms

xasecure.audit.hdfs.config.encoding

xasecure.audit.hdfs.config.destination.directory

xasecure.audit.hdfs.config.destination.file

xasecure.audit.hdfs.config.destination.flush.interval.seconds

xasecure.audit.hdfs.config.destination.rollover.interval.seconds

xasecure.audit.hdfs.config.destination.open.retry.interval.seconds

xasecure.audit.hdfs.config.local.buffer.directory

xasecure.audit.hdfs.config.local.buffer.file

xasecure.audit.hdfs.config.local.buffer.file.buffer.size.bytes

xasecure.audit.hdfs.config.local.buffer.flush.interval.seconds

xasecure.audit.hdfs.config.local.buffer.rollover.interval.seconds

xasecure.audit.hdfs.config.local.archive.directory

xasecure.audit.hdfs.config.local.archive.max.file.count

// Log4j audit parameters

xasecure.audit.log4j.is.enabled

xasecure.audit.log4j.is.async

xasecure.audit.log4j.async.max.queue.size

xasecure.audit.log4j.async.max.flush.interval.ms

// Kafka audit parameters :cite[3]

xasecure.audit.kafka.is.enabled

xasecure.audit.kafka.async.max.queue.size

xasecure.audit.kafka.async.max.flush.interval.ms

xasecure.audit.kafka.broker_list

xasecure.audit.kafka.topic_name

// Solr audit parameters :cite[1]:cite[3]:cite[6]

xasecure.audit.solr.is.enabled

xasecure.audit.solr.async.max.queue.size

xasecure.audit.solr.async.max.flush.interval.ms

xasecure.audit.solr.solr_url

// Ranger security core parameters

ranger.plugin.hive.service.name // Ranger service name (e.g., hivedev)

ranger.plugin.hive.policy.source.impl // Policy retrieval class

ranger.plugin.hive.policy.rest.url // URL to Ranger Admin (critical for policy sync)

ranger.plugin.hive.policy.rest.ssl.config.file // SSL config path

ranger.plugin.hive.policy.pollIntervalMs // Policy refresh interval (default: 30s)

ranger.plugin.hive.policy.cache.dir // Policy cache directory

// Policy synchronization controls

xasecure.hive.update.xapolicies.on.grant.revoke // Sync Ranger policies on GRANT/REVOKE :cite[5]

xasecure.hive.uri.permission.coarse.check // Skip recursive URI checks (optimization)

// Connection tuning

ranger.plugin.hive.policy.rest.client.connection.timeoutMs // REST client timeout

ranger.plugin.hive.policy.rest.client.read.timeoutMs // REST read timeout

// Ranger audit parameters (from previous integration)

xasecure.audit.is.enabled

xasecure.audit.solr.is.enabled

// SSL/TLS Configuration (ranger-hive-policymgr-ssl.xml)

xasecure.policymgr.clientssl.keystore // Keystore file path

xasecure.policymgr.clientssl.truststore // Truststore file path

xasecure.policymgr.clientssl.keystore.credential.file // Keystore credentials

xasecure.policymgr.clientssl.truststore.credential.file // Truststore credentials

// Beeline Log4j2 Configuration Parameters

status

name

packages

property.hive.log.level

property.hive.root.logger

appenders

appender.console.type

appender.console.name

appender.console.target

appender.console.layout.type

appender.console.layout.pattern

loggers

logger.HiveConnection.name

logger.HiveConnection.level

logger.HiveJDBC.name

logger.HiveJDBC.level

rootLogger.level

rootLogger.appenderRefs

rootLogger.appenderRef.root.ref

// Log4j2 Configuration Parameters

status

name

packages

property.hive.log.level

property.hive.root.logger

property.hive.query.id

property.hive.log.dir

property.hive.log.file

appenders

appender.console.type

appender.console.name

appender.console.target

appender.console.layout.type

appender.console.layout.pattern

appender.FA.type

appender.FA.name

appender.FA.fileName

appender.FA.layout.type

appender.FA.layout.pattern

loggers

logger.NIOServerCnxn.name

logger.NIOServerCnxn.level

logger.ClientCnxnSocketNIO.name

logger.ClientCnxnSocketNIO.level

logger.DataNucleus.name

logger.DataNucleus.level

logger.Datastore.name

logger.Datastore.level

logger.JPOX.name

logger.JPOX.level

rootLogger.level

rootLogger.appenderRefs

rootLogger.appenderRef.root.ref

// hive-log4j2.properties parameters

name

property.hive.log.level

property.hive.root.logger

property.hive.log.dir

property.hive.log.file

property.hive.test.console.log.level

appender.console.type

appender.console.name

appender.console.target

appender.console.layout.type

appender.console.layout.pattern

appender.DRFA.type

appender.DRFA.name

appender.DRFA.fileName

appender.DRFA.filePattern

appender.DRFA.layout.type

appender.DRFA.layout.pattern

appender.DRFA.policies.type

appender.DRFA.policies.time.type

appender.DRFA.policies.time.interval

appender.DRFA.policies.time.modulate

appender.DRFA.strategy.type

appender.DRFA.strategy.max

logger.HadoopIPC.name

logger.HadoopIPC.level

logger.HadoopSecurity.name

logger.HadoopSecurity.level

logger.Hdfs.name

logger.Hdfs.level

logger.HdfsServer.name

logger.HdfsServer.level

logger.HadoopMetrics2.name

logger.HadoopMetrics2.level

logger.Mortbay.name

logger.Mortbay.level

logger.Yarn.name

logger.Yarn.level

logger.YarnServer.name

logger.YarnServer.level

logger.Tez.name

logger.Tez.level

logger.HadoopConf.name

logger.HadoopConf.level

logger.Zookeeper.name

logger.Zookeeper.level

logger.ServerCnxn.name

logger.ServerCnxn.level

logger.NIOServerCnxn.name

logger.NIOServerCnxn.level

logger.ClientCnxn.name

logger.ClientCnxn.level

logger.ClientCnxnSocket.name

logger.ClientCnxnSocket.level

logger.ClientCnxnSocketNIO.name

logger.ClientCnxnSocketNIO.level

logger.DataNucleus.name

logger.DataNucleus.level

logger.Datastore.name

logger.Datastore.level

logger.JPOX.name

logger.JPOX.level

logger.Operator.name

logger.Operator.level

logger.Serde2Lazy.name

logger.Serde2Lazy.level

logger.ObjectStore.name

logger.ObjectStore.level

logger.CalcitePlanner.name

logger.CalcitePlanner.level

logger.CBORuleLogger.name

logger.CBORuleLogger.level

logger.CBORuleLogger.filter.marker.type

logger.CBORuleLogger.filter.marker.marker

logger.CBORuleLogger.filter.marker.onMatch

logger.CBORuleLogger.filter.marker.onMismatch

logger.AmazonAws.name

logger.AmazonAws.level

logger.ApacheHttp.name

logger.ApacheHttp.level

logger.Thrift.name

logger.Thrift.level

logger.Jetty.name

logger.Jetty.level

logger.BlockStateChange.name

logger.BlockStateChange.level

rootLogger.level

rootLogger.appenderRefs

rootLogger.appenderRef.root.ref

rootLogger.appenderRef.console.ref

rootLogger.appenderRef.console.level

logger.swo.name

logger.swo.level

// Parquet Logging Configuration (from parquet-logging.properties)

org.apache.parquet.handlers

.level

java.util.logging.ConsoleHandler.level

java.util.logging.ConsoleHandler.formatter

java.util.logging.SimpleFormatter.format

java.util.logging.FileHandler.level

java.util.logging.FileHandler.pattern

java.util.logging.FileHandler.limit

java.util.logging.FileHandler.count

java.util.logging.FileHandler.formatter

// Configuration Parameters from llap-cli-log4j2.properties

status

name

packages

property.hive.log.level

property.hive.root.logger

property.hive.log.dir

property.hive.log.file

property.hive.llapstatus.consolelogger.level

appenders

appender.console.type

appender.console.name

appender.console.target

appender.console.layout.type

appender.console.layout.pattern

appender.llapstatusconsole.type

appender.llapstatusconsole.name

appender.llapstatusconsole.target

appender.llapstatusconsole.layout.type

appender.llapstatusconsole.layout.pattern

appender.DRFA.type

appender.DRFA.name

appender.DRFA.fileName

appender.DRFA.filePattern

appender.DRFA.layout.type

appender.DRFA.layout.pattern

appender.DRFA.policies.type

appender.DRFA.policies.time.type

appender.DRFA.policies.time.interval

appender.DRFA.policies.time.modulate

appender.DRFA.strategy.type

appender.DRFA.strategy.max

appender.DRFA.policies.fsize.type

appender.DRFA.policies.fsize.size

loggers

logger.ZooKeeper.name

logger.ZooKeeper.level

logger.DataNucleus.name

logger.DataNucleus.level

logger.Datastore.name

logger.Datastore.level

logger.JPOX.name

logger.JPOX.level

logger.HadoopConf.name

logger.HadoopConf.level

logger.LlapStatusServiceDriverConsole.name

logger.LlapStatusServiceDriverConsole.additivity

logger.LlapStatusServiceDriverConsole.level

rootLogger.level

rootLogger.appenderRefs

rootLogger.appenderRef.root.ref

rootLogger.appenderRef.DRFA.ref

logger.LlapStatusServiceDriverConsole.appenderRefs

logger.LlapStatusServiceDriverConsole.appenderRef.llapstatusconsole.ref

logger.LlapStatusServiceDriverConsole.appenderRef.DRFA.ref

// Extracted from llap-daemon-log4j.properties

llap.daemon.log.level   // Root log level (INFO)

llap.daemon.root.logger   // Default appender (console)

llap.daemon.log.dir   // Log directory (.)

llap.daemon.log.file   // Main log filename (llapdaemon.log)

llap.daemon.historylog.file // History log filename

llap.daemon.log.maxfilesize // Max log file size (256MB)

{ "llap.daemon.log.maxbackupindex", "llap-daemon-log4j.properties" } ,

Kafka

bootstrap.servers

bootstrap.servers

zookeeper.connect

client.id

client.id

listeners

advertised.listeners

// Producer Configurations

acks

retries

batch.size

linger.ms

compression.type

max.request.size

enable.idempotence

buffer.memory

max.block.ms

delivery.timeout.ms

request.timeout.ms

max.in.flight.requests.per.connection

metadata.max.age.ms

send.buffer.bytes

transactional.id

// Consumer Configurations

group.id

auto.offset.reset

enable.auto.commit

max.poll.records

fetch.min.bytes

fetch.max.bytes

heartbeat.interval.ms

max.partition.fetch.bytes

receive.buffer.bytes

partition.assignment.strategy

fetch.max.wait.ms

max.poll.interval.ms

// Broker Configurations

log.dirs

num.partitions

default.replication.factor

offsets.topic.replication.factor

auto.create.topics.enable

log.retention.ms

log.segment.bytes

controlled.shutdown.enable

unclean.leader.election.enable

socket.send.buffer.bytes

socket.receive.buffer.bytes

num.recovery.threads.per.data.dir

log.flush.interval.messages

log.flush.interval.ms

message.max.bytes

auto.leader.rebalance.enable

// Security (SSL/SASL)

security.protocol

security.protocol

security.protocol

ssl.keystore.location

ssl.keystore.location

ssl.keystore.location

ssl.truststore.location

ssl.truststore.location

ssl.truststore.location

ssl.keystore.password

ssl.keystore.password

ssl.keystore.password

ssl.truststore.password

ssl.truststore.password

ssl.truststore.password

ssl.key.password

ssl.key.password

ssl.key.password

ssl.endpoint.identification.algorithm

ssl.endpoint.identification.algorithm

ssl.endpoint.identification.algorithm

sasl.mechanism

sasl.mechanism

sasl.mechanism

sasl.jaas.config

sasl.jaas.config

sasl.jaas.config

// Performance Tuning

log.retention.hours

log.retention.bytes

num.io.threads

num.network.threads

log.retention.ms

log.segment.bytes

xasecure.audit.is.enabled

xasecure.audit.hdfs.is.enabled

xasecure.audit.hdfs.is.async

xasecure.audit.hdfs.async.max.queue.size

xasecure.audit.hdfs.async.max.flush.interval.ms

xasecure.audit.hdfs.config.encoding

xasecure.audit.hdfs.config.destination.directory

xasecure.audit.hdfs.config.destination.file

xasecure.audit.hdfs.config.destination.flush.interval.seconds

xasecure.audit.hdfs.config.destination.rollover.interval.seconds

xasecure.audit.hdfs.config.destination.open.retry.interval.seconds

xasecure.audit.hdfs.config.local.buffer.directory

xasecure.audit.hdfs.config.local.buffer.file

xasecure.audit.hdfs.config.local.buffer.file.buffer.size.bytes

xasecure.audit.hdfs.config.local.buffer.flush.interval.seconds

xasecure.audit.hdfs.config.local.buffer.rollover.interval.seconds

xasecure.audit.hdfs.config.local.archive.directory

xasecure.audit.hdfs.config.local.archive.max.file.count

xasecure.audit.log4j.is.enabled

xasecure.audit.log4j.is.async

xasecure.audit.log4j.async.max.queue.size

xasecure.audit.log4j.async.max.flush.interval.ms

xasecure.audit.kafka.is.enabled

xasecure.audit.kafka.async.max.queue.size

xasecure.audit.kafka.async.max.flush.interval.ms

xasecure.audit.kafka.broker_list

xasecure.audit.kafka.topic_name

xasecure.audit.solr.is.enabled

xasecure.audit.solr.async.max.queue.size

xasecure.audit.solr.async.max.flush.interval.ms

xasecure.audit.solr.solr_url

// New parameters from ranger-kafka-security.xml

ranger.plugin.kafka.service.name

ranger.plugin.kafka.policy.source.impl

ranger.plugin.kafka.policy.rest.url

ranger.plugin.kafka.policy.rest.ssl.config.file

ranger.plugin.kafka.policy.pollIntervalMs

ranger.plugin.kafka.policy.cache.dir

ranger.plugin.kafka.policy.rest.client.connection.timeoutMs

ranger.plugin.kafka.policy.rest.client.read.timeoutMs

// New parameters from ranger-kafka-policymgr-ssl.xml

xasecure.policymgr.clientssl.keystore

xasecure.policymgr.clientssl.keystore.password

xasecure.policymgr.clientssl.truststore

xasecure.policymgr.clientssl.truststore.password

xasecure.policymgr.clientssl.keystore.credential.file

xasecure.policymgr.clientssl.truststore.credential.file

// New Settings from log4j.properties

log4j.rootLogger

log4j.appender.stdout

log4j.appender.stdout.layout

log4j.appender.stdout.layout.ConversionPattern

log4j.appender.kafkaAppender

log4j.appender.kafkaAppender.DatePattern

log4j.appender.kafkaAppender.File

log4j.appender.kafkaAppender.layout

log4j.appender.kafkaAppender.layout.ConversionPattern

log4j.appender.stateChangeAppender

log4j.appender.stateChangeAppender.DatePattern

log4j.appender.stateChangeAppender.File

log4j.appender.stateChangeAppender.layout

log4j.appender.stateChangeAppender.layout.ConversionPattern

log4j.appender.requestAppender

log4j.appender.requestAppender.DatePattern

log4j.appender.requestAppender.File

log4j.appender.requestAppender.layout

log4j.appender.requestAppender.layout.ConversionPattern

log4j.appender.cleanerAppender

log4j.appender.cleanerAppender.DatePattern

log4j.appender.cleanerAppender.File

log4j.appender.cleanerAppender.layout

log4j.appender.cleanerAppender.layout.ConversionPattern

log4j.appender.controllerAppender

log4j.appender.controllerAppender.DatePattern

log4j.appender.controllerAppender.File

log4j.appender.controllerAppender.layout

log4j.appender.controllerAppender.layout.ConversionPattern

log4j.appender.authorizerAppender

log4j.appender.authorizerAppender.DatePattern

log4j.appender.authorizerAppender.File

log4j.appender.authorizerAppender.layout

log4j.appender.authorizerAppender.layout.ConversionPattern

log4j.logger.org.apache.zookeeper

log4j.logger.kafka

log4j.logger.org.apache.kafka

log4j.logger.kafka.request.logger

log4j.additivity.kafka.request.logger

log4j.logger.kafka.network.RequestChannel$

log4j.additivity.kafka.network.RequestChannel$

log4j.logger.org.apache.kafka.controller

log4j.additivity.org.apache.kafka.controller

log4j.logger.kafka.controller

log4j.additivity.kafka.controller

log4j.logger.kafka.log.LogCleaner

log4j.additivity.kafka.log.LogCleaner

log4j.logger.state.change.logger

log4j.additivity.state.change.logger

log4j.logger.kafka.authorizer.logger

log4j.additivity.kafka.authorizer.logger

Livy

livy.server.port

livy.server.host

livy.server.session.timeout

livy.server.session-state-retain.sec

livy.server.session.factory

livy.server.recovery.mode

livy.server.recovery.state-store

livy.server.recovery.state-store.url

/* Security & Authentication */

livy.server.auth.type

livy.keystore

livy.keystore.password

livy.truststore

livy.truststore.password

livy.server.auth.kerberos.principal

livy.server.auth.kerberos.keytab

livy.server.auth.ldap.url

livy.server.auth.ldap.baseDN

livy.server.auth.ldap.userDNPattern

livy.server.auth.ldap.groupDNPattern

livy.server.auth.jwt.public-key

livy.server.auth.jwt.issuer

livy.server.auth.jwt.audience

livy.server.impersonation.enabled

livy.server.impersonation.allowed.users

livy.server.access-control.enabled

livy.server.access-control.users

livy.server.access-control.groups

livy.server.launch.kerberos.principal

livy.server.launch.kerberos.keytab

livy.server.superusers

/* Spark Configuration */

livy.spark.master

livy.spark.deploy-mode

livy.spark.home

livy.spark.submit.deployMode

livy.spark.submit.proxyUser

livy.spark.driver.cores

livy.spark.driver.memory

livy.spark.executor.cores

livy.spark.executor.memory

livy.spark.dynamicAllocation.enabled

livy.spark.dynamicAllocation.minExecutors

livy.spark.dynamicAllocation.maxExecutors

livy.spark.dynamicAllocation.initialExecutors

/* Resource Management */

livy.spark.yarn.queue

livy.spark.yarn.archives

livy.spark.yarn.dist.files

livy.spark.yarn.maxAppAttempts

livy.spark.kubernetes.namespace

livy.spark.kubernetes.container.image

livy.spark.kubernetes.authenticate.driver.serviceAccountName

livy.spark.kubernetes.driver.podTemplateFile

livy.spark.kubernetes.executor.podTemplateFile

/* Session Management */

livy.file.local-dir

livy.file.local-dir-whitelist

livy.server.session.max_creation_time

livy.server.session.heartbeat.timeout

livy.server.session.max_sessions_per_user

livy.rsc.server-address

livy.rsc.jvm.opts

livy.rsc.sparkr.package

livy.rsc.livy-jars

/* Interactive & Batch Processing */

livy.repl.enableHiveContext

livy.batch.retained

/* UI & Monitoring */

livy.ui.enabled

livy.ui.session-list.max

livy.metrics.enabled

livy.metrics.reporters

livy.metrics.jmx.domain

livy.server.request-log.enabled

livy.server.access-log.enabled

/* Network & Security Protocols */

livy.server.csrf-protection.enabled

livy.server.cors.enabled

livy.server.cors.allowed-origins

livy.server.cors.allowed-methods

livy.server.cors.allowed-headers

livy.server.cors.exposed-headers

/* YARN/Kubernetes Specific */

livy.yarn.app-name

livy.yarn.config-file

livy.yarn.jar

livy.yarn.poll-interval

livy.kubernetes.truststore.secret

livy.kubernetes.truststore.password.secret

livy.kubernetes.keystore.secret

livy.kubernetes.keystore.password.secret

log4j.rootCategory

log4j.appender.console

log4j.appender.console.target

log4j.appender.console.layout

log4j.appender.console.layout.ConversionPattern

log4j.logger.org.eclipse.jetty

livy.client.http.connection.timeout

livy.client.http.connection.socket.timeout

livy.client.http.content.compress.enable

livy.client.http.connection.idle.timeout

livy.client.http.job.initial-poll-interval

livy.client.http.job.max-poll-interval

livy.rsc.client.auth.id

livy.rsc.client.auth.secret

livy.rsc.client.shutdown-timeout

livy.rsc.driver-class

livy.rsc.session.kind

livy.rsc.jars

livy.rsc.sparkr.package

livy.rsc.pyspark.archives

livy.rsc.launcher.address

livy.rsc.launcher.port.range

livy.rsc.server.idle-timeout

livy.rsc.proxy-user

livy.rsc.rpc.server.address

livy.rsc.server.connect.timeout

livy.rsc.channel.log.level

livy.rsc.rpc.sasl.mechanisms

livy.rsc.rpc.sasl.qop

livy.rsc.job-cancel.trigger-interval

livy.rsc.job-cancel.timeout

livy.rsc.retained-statements

/* Spark Blacklist Configuration */

spark.master

spark.submit.deployMode

spark.yarn.jar

spark.yarn.jars

spark.yarn.archive

livy.rsc.server.idle-timeout

pig

pig.exec.mapPartAgg

pig.skewedjoin.reduce.memusage

pig.cachedbag.memusage

pig.maxCombinedSplitSize

pig.optimizer.multiquery

pig.tmpFileCompression

pig.exec.nocombiner

pig.user.cache.location

pig.exec.reducers.bytes.per.reducer

pig.exec.reducers.max

pig.exec.mapPartAgg.minFraction

pig.join.optimized

pig.skewedjoin.minimizeDataSkew

pig.auto.local.enabled

pig.auto.local.input.maxbytes

pig.script.allow.udf.import

pig.logfile

pig.stats.logging.level

pig.job.priority

pig.script.udf.import.path

// New parameters covering Pig's full configuration capabilities

pig.default.parallel

pig.splitCombination

pig.exec.mapPartition

pig.broadcast.join.threshold

pig.join.tuples.batch.size

pig.mergeCombinedSplitSize

pig.output.lzo.enabled

pig.optimizer.list

pig.jar

pig.udf.profiles

pig.task.agg.memusage

pig.spill.size.threshold

pig.optimizer.rules.disabled

pig.hadoop.version

pig.execution.mode

pig.schema.tuple.enable

pig.datetime.default.tz

pig.optimizer.uniqueKey

pig.stats.reliability

pig.optimizer.disable.splitcombiner

pig.udf.import.list

pig.jobcontrol.statement.retry.max

pig.jobcontrol.statement.retry.interval

pig.output.compression.enabled

pig.output.compression.codec

pig.relocation.jars

pig.script.auto.progress

pig.tez.jvm.args

pig.tez.container.reuse

Presto

node.id

node.environment

node.data-dir

node.launcher-log-file

node.server-log-file

node.presto-version

node.allow-version-mismatch

// config.properties - Coordinator & Discovery

coordinator

discovery-server.enabled

discovery.uri

// HTTP Server

http-server.http.port

http-server.https.port

http-server.https.enabled

http-server.https.keystore.path

http-server.https.keystore.key

http-server.https.truststore.path

http-server.log.path

http-server.log.enabled

http-server.authentication.type

http-server.process-forwarded

// Query Management

query.max-memory

query.max-memory-per-node

query.max-total-memory-per-node

query.max-execution-time

query.max-run-time

query.client.timeout

query.min-expire-age

// Memory Management

memory.heap-headroom-per-node

memory.max-revokable-memory-per-node

// Task & Scheduler

task.concurrency

task.http-response-threads

task.info-update-interval

scheduler.http-client.max-connections

scheduler.http-client.max-connections-per-server

scheduler.include-coordinator

node-scheduler.network-topology

// Exchange

exchange.client-threads

exchange.max-buffer-size

// Optimizer

optimizer.dictionary-aggregation

optimizer.optimize-hash-generation

redistribute-writes

// JMX

jmx.base-name

// Security

internal-communication.https.required

// Experimental/Spilling

experimental.spiller-spill-path

spill-enabled

// Resource Management

resource-manager

resource-group-manager

// Additional parameters

join-distribution-type

task.writer-count

http-server.https.sni-host-check

query.max-stage-count

spark

spark.master

spark.app.name

spark.executor.memory

spark.driver.memory

spark.serializer

spark.sql.shuffle.partitions

spark.default.parallelism

spark.executor.cores

spark.shuffle.service.enabled

spark.dynamicAllocation.enabled

spark.eventLog.enabled

spark.yarn.queue

spark.submit.deployMode

spark.network.timeout

spark.ui.port

spark.driver.maxResultSize

spark.executor.instances

spark.sql.autoBroadcastJoinThreshold

spark.memory.fraction

spark.locality.wait

// Additional comprehensive configurations

spark.driver.cores

spark.memory.offHeap.enabled

spark.memory.offHeap.size

spark.executor.memoryOverhead

spark.driver.memoryOverhead

spark.shuffle.compress

spark.shuffle.spill.compress

spark.io.compression.codec

spark.shuffle.file.buffer

spark.reducer.maxSizeInFlight

spark.dynamicAllocation.minExecutors

spark.dynamicAllocation.maxExecutors

spark.dynamicAllocation.initialExecutors

spark.dynamicAllocation.executorIdleTimeout

spark.sql.adaptive.enabled

spark.sql.files.maxPartitionBytes

spark.sql.sources.partitionOverwriteMode

spark.sql.cbo.enabled

spark.streaming.backpressure.enabled

spark.streaming.kafka.maxRatePerPartition

spark.ui.enabled

spark.eventLog.dir

spark.eventLog.compress

spark.authenticate

spark.ssl.enabled

spark.yarn.am.memory

spark.yarn.executor.memoryOverhead

spark.yarn.driver.memoryOverhead

spark.rpc.message.maxSize

spark.blockManager.port

spark.scheduler.mode

spark.checkpoint.compress

spark.pyspark.python

rootLogger.level

rootLogger.appenderRef.stdout.ref

appender.console.type

appender.console.name

appender.console.target

appender.console.layout.type

appender.console.layout.pattern

logger.repl.name

logger.repl.level

logger.thriftserver.name

logger.thriftserver.level

logger.jetty1.name

logger.jetty1.level

logger.jetty2.name

logger.jetty2.level

logger.replexprTyper.name

logger.replexprTyper.level

logger.replSparkILoopInterpreter.name

logger.replSparkILoopInterpreter.level

logger.parquet1.name

logger.parquet1.level

logger.parquet2.name

logger.parquet2.level

logger.RetryingHMSHandler.name

logger.RetryingHMSHandler.level

logger.FunctionRegistry.name

logger.FunctionRegistry.level

// Metrics.properties configurations (144 new entries)

// Class properties for sinks

*.sink.console.class

master.sink.console.class

worker.sink.console.class

executor.sink.console.class

driver.sink.console.class

applications.sink.console.class

*.sink.csv.class

master.sink.csv.class

worker.sink.csv.class

executor.sink.csv.class

driver.sink.csv.class

applications.sink.csv.class

*.sink.ganglia.class

master.sink.ganglia.class

worker.sink.ganglia.class

executor.sink.ganglia.class

driver.sink.ganglia.class

applications.sink.ganglia.class

*.sink.jmx.class

master.sink.jmx.class

worker.sink.jmx.class

executor.sink.jmx.class

driver.sink.jmx.class

applications.sink.jmx.class

*.sink.graphite.class

master.sink.graphite.class

worker.sink.graphite.class

executor.sink.graphite.class

driver.sink.graphite.class

applications.sink.graphite.class

// Console sink options

*.sink.console.period

master.sink.console.period

worker.sink.console.period

executor.sink.console.period

driver.sink.console.period

applications.sink.console.period

*.sink.console.unit

master.sink.console.unit

worker.sink.console.unit

executor.sink.console.unit

driver.sink.console.unit

applications.sink.console.unit

// CSV sink options

*.sink.csv.period

master.sink.csv.period

worker.sink.csv.period

executor.sink.csv.period

driver.sink.csv.period

applications.sink.csv.period

*.sink.csv.unit

master.sink.csv.unit

worker.sink.csv.unit

executor.sink.csv.unit

driver.sink.csv.unit

applications.sink.csv.unit

*.sink.csv.directory

master.sink.csv.directory

worker.sink.csv.directory

executor.sink.csv.directory

driver.sink.csv.directory

applications.sink.csv.directory

// Ganglia sink options

*.sink.ganglia.host

master.sink.ganglia.host

worker.sink.ganglia.host

executor.sink.ganglia.host

driver.sink.ganglia.host

applications.sink.ganglia.host

*.sink.ganglia.port

master.sink.ganglia.port

worker.sink.ganglia.port

executor.sink.ganglia.port

driver.sink.ganglia.port

applications.sink.ganglia.port

*.sink.ganglia.period

master.sink.ganglia.period

worker.sink.ganglia.period

executor.sink.ganglia.period

driver.sink.ganglia.period

applications.sink.ganglia.period

*.sink.ganglia.unit

master.sink.ganglia.unit

worker.sink.ganglia.unit

executor.sink.ganglia.unit

driver.sink.ganglia.unit

applications.sink.ganglia.unit

*.sink.ganglia.ttl

master.sink.ganglia.ttl

worker.sink.ganglia.ttl

executor.sink.ganglia.ttl

driver.sink.ganglia.ttl

applications.sink.ganglia.ttl

*.sink.ganglia.mode

master.sink.ganglia.mode

worker.sink.ganglia.mode

executor.sink.ganglia.mode

driver.sink.ganglia.mode

applications.sink.ganglia.mode

// Graphite sink options

*.sink.graphite.host

master.sink.graphite.host

worker.sink.graphite.host

executor.sink.graphite.host

driver.sink.graphite.host

applications.sink.graphite.host

*.sink.graphite.port

master.sink.graphite.port

worker.sink.graphite.port

executor.sink.graphite.port

driver.sink.graphite.port

applications.sink.graphite.port

*.sink.graphite.period

master.sink.graphite.period

worker.sink.graphite.period

executor.sink.graphite.period

driver.sink.graphite.period

applications.sink.graphite.period

*.sink.graphite.unit

master.sink.graphite.unit

worker.sink.graphite.unit

executor.sink.graphite.unit

driver.sink.graphite.unit

applications.sink.graphite.unit

*.sink.graphite.prefix

master.sink.graphite.prefix

worker.sink.graphite.prefix

executor.sink.graphite.prefix

driver.sink.graphite.prefix

applications.sink.graphite.prefix

// MetricsServlet options

*.sink.MetricsServlet.path

master.sink.MetricsServlet.path

worker.sink.MetricsServlet.path

executor.sink.MetricsServlet.path

driver.sink.MetricsServlet.path

applications.sink.MetricsServlet.path

*.sink.MetricsServlet.sample

master.sink.MetricsServlet.sample

worker.sink.MetricsServlet.sample

executor.sink.MetricsServlet.sample

driver.sink.MetricsServlet.sample

applications.sink.MetricsServlet.sample

// JVM source class

*.source.jvm.class

master.source.jvm.class

worker.source.jvm.class

executor.source.jvm.class

driver.source.jvm.class

applications.source.jvm.class

storm

storm.zookeeper.servers

storm.zookeeper.port

storm.zookeeper.root

storm.zookeeper.session.timeout

storm.zookeeper.connection.timeout

storm.local.dir

storm.cluster.mode

// Nimbus Configuration

nimbus.seeds

nimbus.host

nimbus.thrift.port

nimbus.task.launch.secs

nimbus.task.timeout.secs

nimbus.supervisor.timeout.secs

nimbus.code.sync.freq.secs

nimbus.blobstore.class

// Supervisor Configuration

supervisor.slots.ports

supervisor.worker.timeout.secs

supervisor.cpu.capacity

supervisor.memory.capacity.mb

supervisor.heartbeat.frequency.secs

supervisor.monitor.frequency.secs

supervisor.enable

supervisor.worker.port

// Worker Configuration

worker.childopts

worker.heap.memory.mb

worker.gc.childopts

worker.log.level.reset.interval.secs

worker.profiler.enabled

// Network and Messaging

storm.messaging.transport

storm.messaging.netty.buffer_size

storm.network.topography.plugin

storm.thrift.socket.timeout.ms

// UI and Logging

ui.port

ui.host

ui.http.x-frame-options

logviewer.port

logviewer.max.per.worker.logs.mb

storm.log4j2.conf.dir

// Security

storm.kerberos.principal

storm.kerberos.keytab

java.security.auth.login.config

supervisor.run.worker.as.user

// DRPC Configuration

drpc.servers

drpc.port

drpc.worker.threads

drpc.queue.size

// Resource Management

topology.priority

topology.scheduler.strategy

topology.component.resources.onheap.memory.mb

topology.component.resources.offheap.memory.mb

topology.component.cpu.pcore.percent

// Topology Execution

topology.workers

topology.acker.executors

topology.max.spout.pending

topology.message.timeout.secs

topology.debug

topology.tasks

topology.state.checkpoint.interval.ms

topology.enable.message.timeouts

// Fault Tolerance

topology.state.synchronization.timeout.secs

topology.max.task.parallelism

topology.worker.gc.ratio

// Serialization

topology.multilang.serializer

topology.skip.missing.kryo.registrations

topology.fall.back.on.java.serialization

// Metrics and Monitoring

topology.builtin.metrics.bucket.size.secs

topology.stats.sample.rate

topology.metrics.consumer.register

// Advanced Configuration

storm.blobstore.replication.factor

storm.health.check.timeout.ms

topology.auto-credentials

topology.enable.classloader

topology.testing.always.try.serialize

// Transactional Topologies

topology.transactional.id.seed

topology.state.provider

zeppelin

zeppelin.server.port

zeppelin.server.addr

zeppelin.server.context.path

zeppelin.ssl.enabled

zeppelin.ssl.keystore.path

zeppelin.ssl.truststore.path

// Notebook Management

zeppelin.notebook.storage

zeppelin.notebook.dir

zeppelin.notebook.git.remote.url

zeppelin.notebook.git.username

zeppelin.notebook.auto.commit

// Interpreter Configuration

zeppelin.interpreter.localRepo

zeppelin.interpreter.group

zeppelin.interpreter.connect.timeout

zeppelin.interpreter.isolation

zeppelin.interpreter.process.max_threads

// Resource Management

zeppelin.executor.memory

zeppelin.resource.pool.size

zeppelin.memory.allocator.max

// Backend Integration

zeppelin.spark.master

zeppelin.spark.executor.cores

zeppelin.flink.jobmanager.url

zeppelin.hive.hiveserver2.url

zeppelin.jdbc.drivers

// Security & Authentication

shiro.realm

shiro.ldap.contextFactory.url

shiro.ldap.userDnTemplate

shiro.activeDirectoryRealm.domain

shiro.oauth2.clientId

shiro.oauth2.callbackUrl

// High Availability & Clustering

zeppelin.ha.enabled

zeppelin.ha.zookeeper.quorum

zeppelin.cluster.addr

// REST API & Monitoring

zeppelin.server.rest.api.port

zeppelin.monitoring.enabled

// Logging & Diagnostics

zeppelin.log.dir

zeppelin.log.level

// Dependency Management

zeppelin.dep.additionalRemoteRepository

// User Interface

zeppelin.helium.registry

zeppelin.notebook.collaborative.mode

// Session Management

zeppelin.session.timeout

shiro.sessionTimeout

// External Systems Integration

zeppelin.config.fs.dir

zeppelin.credentials.file

// New Configuration Parameters (from log4j.properties)

log4j.rootLogger

log4j.appender.stdout

log4j.appender.stdout.layout

log4j.appender.stdout.layout.ConversionPattern

log4j.appender.dailyfile.DatePattern

log4j.appender.dailyfile.DEBUG

log4j.appender.dailyfile

log4j.appender.dailyfile.File

log4j.appender.dailyfile.layout

log4j.appender.dailyfile.layout.ConversionPattern

log4j.logger.org.apache.zeppelin.python

log4j.logger.org.apache.zeppelin.spark

// New entries from shiro.ini

// [users] section

shiro.user.user1

shiro.user.user2

shiro.user.user3

// [main] section

shiro.main.sessionManager

shiro.main.cookie

shiro.main.cookie.name

shiro.main.cookie.httpOnly

shiro.main.sessionManager.sessionIdCookie

shiro.main.securityManager.sessionManager

shiro.main.securityManager.sessionManager.globalSessionTimeout

shiro.main.shiro.loginUrl

// [roles] section

shiro.role.role1

shiro.role.role2

shiro.role.role3

shiro.role.admin

// [urls] section

shiro.url./api/version

shiro.url./api/cluster/address

shiro.url./api/interpreter/setting/restart/**

shiro.url./api/interpreter/**

shiro.url./api/notebook-repositories/**

shiro.url./api/configurations/**

shiro.url./api/credential/**

shiro.url./api/admin/**

shiro.url./**

zookeeper

clientPort

dataDir

tickTime

initLimit

syncLimit

maxClientCnxns

autopurge.snapRetainCount

autopurge.purgeInterval

minSessionTimeout

maxSessionTimeout

electionPort

leaderServes

server.id // Dynamic server entries

cnxTimeout

standaloneEnabled

reconfigEnabled

4lw.commands.whitelist

globalOutstandingLimit

preAllocSize

snapCount

// Security & Authentication

clientPortAddress

secureClientPort

ssl.keyStore.location

ssl.keyStore.password

ssl.trustStore.location

ssl.trustStore.password

ssl.hostnameVerification

authProvider.sasl

jaasLoginRenew

sasl.client.id

kerberos.removeHostFromPrincipal

kerberos.removeRealmFromPrincipal

ssl.clientAuth

zookeeper.superUser

// Quorum & Ensemble Management

quorum.enableSasl

quorum.auth.learnerRequireSasl

quorum.auth.serverRequireSasl

quorum.cnxTimeout

quorum.electionAlg

quorum.portUnification

// ACLs & Data Security

skipACL

aclProvider

// Performance & Advanced Tuning

jute.maxbuffer

commitProcessor.numWorkerThreads

fsync.warningthresholdms

forceSync

syncEnabled

connectTimeout

readTimeout

// Dynamic Configuration & Admin

dynamicConfigFile

admin.enableServer

admin.serverPort

admin.serverAddress

// Metrics & Monitoring

metricsProvider.className

// Network & Client Settings

clientCnxnSocket

client.secure

// Additional 4LW Controls

4lw.commands.enabled

// Advanced Throttling and NIO

zookeeper.request_throttler.shutdownTimeout

zookeeper.nio.numSelectorThreads

zookeeper.nio.numWorkerThreads

zookeeper.nio.directBufferBytes

// New Parameters from log4j.properties

log4j.rootLogger

log4j.appender.CONSOLE

log4j.appender.CONSOLE.Threshold

log4j.appender.CONSOLE.layout

log4j.appender.CONSOLE.layout.ConversionPattern

log4j.appender.ROLLINGFILE

log4j.appender.ROLLINGFILE.Threshold

log4j.appender.ROLLINGFILE.File

log4j.appender.ROLLINGFILE.MaxFileSize

log4j.appender.ROLLINGFILE.layout

log4j.appender.ROLLINGFILE.layout.ConversionPattern

log4j.appender.TRACEFILE

log4j.appender.TRACEFILE.Threshold

log4j.appender.TRACEFILE.File

log4j.appender.TRACEFILE.layout

log4j.appender.TRACEFILE.layout.ConversionPattern

Atlas

atlas.server.http.port

atlas.server.https.port

atlas.server.bind.address

atlas.server.admin.port

atlas.rest.address

atlas.server.data

atlas.server.ha.enabled

// Security & Authentication

atlas.enableTLS

atlas.ssl.keystore.file

atlas.ssl.keystore.password

atlas.ssl.truststore.file

atlas.ssl.truststore.password

atlas.authentication.method.ldap.url

atlas.authentication.method.ldap.userDNpattern

atlas.authentication.method.kerberos.keytab

atlas.authentication.method.oidc.issuer.url

atlas.authorization.simple.authz.policy.file

// Storage & Backend

atlas.graph.storage.backend

atlas.graph.storage.hbase.table

atlas.graph.storage.cassandra.keyspace

atlas.graph.index.search.backend

atlas.graph.index.search.solr.zookeeper-url

atlas.graph.index.search.elasticsearch.hosts

// Metadata & Governance

atlas.metadata.namespace

atlas.entity.audit.export

atlas.entity.audit.retention.days

atlas.glossary.import.file

atlas.tag.policy.file

// Notification & Messaging

atlas.notification.embedded

atlas.kafka.zookeeper.connect

atlas.notification.create.topics

atlas.notification.max.retries

// High Availability

atlas.server.ha.zookeeper.connect

atlas.server.ha.zookeeper.session.timeout

atlas.server.ha.id

// Performance & Monitoring

atlas.metrics.enabled

atlas.metrics.reporters

atlas.performance.cache.size

// Data Governance

atlas.data.quality.validator.class

atlas.lineage.audit.enabled

atlas.policy.evaluation.enabled

// UI Configuration

atlas.ui.default.namespace

atlas.ui.search.result.limit

// Advanced Features

atlas.titan.attribute.ids.enabled

atlas.fulltext.search.enabled

atlas.entity.relationships.enabled

solr

coreRootDirectory

// SolrCloud/ZooKeeper (solr.xml)

zkHost

zkClientTimeout

cloud.collection.configName

numShards

// Replication/Sharding (solr.xml)

shardHandlerFactory.socketTimeout

replication.factor

// Monitoring/Logging (solr.xml)

metrics.reporter.jmx

logging.watcher.threshold

// HTTP/Network Settings (solr.xml)

hostContext

http.maxConnections

// Legacy Parameters (solr.xml only)

transientCacheSize

// solr-log4j.properties parameters

solr.log

log4j.rootLogger

log4j.appender.CONSOLE

log4j.appender.CONSOLE.layout

log4j.appender.CONSOLE.layout.ConversionPattern

log4j.logger.org.apache.zookeeper

log4j.logger.org.apache.hadoop

log4j.logger.org.eclipse.jetty

log4j.logger.org.eclipse.jetty.server.Server

log4j.logger.org.eclipse.jetty.server.ServerConnector

log4j.logger.org.apache.solr.update.LoggingInfoStream

tez

tez.am.resource.memory.mb,

tez.task.resource.memory.mb,

tez.am.resource.cpu.vcores,

tez.task.resource.cpu.vcores,

tez.am.container.heap.memory-mb.ratio,

tez.am.container.java.opts,

tez.am.launch.cmd-opts,

// Queuing and Scheduling

tez.queue.name,

tez.am.node-blacklisting.enabled,

tez.am.node-blacklisting.ignore-threshold.node-percent,

// Execution Control

tez.am.container.reuse.enabled,

tez.am.container.reuse.rack-fallback.enabled,

tez.am.container.idle.release-timeout-min.millis,

tez.am.container.idle.release-timeout-max.millis,

// Shuffle and Sorting

tez.runtime.io.sort.mb,

tez.runtime.io.sort.factor,

tez.runtime.unordered.output.buffer.size.mb,

tez.runtime.shuffle.parallel.copies,

tez.runtime.shuffle.fetch.buffer.percent,

tez.runtime.shuffle.merge.percent,

tez.runtime.sort.spill.percent,

// Compression

tez.runtime.compress,

tez.runtime.compress.codec,

tez.runtime.shuffle.enable.ssl,

// Grouping and Parallelism

tez.grouping.split-count,

tez.grouping.max-size,

tez.grouping.min-size,

tez.grouping.shuffle.enabled,

tez.vertex.max.output.consumers,

// Fault Tolerance

tez.am.task.max.failed.attempts,

tez.task.skip.enable,

tez.am.task.preemption.wait.timeout.millis,

// Logging and Monitoring

tez.staging-dir,

tez.am.application.tag,

tez.am.log.level,

tez.task.log.level,

tez.task.profiling.enabled,

tez.task.profiling.interval.millis,

// Counters and Limits

tez.counters.max,

tez.counters.groups.max,

tez.task.max.output.limit,

// Session Management

tez.session.mode,

tez.session.client.timeout.sec,

// Advanced Runtime

tez.runtime.transfer.data-via-events.enabled,

tez.runtime.pipelined-shuffle.enabled,

tez.runtime.optimize.local.fetch,

tez.runtime.ifile.readahead,

tez.runtime.ifile.readahead.bytes,

// Security

tez.am.view-acls,

tez.am.modify-acls,

tez.am.acls.enabled,

// Speculation

tez.am.speculation.enabled,

tez.am.speculation.speculative-capacity-factor,

// Recovery

tez.am.dag.recovery.enabled,

tez.am.dag.recovery.timeout.sec,

// Advanced Configuration

tez.task.get.task.sleep.interval-ms.max,

tez.am.heartbeat.interval-ms.max,

tez.runtime.key.class,

tez.runtime.value.class,

tez.runtime.key.comparator.class,