name value description hadoop.common.configuration.version 3.0.0 version of this configuration file hadoop.tmp.dir /tmp/hadoop-${user.name} A base for other temporary directories. hadoop.http.filter.initializers org.apache.hadoop.http.lib.StaticUserWebFilter A comma separated list of class names. Each class in the list must extend org.apache.hadoop.http.FilterInitializer. The corresponding Filter will be initialized. Then, the Filter will be applied to all user facing jsp and servlet web pages. The ordering of the list defines the ordering of the filters. hadoop.security.authorization false Is service-level authorization enabled? hadoop.security.instrumentation.requires.admin false Indicates if administrator ACLs are required to access instrumentation servlets (JMX, METRICS, CONF, STACKS). hadoop.security.authentication simple Possible values are simple (no authentication), and kerberos hadoop.security.group.mapping org.apache.hadoop.security.JniBasedUnixGroupsMappingWithFallback Class for user to group mapping (get groups for a given user) for ACL. The default implementation, org.apache.hadoop.security.JniBasedUnixGroupsMappingWithFallback, will determine if the Java Native Interface (JNI) is available. If JNI is available the implementation will use the API within hadoop to resolve a list of groups for a user. If JNI is not available then the shell implementation, ShellBasedUnixGroupsMapping, is used. This implementation shells out to the Linux/Unix environment with the bash -c groups command to resolve a list of groups for a user. hadoop.security.dns.interface The name of the Network Interface from which the service should determine its host name for Kerberos login. e.g. eth2. In a multi-homed environment, the setting can be used to affect the _HOST substitution in the service Kerberos principal. If this configuration value is not set, the service will use its default hostname as returned by InetAddress.getLocalHost().getCanonicalHostName(). Most clusters will not require this setting. hadoop.security.dns.nameserver The host name or IP address of the name server (DNS) which a service Node should use to determine its own host name for Kerberos Login. Requires hadoop.security.dns.interface. Most clusters will not require this setting. hadoop.security.dns.log-slow-lookups.enabled false Time name lookups (via SecurityUtil) and log them if they exceed the configured threshold. hadoop.security.dns.log-slow-lookups.threshold.ms 1000 If slow lookup logging is enabled, this threshold is used to decide if a lookup is considered slow enough to be logged. hadoop.security.groups.cache.secs 300 This is the config controlling the validity of the entries in the cache containing the user->group mapping. When this duration has expired, then the implementation of the group mapping provider is invoked to get the groups of the user and then cached back. hadoop.security.groups.negative-cache.secs 30 Expiration time for entries in the the negative user-to-group mapping caching, in seconds. This is useful when invalid users are retrying frequently. It is suggested to set a small value for this expiration, since a transient error in group lookup could temporarily lock out a legitimate user. Set this to zero or negative value to disable negative user-to-group caching. hadoop.security.groups.cache.warn.after.ms 5000 If looking up a single user to group takes longer than this amount of milliseconds, we will log a warning message. hadoop.security.groups.cache.background.reload false Whether to reload expired user->group mappings using a background thread pool. If set to true, a pool of hadoop.security.groups.cache.background.reload.threads is created to update the cache in the background. hadoop.security.groups.cache.background.reload.threads 3 Only relevant if hadoop.security.groups.cache.background.reload is true. Controls the number of concurrent background user->group cache entry refreshes. Pending refresh requests beyond this value are queued and processed when a thread is free. hadoop.security.groups.shell.command.timeout 0s Used by the ShellBasedUnixGroupsMapping class, this property controls how long to wait for the underlying shell command that is run to fetch groups. Expressed in seconds (e.g. 10s, 1m, etc.), if the running command takes longer than the value configured, the command is aborted and the groups resolver would return a result of no groups found. A value of 0s (default) would mean an infinite wait (i.e. wait until the command exits on its own). hadoop.security.group.mapping.ldap.connection.timeout.ms 60000 This property is the connection timeout (in milliseconds) for LDAP operations. If the LDAP provider doesn't establish a connection within the specified period, it will abort the connect attempt. Non-positive value means no LDAP connection timeout is specified in which case it waits for the connection to establish until the underlying network times out. hadoop.security.group.mapping.ldap.read.timeout.ms 60000 This property is the read timeout (in milliseconds) for LDAP operations. If the LDAP provider doesn't get a LDAP response within the specified period, it will abort the read attempt. Non-positive value means no read timeout is specified in which case it waits for the response infinitely. hadoop.security.group.mapping.ldap.url The URL of the LDAP server to use for resolving user groups when using the LdapGroupsMapping user to group mapping.

hadoop.security.group.mapping.ldap.ssl false Whether or not to use SSL when connecting to the LDAP server. hadoop.security.group.mapping.ldap.ssl.keystore File path to the SSL keystore that contains the SSL certificate required by the LDAP server. hadoop.security.group.mapping.ldap.ssl.keystore.password.file The path to a file containing the password of the LDAP SSL keystore. If the password is not configured in credential providers and the property hadoop.security.group.mapping.ldap.ssl.keystore.password is not set, LDAPGroupsMapping reads password from the file. IMPORTANT: This file should be readable only by the Unix user running the daemons and should be a local file.
hadoop.security.group.mapping.ldap.ssl.keystore.password The password of the LDAP SSL keystore. this property name is used as an alias to get the password from credential providers. If the password can not be found and hadoop.security.credential.clear-text-fallback is true LDAPGroupsMapping uses the value of this property for password. hadoop.security.credential.clear-text-fallback true true or false to indicate whether or not to fall back to storing credential password as clear text. The default value is true. This property only works when the password can't not be found from credential providers.
hadoop.security.credential.provider.path A comma-separated list of URLs that indicates the type and location of a list of providers that should be consulted. hadoop.security.credstore.java-keystore-provider.password-file The path to a file containing the custom password for all keystores that may be configured in the provider path. hadoop.security.group.mapping.ldap.bind.user The distinguished name of the user to bind as when connecting to the LDAP server. This may be left blank if the LDAP server supports anonymous binds.
hadoop.security.group.mapping.ldap.bind.password.file The path to a file containing the password of the bind user. If the password is not configured in credential providers and the property hadoop.security.group.mapping.ldap.bind.password is not set, LDAPGroupsMapping reads password from the file. IMPORTANT: This file should be readable only by the Unix user running the daemons and should be a local file. hadoop.security.group.mapping.ldap.bind.password The password of the bind user. this property name is used as an alias to get the password from credential providers. If the password can not be found and hadoop.security.credential.clear-text-fallback is true LDAPGroupsMapping uses the value of this property for password. hadoop.security.group.mapping.ldap.base The search base for the LDAP connection. This is a distinguished name, and will typically be the root of the LDAP directory.
hadoop.security.group.mapping.ldap.userbase The search base for the LDAP connection for user search query. This is a distinguished name, and its the root of the LDAP directory for users. If not set, hadoop.security.group.mapping.ldap.base is used. hadoop.security.group.mapping.ldap.groupbase The search base for the LDAP connection for group search . This is a distinguished name, and its the root of the LDAP directory for groups. If not set, hadoop.security.group.mapping.ldap.base is used.
hadoop.security.group.mapping.ldap.search.filter.user (&(objectClass=user)(sAMAccountName={0})) An additional filter to use when searching for LDAP users. The default will usually be appropriate for Active Directory installations. If connecting to an LDAP server with a non-AD schema, this should be replaced with (&(objectClass=inetOrgPerson)(uid={0}). {0} is a special string used to denote where the username fits into the filter. If the LDAP server supports posixGroups, Hadoop can enable the feature by setting the value of this property to "posixAccount" and the value of the hadoop.security.group.mapping.ldap.search.filter.group property to "posixGroup".
hadoop.security.group.mapping.ldap.search.filter.group (objectClass=group) An additional filter to use when searching for LDAP groups. This should be changed when resolving groups against a non-Active Directory installation. See the description of hadoop.security.group.mapping.ldap.search.filter.user to enable posixGroups support. hadoop.security.group.mapping.ldap.search.attr.memberof The attribute of the user object that identifies its group objects. By default, Hadoop makes two LDAP queries per user if this value is empty. If set, Hadoop will attempt to resolve group names from this attribute, instead of making the second LDAP query to get group objects. The value should be 'memberOf' for an MS AD installation.
hadoop.security.group.mapping.ldap.search.attr.member member The attribute of the group object that identifies the users that are members of the group. The default will usually be appropriate for any LDAP installation. hadoop.security.group.mapping.ldap.search.attr.group.name cn The attribute of the group object that identifies the group name. The default will usually be appropriate for all LDAP systems.
hadoop.security.group.mapping.ldap.search.group.hierarchy.levels 0 The number of levels to go up the group hierarchy when determining which groups a user is part of. 0 Will represent checking just the group that the user belongs to. Each additional level will raise the time it takes to execute a query by at most hadoop.security.group.mapping.ldap.directory.search.timeout. The default will usually be appropriate for all

LDAP systems. hadoop.security.group.mapping.ldap.posix.attr.uid.name uidNumber The attribute of posixAccount to use when groups for membership. Mostly useful for schemas wherein groups have memberUids that use an attribute other than uidNumber. hadoop.security.group.mapping.ldap.posix.attr.gid.name gidNumber The attribute of posixAccount indicating the group id. hadoop.security.group.mapping.ldap.directory.search.timeout 10000 The attribute applied to the LDAP SearchControl properties to set a maximum time limit when searching and awaiting a result. Set to 0 if infinite wait period is desired. Default is 10 seconds. Units in milliseconds. hadoop.security.group.mapping.providers Comma separated of names of other providers to provide user to group mapping. Used by CompositeGroupsMapping. hadoop.security.group.mapping.providers.combined true true or false to indicate whether groups from the providers are combined or not. The default value is true. If true, then all the providers will be tried to get groups and all the groups are combined to return as the final results. Otherwise, providers are tried one by one in the configured list order, and if any groups are retrieved from any provider, then the groups will be returned without trying the left ones. hadoop.security.service.user.name.key For those cases where the same RPC protocol is implemented by multiple servers, this configuration is required for specifying the principal name to use for the service when the client wishes to make an RPC call. fs.azure.user.agent.prefix unknown WASB passes User-Agent header to the Azure back-end. The default value contains WASB version, Java Runtime version, Azure Client library version, and the value of the configuration option fs.azure.user.agent.prefix. hadoop.security.uid.cache.secs 14400 This is the config controlling the validity of the entries in the cache containing the userId to userName and groupId to groupName used by NativeIO getFstat(). hadoop.rpc.protection authentication A comma-separated list of protection values for secured sasl connections. Possible values are authentication, integrity and privacy. authentication means authentication only and no integrity or privacy; integrity implies authentication and integrity are enabled; and privacy implies all of authentication, integrity and privacy are enabled. hadoop.security.saslproperties.resolver.class can be used to override the hadoop.rpc.protection for a connection at the server side. hadoop.security.saslproperties.resolver.class SaslPropertiesResolver used to resolve the QOP used for a connection. If not specified, the full set of values specified in hadoop.rpc.protection is used while determining the QOP used for the connection. If a class is specified, then the QOP values returned by the class will be used while determining the QOP used for the connection. hadoop.security.sensitive-config-keys secret$ password$ ssl.keystore.pass$ fs.s3.*[Ss]ecret.?[Kk]ey fs.s3a.*.server-side-encryption.key fs.azure.account.key.* credential$ oauth.*token$ hadoop.security.sensitive-config-keys A comma-separated or multi-line list of regular expressions to match configuration keys that should be redacted where appropriate, for example, when logging modified properties during a reconfiguration, private credentials should not be logged. hadoop.workaround.non.threadsafe.getpwuid true Some operating systems or authentication modules are known to have broken implementations of getpwuid_r and getpwgid_r, such that these calls are not thread-safe. Symptoms of this problem include JVM crashes with a stack trace inside these functions. If your system exhibits this issue, enable this configuration parameter to include a lock around the calls as a workaround. An incomplete list of some systems known to have this issue is available at http://wiki.apache.org/hadoop/KnownBrokenPwuidImplementations hadoop.kerberos.kinit.command kinit Used to periodically renew Kerberos credentials when provided to Hadoop. The default setting assumes that kinit is in the PATH of users running the Hadoop client. Change this to the absolute path to kinit if this is not the case. hadoop.kerberos.min.seconds.before.relogin 60 The minimum time between relogin attempts for Kerberos, in seconds. hadoop.security.auth_to_local Maps kerberos principals to local user names hadoop.token.files List of token cache files that have delegation tokens for hadoop service io.file.buffer.size 4096 The size of buffer for use in sequence files. The size of this buffer should probably be a multiple of hardware page size (4096 on Intel x86), and it determines how much data is buffered during read and write operations. io.bytes.per.checksum 512 The number of bytes per checksum. Must not be larger than io.file.buffer.size. io.skip.checksum.errors false If true, when a checksum error is encountered while reading a sequence file, entries are skipped, instead of throwing an exception. io.compression.codecs A comma-separated list of the compression codec classes that can be used for compression/decompression. In addition to any classes specified with this property (which take precedence), codec classes on the classpath are discovered using a Java ServiceLoader. io.compression.codec.bzip2.library system-native The native-code library to be used for compression and decompression by the bzip2 codec. This library could be specified either by by name or the full pathname. In the former case, the library is located by the dynamic

linker, usually searching the directories specified in the environment variable LD_LIBRARY_PATH. The value of "system-native" indicates that the default system library should be used. To indicate that the algorithm should operate entirely in Java, specify "java-builtin". io.serializations org.apache.hadoop.io.serializer.WritableSerialization, org.apache.hadoop.io.serializer.avro.AvroSpecificSerialization, org.apache.hadoop.io.serializer.avro.AvroReflectSerialization A list of serialization classes that can be used for obtaining serializers and deserializers. io.seqfile.local.dir ${hadoop.tmp.dir}/io/local The local directory where sequence file stores intermediate data files during merge. May be a comma-separated list of directories on different devices in order to spread disk i/o. Directories that do not exist are ignored. io.map.index.skip 0 Number of index entries to skip between each entry. Zero by default. Setting this to values larger than zero can facilitate opening large MapFiles using less memory. io.map.index.interval 128 MapFile consist of two files - data file (tuples) and index file (keys). For every io.map.index.interval records written in the data file, an entry (record-key, data-file-position) is written in the index file. This is to allow for doing binary search later within the index file to look up records by their keys and get their closest positions in the data file. io.erasurecode.codec.rs.rawcoders rs_native,rs_java Comma separated raw coder implementations for the rs codec. The earlier factory is prior to followings in case of failure of creating raw coders. io.erasurecode.codec.rs-legacy.rawcoders rs-legacy_java Comma separated raw coder implementations for the rs-legacy codec. The earlier factory is prior to followings in case of failure of creating raw coders. io.erasurecode.codec.xor.rawcoders xor_native,xor_java Comma separated raw coder implementations for the xor codec. The earlier factory is prior to followings in case of failure of creating raw coders. fs.defaultFS file:/// The name of the default file system. A URI whose scheme and authority determine the FileSystem implementation. The uri's scheme determines the config property (fs.SCHEME.impl) naming the FileSystem implementation class. The uri's authority is used to determine the host, port, etc. for a filesystem. fs.default.name file:/// Deprecated. Use (fs.defaultFS) property instead fs.trash.interval 0 Number of minutes after which the checkpoint gets deleted. If zero, the trash feature is disabled. This option may be configured both on the server and the client. If trash is disabled server side then the client side configuration is checked. If trash is enabled on the server side then the value configured on the server is used and the client configuration value is ignored. fs.trash.checkpoint.interval 0 Number of minutes between trash checkpoints. Should be smaller or equal to fs.trash.interval. If zero, the value is set to the value of fs.trash.interval. Every time the checkpointer runs it creates a new checkpoint out of current and removes checkpoints created more than fs.trash.interval minutes ago. fs.protected.directories A comma-separated list of directories which cannot be deleted even by the superuser unless they are empty. This setting can be used to guard important system directories against accidental deletion due to administrator error. fs.AbstractFileSystem.file.impl org.apache.hadoop.fs.local.LocalFs The AbstractFileSystem for file: uris. fs.AbstractFileSystem.har.impl org.apache.hadoop.fs.HarFs The AbstractFileSystem for har: uris. fs.AbstractFileSystem.hdfs.impl org.apache.hadoop.fs.Hdfs The FileSystem for hdfs: uris. fs.AbstractFileSystem.viewfs.impl org.apache.hadoop.fs.viewfs.ViewFs The AbstractFileSystem for view file system for viewfs: uris (ie client side mount table:). fs.viewfs.rename.strategy SAME_MOUNTPOINT Allowed rename strategy to rename between multiple mountpoints. Allowed values are SAME_MOUNTPOINT,SAME_TARGET_URI_ACROSS_MOUNTPOINT and SAME_FILESYSTEM_ACROSS_MOUNTPOINT. fs.AbstractFileSystem.ftp.impl org.apache.hadoop.fs.ftp.FtpFs The FileSystem for Ftp: uris. fs.AbstractFileSystem.webhdfs.impl org.apache.hadoop.fs.WebHdfs The FileSystem for webhdfs: uris. fs.AbstractFileSystem.swebhdfs.impl org.apache.hadoop.fs.SWebHdfs The FileSystem for swebhdfs: uris. fs.ftp.host 0.0.0.0 FTP filesystem connects to this server fs.ftp.host.port 21 FTP filesystem connects to fs.ftp.host on this port fs.ftp.data.connection.mode ACTIVE_LOCAL_DATA_CONNECTION_MODE Set the FTPClient's data connection mode based on configuration. Valid values are ACTIVE_LOCAL_DATA_CONNECTION_MODE, PASSIVE_LOCAL_DATA_CONNECTION_MODE and PASSIVE_REMOTE_DATA_CONNECTION_MODE. fs.ftp.transfer.mode BLOCK_TRANSFER_MODE Set FTP's transfer mode based on configuration. Valid values are STREAM_TRANSFER_MODE, BLOCK_TRANSFER_MODE and COMPRESSED_TRANSFER_MODE. fs.df.interval 60000 Disk usage statistics refresh interval in msec. fs.du.interval 600000 File space usage statistics refresh interval in msec. fs.swift.impl org.apache.hadoop.fs.swift.snative.SwiftNativeFileSystem The implementation class of the OpenStack Swift

Filesystem fs.automatic.close true By default, FileSystem instances are automatically closed at program exit using a JVM shutdown hook. Setting this property to false disables this behavior. This is an advanced option that should only be used by server applications requiring a more carefully orchestrated shutdown sequence. fs.s3a.access.key AWS access key ID used by S3A file system. Omit for IAM role-based or provider-based authentication. fs.s3a.secret.key AWS secret key used by S3A file system. Omit for IAM role-based or provider-based authentication. fs.s3a.aws.credentials.provider Comma-separated class names of credential provider classes which implement com.amazonaws.auth.AWSCredentialsProvider. These are loaded and queried in sequence for a valid set of credentials. Each listed class must implement one of the following means of construction, which are attempted in order: 1. a public constructor accepting java.net.URI and org.apache.hadoop.conf.Configuration, 2. a public static method named getInstance that accepts no arguments and returns an instance of com.amazonaws.auth.AWSCredentialsProvider, or 3. a public default constructor. Specifying org.apache.hadoop.fs.s3a.AnonymousAWSCredentialsProvider allows anonymous access to a publicly accessible S3 bucket without any credentials. Please note that allowing anonymous access to an S3 bucket compromises security and therefore is unsuitable for most use cases. It can be useful for accessing public data sets without requiring AWS credentials. If unspecified, then the default list of credential provider classes, queried in sequence, is: 1. org.apache.hadoop.fs.s3a.BasicAWSCredentialsProvider: supports static configuration of AWS access key ID and secret access key. See also fs.s3a.access.key and fs.s3a.secret.key. 2. com.amazonaws.auth.EnvironmentVariableCredentialsProvider: supports configuration of AWS access key ID and secret access key in environment variables named AWS_ACCESS_KEY_ID and AWS_SECRET_ACCESS_KEY, as documented in the AWS SDK. 3. com.amazonaws.auth.InstanceProfileCredentialsProvider: supports use of instance profile credentials if running in an EC2 VM. fs.s3a.session.token Session token, when using org.apache.hadoop.fs.s3a.TemporaryAWSCredentialsProvider as one of the providers. fs.s3a.security.credential.provider.path Optional comma separated list of credential providers, a list which is prepended to that set in hadoop.security.credential.provider.path fs.s3a.connection.maximum 15 Controls the maximum number of simultaneous connections to S3. fs.s3a.connection.ssl.enabled true Enables or disables SSL connections to S3. fs.s3a.endpoint AWS S3 endpoint to connect to. An up-to-date list is provided in the AWS Documentation: regions and endpoints. Without this property, the standard region (s3.amazonaws.com) is assumed. fs.s3a.path.style.access false Enable S3 path style access ie disabling the default virtual hosting behaviour. Useful for S3A-compliant storage providers as it removes the need to set up DNS for virtual hosting. fs.s3a.proxy.host Hostname of the (optional) proxy server for S3 connections. fs.s3a.proxy.port Proxy server port. If this property is not set but fs.s3a.proxy.host is, port 80 or 443 is assumed (consistent with the value of fs.s3a.connection.ssl.enabled). fs.s3a.proxy.username Username for authenticating with proxy server. fs.s3a.proxy.password Password for authenticating with proxy server. fs.s3a.proxy.domain Domain for authenticating with proxy server. fs.s3a.proxy.workstation Workstation for authenticating with proxy server. fs.s3a.attempts.maximum 20 How many times we should retry commands on transient errors. fs.s3a.connection.establish.timeout 5000 Socket connection setup timeout in milliseconds. fs.s3a.connection.timeout 200000 Socket connection timeout in milliseconds. fs.s3a.socket.send.buffer 8192 Socket send buffer hint to amazon connector. Represented in bytes. fs.s3a.socket.recv.buffer 8192 Socket receive buffer hint to amazon connector. Represented in bytes. fs.s3a.paging.maximum 5000 How many keys to request from S3 when doing directory listings at a time. fs.s3a.threads.max 10 The total number of threads available in the filesystem for data uploads *or any other queued filesystem operation*. fs.s3a.threads.keepalivetime 60 Number of seconds a thread can be idle before being terminated. fs.s3a.max.total.tasks 5 The number of operations which can be queued for execution fs.s3a.multipart.size 100M How big (in bytes) to split upload or copy operations up into. A suffix from the set {K,M,G,T,P} may be used to scale the numeric value. fs.s3a.multipart.threshold 2147483647 How big (in bytes) to split upload or copy operations up into. This also controls the partition size in renamed files, as rename() involves copying the source file(s). A suffix from the set {K,M,G,T,P} may be used to scale the numeric value. fs.s3a.multiobjectdelete.enable true When enabled, multiple single-object delete requests are replaced by a single 'delete multiple objects'-request, reducing the number of requests. Beware: legacy S3-compatible object stores might not support this request. fs.s3a.acl.default Set a canned ACL for newly created and copied objects. Value may be Private, PublicRead, PublicReadWrite, AuthenticatedRead, LogDeliveryWrite, BucketOwnerRead, or BucketOwnerFullControl. fs.s3a.multipart.purge false True if you

want to purge existing multipart uploads that may not have been completed/aborted correctly. The corresponding purge age is defined in fs.s3a.multipart.purge.age. If set, when the filesystem is instantiated then all outstanding uploads older than the purge age will be terminated -across the entire bucket. This will impact multipart uploads by other applications and users. so should be used sparingly, with an age value chosen to stop failed uploads, without breaking ongoing operations. fs.s3a.multipart.purge.age 86400 Minimum age in seconds of multipart uploads to purge. fs.s3a.server-side-encryption-algorithm Specify a server-side encryption algorithm for s3a: file system. Unset by default. It supports the following values: 'AES256' (for SSE-S3), 'SSE-KMS' and 'SSE-C'. fs.s3a.server-side-encryption.key Specific encryption key to use if fs.s3a.server-side-encryption-algorithm has been set to 'SSE-KMS' or 'SSE-C'. In the case of SSE-C, the value of this property should be the Base64 encoded key. If you are using SSE-KMS and leave this property empty, you'll be using your default's S3 KMS key, otherwise you should set this property to the specific KMS key id. fs.s3a.signing-algorithm Override the default signing algorithm so legacy implementations can still be used fs.s3a.block.size 32M Block size to use when reading files using s3a: file system. A suffix from the set {K,M,G,T,P} may be used to scale the numeric value. fs.s3a.buffer.dir ${hadoop.tmp.dir}/s3a Comma separated list of directories that will be used to buffer file uploads to. fs.s3a.fast.upload.buffer disk The buffering mechanism to for data being written. Values: disk, array, bytebuffer. "disk" will use the directories listed in fs.s3a.buffer.dir as the location(s) to save data prior to being uploaded. "array" uses arrays in the JVM heap "bytebuffer" uses off-heap memory within the JVM. Both "array" and "bytebuffer" will consume memory in a single stream up to the number of blocks set by: fs.s3a.multipart.size * fs.s3a.fast.upload.active.blocks. If using either of these mechanisms, keep this value low The total number of threads performing work across all threads is set by fs.s3a.threads.max, with fs.s3a.max.total.tasks values setting the number of queued work items. fs.s3a.fast.upload.active.blocks 4 Maximum Number of blocks a single output stream can have active (uploading, or queued to the central FileSystem instance's pool of queued operations. This stops a single stream overloading the shared thread pool. fs.s3a.readahead.range 64K Bytes to read ahead during a seek() before closing and re-opening the S3 HTTP connection. This option will be overridden if any call to setReadahead() is made to an open stream. A suffix from the set {K,M,G,T,P} may be used to scale the numeric value. fs.s3a.user.agent.prefix Sets a custom value that will be prepended to the User-Agent header sent in HTTP requests to the S3 back-end by S3AFileSystem. The User-Agent header always includes the Hadoop version number followed by a string generated by the AWS SDK. An example is "User-Agent: Hadoop 2.8.0, aws-sdk-java/1.10.6". If this optional property is set, then its value is prepended to create a customized User-Agent. For example, if this configuration property was set to "MyApp", then an example of the resulting User-Agent would be "User-Agent: MyApp, Hadoop 2.8.0, aws-sdk-java/1.10.6". fs.s3a.metadatastore.authoritative false When true, allow MetadataStore implementations to act as source of truth for getting file status and directory listings. Even if this is set to true, MetadataStore implementations may choose not to return authoritative results. If the configured MetadataStore does not support being authoritative, this setting will have no effect. fs.s3a.metadatastore.impl org.apache.hadoop.fs.s3a.s3guard.NullMetadataStore Fully-qualified name of the class that implements the MetadataStore to be used by s3a. The default class, NullMetadataStore, has no effect: s3a will continue to treat the backing S3 service as the one and only source of truth for file and directory metadata. fs.s3a.s3guard.cli.prune.age 86400000 Default age (in milliseconds) after which to prune metadata from the metadatastore when the prune command is run. Can be overridden on the command-line. fs.s3a.impl org.apache.hadoop.fs.s3a.S3AFileSystem The implementation class of the S3A Filesystem fs.s3a.s3guard.ddb.region AWS DynamoDB region to connect to. An up-to-date list is provided in the AWS Documentation: regions and endpoints. Without this property, the S3Guard will operate table in the associated S3 bucket region. fs.s3a.s3guard.ddb.table The DynamoDB table name to operate. Without this property, the respective S3 bucket name will be used. fs.s3a.s3guard.ddb.table.create false If true, the S3A client will create the table if it does not already exist. fs.s3a.s3guard.ddb.table.capacity.read 500 Provisioned throughput requirements for read operations in terms of capacity units for the DynamoDB table. This config value will only be used when creating a new DynamoDB table, though later you can manually provision by increasing or decreasing read capacity as needed for existing tables. See DynamoDB documents for more information. fs.s3a.s3guard.ddb.table.capacity.write 100 Provisioned throughput requirements for write operations in terms of capacity units for the DynamoDB table. Refer to related config fs.s3a.s3guard.ddb.table.capacity.read before usage. fs.s3a.s3guard.ddb.max.retries 9 Max retries on batched DynamoDB operations before giving up and throwing an IOException. Each retry is delayed with an

exponential backoff timer which starts at 100 milliseconds and approximately doubles each time. The minimum wait before throwing an exception is sum(100, 200, 400, 800, .. 100*2^N-1) == 100 * ((2^N)-1) So N = 9 yields at least 51.1 seconds (51,100) milliseconds of blocking before throwing an IOException. fs.s3a.s3guard.ddb.background.sleep 25 Length (in milliseconds) of pause between each batch of deletes when pruning metadata. Prevents prune operations (which can typically be low priority background operations) from overly interfering with other I/O operations. fs.AbstractFileSystem.s3a.impl org.apache.hadoop.fs.s3a.S3A The implementation class of the S3A AbstractFileSystem. fs.s3a.list.version 2 Select which version of the S3 SDK's List Objects API to use. Currently support 2 (default) and 1 (older API). fs.wasb.impl org.apache.hadoop.fs.azure.NativeAzureFileSystem The implementation class of the Native Azure Filesystem fs.wasbs.impl org.apache.hadoop.fs.azure.NativeAzureFileSystem$Secure The implementation class of the Secure Native Azure Filesystem fs.azure.secure.mode false Config flag to identify the mode in which fs.azure.NativeAzureFileSystem needs to run under. Setting it "true" would make fs.azure.NativeAzureFileSystem use SAS keys to communicate with Azure storage. fs.azure.local.sas.key.mode false Works in conjuction with fs.azure.secure.mode. Setting this config to true results in fs.azure.NativeAzureFileSystem using the local SAS key generation where the SAS keys are generating in the same process as fs.azure.NativeAzureFileSystem. If fs.azure.secure.mode flag is set to false, this flag has no effect. fs.azure.sas.expiry.period 90d The default value to be used for expiration period for SAS keys generated. Can use the following suffix (case insensitive): ms(millis), s(sec), m(min), h(hour), d(day) to specify the time (such as 2s, 2m, 1h, etc.). fs.azure.authorization false Config flag to enable authorization support in WASB. Setting it to "true" enables authorization support to WASB. Currently WASB authorization requires a remote service to provide authorization that needs to be specified via fs.azure.authorization.remote.service.url configuration fs.azure.authorization.caching.enable true Config flag to enable caching of authorization results and saskeys in WASB. This flag is relevant only when fs.azure.authorization is enabled. fs.azure.saskey.usecontainersaskeyforallaccess true Use container saskey for access to all blobs within the container. Blob-specific saskeys are not used when this setting is enabled. This setting provides better performance compared to blob-specific saskeys. io.seqfile.compress.blocksize 1000000 The minimum block size for compression in block compressed SequenceFiles.

io.mapfile.bloom.size 1048576 The size of BloomFilter-s used in BloomMapFile. Each time this many keys is appended the next BloomFilter will be created (inside a DynamicBloomFilter). Larger values minimize the number of filters, which slightly increases the performance, but may waste too much space if the total number of keys is usually much smaller than this number. io.mapfile.bloom.error.rate 0.005 The rate of false positives in BloomFilter-s used in BloomMapFile. As this value decreases, the size of BloomFilter-s increases exponentially. This value is the probability of encountering false positives (default is 0.5%). hadoop.util.hash.type murmur The default implementation of Hash. Currently this can take one of the two values: 'murmur' to select MurmurHash and 'jenkins' to select JenkinsHash. ipc.client.idlethreshold 4000 Defines the threshold number of connections after which connections will be inspected for idleness. ipc.client.kill.max 10 Defines the maximum number of clients to disconnect in one go. ipc.client.connection.maxidletime 10000 The maximum time in msec after which a client will bring down the connection to the server. ipc.client.connect.max.retries 10 Indicates the number of retries a client will make to establish a server connection. ipc.client.connect.retry.interval 1000 Indicates the number of milliseconds a client will wait for before retrying to establish a server connection. ipc.client.connect.timeout 20000 Indicates the number of milliseconds a client will wait for the socket to establish a server connection. ipc.client.connect.max.retries.on.timeouts 45 Indicates the number of retries a client will make on socket timeout to establish a server connection. ipc.client.tcpnodelay true Use TCP_NODELAY flag to bypass Nagle's algorithm transmission delays. ipc.client.low-latency false Use low-latency QoS markers for IPC connections. ipc.client.ping true Send a ping to the server when timeout on reading the response, if set to true. If no failure is detected, the client retries until at least a byte is read or the time given by ipc.client.rpc-timeout.ms is passed. ipc.ping.interval 60000 Timeout on waiting response from server, in milliseconds. The client will send ping when the interval is passed without receiving bytes, if ipc.client.ping is set to true. ipc.client.rpc-timeout.ms 0 Timeout on waiting response from server, in milliseconds. If ipc.client.ping is set to true and this rpc-timeout is greater than the value of ipc.ping.interval, the effective value of the rpc-timeout is rounded up to multiple of ipc.ping.interval. ipc.server.listen.queue.size 128 Indicates the length of the listen queue for servers accepting client connections. ipc.server.log.slow.rpc false This setting is useful to troubleshoot performance issues for various services. If this value is set to true then we log requests that fall

into 99th percentile as well as increment RpcSlowCalls counter. ipc.maximum.data.length 67108864 This indicates the maximum IPC message length (bytes) that can be accepted by the server. Messages larger than this value are rejected by the immediately to avoid possible OOMs. This setting should rarely need to be changed. ipc.maximum.response.length 134217728 This indicates the maximum IPC message length (bytes) that can be accepted by the client. Messages larger than this value are rejected immediately to avoid possible OOMs. This setting should rarely need to be changed. Set to 0 to disable. hadoop.security.impersonation.provider.class A class which implements ImpersonationProvider interface, used to authorize whether one user can impersonate a specific user. If not specified, the DefaultImpersonationProvider will be used. If a class is specified, then that class will be used to determine the impersonation capability. hadoop.rpc.socket.factory.class.default org.apache.hadoop.net.StandardSocketFactory Default SocketFactory to use. This parameter is expected to be formatted as "package.FactoryClassName". hadoop.rpc.socket.factory.class.ClientProtocol SocketFactory to use to connect to a DFS. If null or empty, use hadoop.rpc.socket.class.default. This socket factory is also used by DFSClient to create sockets to DataNodes. hadoop.socks.server Address (host:port) of the SOCKS server to be used by the SocksSocketFactory. net.topology.node.switch.mapping.impl org.apache.hadoop.net.ScriptBasedMapping The default implementation of the DNSToSwitchMapping. It invokes a script specified in net.topology.script.file.name to resolve node names. If the value for net.topology.script.file.name is not set, the default value of DEFAULT_RACK is returned for all node names. net.topology.impl org.apache.hadoop.net.NetworkTopology The default implementation of NetworkTopology which is classic three layer one. net.topology.script.file.name The script name that should be invoked to resolve DNS names to NetworkTopology names. Example: the script would take host.foo.bar as an argument, and return /rack1 as the output. net.topology.script.number.args 100 The max number of args that the script configured with net.topology.script.file.name should be run with. Each arg is an IP address. net.topology.table.file.name The file name for a topology file, which is used when the net.topology.node.switch.mapping.impl property is set to org.apache.hadoop.net.TableMapping. The file format is a two column text file, with columns separated by whitespace. The first column is a DNS or IP address and the second column specifies the rack where the address maps. If no entry corresponding to a host in the cluster is found, then /default-rack is assumed. file.stream-buffer-size 4096 The size of buffer to stream files. The size of this buffer should probably be a multiple of hardware page size (4096 on Intel x86), and it determines how much data is buffered during read and write operations. file.bytes-per-checksum 512 The number of bytes per checksum. Must not be larger than file.stream-buffer-size file.client-write-packet-size 65536 Packet size for clients to write file.blocksize 67108864 Block size file.replication 1 Replication factor ftp.stream-buffer-size 4096 The size of buffer to stream files. The size of this buffer should probably be a multiple of hardware page size (4096 on Intel x86), and it determines how much data is buffered during read and write operations. ftp.bytes-per-checksum 512 The number of bytes per checksum. Must not be larger than ftp.stream-buffer-size ftp.client-write-packet-size 65536 Packet size for clients to write ftp.blocksize 67108864 Block size ftp.replication 3 Replication factor tfile.io.chunk.size 1048576 Value chunk size in bytes. Default to 1MB. Values of the length less than the chunk size is guaranteed to have known value length in read time (See also TFile.Reader.Scanner.Entry.isValueLengthKnown()). tfile.fs.output.buffer.size 262144 Buffer size used for FSDataOutputStream in bytes. tfile.fs.input.buffer.size 262144 Buffer size used for FSDataInputStream in bytes. hadoop.http.authentication.type simple Defines authentication used for Oozie HTTP endpoint. Supported values are: simple | kerberos | #AUTHENTICATION_HANDLER_CLASSNAME# hadoop.http.authentication.token.validity 36000 Indicates how long (in seconds) an authentication token is valid before it has to be renewed. hadoop.http.authentication.signature.secret.file ${user.home}/hadoop-http-auth-signature-secret The signature secret for signing the authentication tokens. The same secret should be used for JT/NN/DN/TT configurations. hadoop.http.authentication.cookie.domain The domain to use for the HTTP cookie that stores the authentication token. In order to authentiation to work correctly across all Hadoop nodes web-consoles the domain must be correctly set. IMPORTANT: when using IP addresses, browsers ignore cookies with domain settings. For this setting to work properly all nodes in the cluster must be configured to generate URLs with hostname.domain names on it. hadoop.http.authentication.simple.anonymous.allowed true Indicates if anonymous requests are allowed when using 'simple' authentication. hadoop.http.authentication.kerberos.principal HTTP/_HOST@LOCALHOST Indicates the Kerberos principal to be used for HTTP endpoint. The principal MUST start with 'HTTP/' as per Kerberos HTTP

SPNEGO specification. hadoop.http.authentication.kerberos.keytab ${user.home}/hadoop.keytab Location of the keytab file with the credentials for the principal. Referring to the same keytab file Oozie uses for its Kerberos credentials for Hadoop. hadoop.http.cross-origin.enabled false Enable/disable the cross-origin (CORS) filter. hadoop.http.cross-origin.allowed-origins * Comma separated list of origins that are allowed for web services needing cross-origin (CORS) support. Wildcards (*) and patterns allowed hadoop.http.cross-origin.allowed-methods GET,POST,HEAD Comma separated list of methods that are allowed for web services needing cross-origin (CORS) support. hadoop.http.cross-origin.allowed-headers X-Requested-With,Content-Type,Accept,Origin Comma separated list of headers that are allowed for web services needing cross-origin (CORS) support. hadoop.http.cross-origin.max-age 1800 The number of seconds a pre-flighted request can be cached for web services needing cross-origin (CORS) support. dfs.ha.fencing.methods List of fencing methods to use for service fencing. May contain builtin methods (eg shell and sshfence) or user-defined method. dfs.ha.fencing.ssh.connect-timeout 30000 SSH connection timeout, in milliseconds, to use with the builtin sshfence fencer. dfs.ha.fencing.ssh.private-key-files The SSH private key files to use with the builtin sshfence fencer. ha.zookeeper.quorum A list of ZooKeeper server addresses, separated by commas, that are to be used by the ZKFailoverController in automatic failover. ha.zookeeper.session-timeout.ms 5000 The session timeout to use when the ZKFC connects to ZooKeeper. Setting this value to a lower value implies that server crashes will be detected more quickly, but risks triggering failover too aggressively in the case of a transient error or network blip. ha.zookeeper.parent-znode /hadoop-ha The ZooKeeper znode under which the ZK failover controller stores its information. Note that the nameservice ID is automatically appended to this znode, so it is not normally necessary to configure this, even in a federated environment. ha.zookeeper.acl world:anyone:rwcda A comma-separated list of ZooKeeper ACLs to apply to the znodes used by automatic failover. These ACLs are specified in the same format as used by the ZooKeeper CLI. If the ACL itself contains secrets, you may instead specify a path to a file, prefixed with the '@' symbol, and the value of this configuration will be loaded from within. ha.zookeeper.auth A comma-separated list of ZooKeeper authentications to add when connecting to ZooKeeper. These are specified in the same format as used by the "addauth" command in the ZK CLI. It is important that the authentications specified here are sufficient to access znodes with the ACL specified in ha.zookeeper.acl. If the auths contain secrets, you may instead specify a path to a file, prefixed with the '@' symbol, and the value of this configuration will be loaded from within. hadoop.http.staticuser.user dr.who The user name to filter as, on static web filters while rendering content. An example use is the HDFS web UI (user to be used for browsing files). hadoop.ssl.keystores.factory.class org.apache.hadoop.security.ssl.FileBasedKeyStoresFactory The keystores factory to use for retrieving certificates. hadoop.ssl.require.client.cert false Whether client certificates are required hadoop.ssl.hostname.verifier DEFAULT The hostname verifier to provide for HttpsURLConnections. Valid values are: DEFAULT, STRICT, STRICT_IE6, DEFAULT_AND_LOCALHOST and ALLOW_ALL hadoop.ssl.server.conf ssl-server.xml Resource file from which ssl server keystore information will be extracted. This file is looked up in the classpath, typically it should be in Hadoop conf/ directory. hadoop.ssl.client.conf ssl-client.xml Resource file from which ssl client keystore information will be extracted This file is looked up in the classpath, typically it should be in Hadoop conf/ directory. hadoop.ssl.enabled false Deprecated. Use dfs.http.policy and yarn.http.policy instead. hadoop.ssl.enabled.protocols TLSv1,SSLv2Hello,TLSv1.1,TLSv1.2 The supported SSL protocols. hadoop.jetty.logs.serve.aliases true Enable/Disable aliases serving from jetty fs.permissions.umask-mode 022 The umask used when creating files and directories. Can be in octal or in symbolic. Examples are: "022" (octal for u=rwx,g=r-x,o=r-x in symbolic), or "u=rwx,g=rwx,o=" (symbolic for 007 in octal). ha.health-monitor.connect-retry-interval.ms 1000 How often to retry connecting to the service. ha.health-monitor.check-interval.ms 1000 How often to check the service. ha.health-monitor.sleep-after-disconnect.ms 1000 How long to sleep after an unexpected RPC error. ha.health-monitor.rpc-timeout.ms 45000 Timeout for the actual monitorHealth() calls. ha.failover-controller.new-active.rpc-timeout.ms 60000 Timeout that the FC waits for the new active to become active ha.failover-controller.graceful-fence.rpc-timeout.ms 5000 Timeout that the FC waits for the old active to go to standby ha.failover-controller.graceful-fence.connection.retries 1 FC connection retries for graceful fencing ha.failover-controller.cli-check.rpc-timeout.ms 20000 Timeout that the CLI (manual) FC waits for monitorHealth, getServiceState ipc.client.fallback-to-simple-auth-allowed false When a client is configured to attempt a secure connection, but attempts to connect to an insecure server, that server may instruct the client to switch to SASL SIMPLE

(unsecure) authentication. This setting controls whether or not the client will accept this instruction from the server. When false (the default), the client will not allow the fallback to SIMPLE authentication, and will abort the connection. fs.client.resolve.remote.symlinks true Whether to resolve symlinks when accessing a remote Hadoop filesystem. Setting this to false causes an exception to be thrown upon encountering a symlink. This setting does not apply to local filesystems, which automatically resolve local symlinks. nfs.exports.allowed.hosts * rw By default, the export can be mounted by any client. The value string contains machine name and access privilege, separated by whitespace characters. The machine name format can be a single host, a Java regular expression, or an IPv4 address. The access privilege uses rw or ro to specify read/write or read-only access of the machines to exports. If the access privilege is not provided, the default is read-only. Entries are separated by ";". For example: "192.168.0.0/22 rw ; host.*\.example\.com ; host1.test.org ro;". Only the NFS gateway needs to restart after this property is updated. hadoop.user.group.static.mapping.overrides dr.who=; Static mapping of user to groups. This will override the groups if available in the system for the specified user. In other words, groups look-up will not happen for these users, instead groups mapped in this configuration will be used. Mapping should be in this format. user1=group1,group2;user2=;user3=group2; Default, "dr.who=;" will consider "dr.who" as user without groups. rpc.metrics.quantile.enable false Setting this property to true and rpc.metrics.percentiles.intervals to a comma-separated list of the granularity in seconds, the 50/75/90/95/99th percentile latency for rpc queue/processing time in milliseconds are added to rpc metrics. rpc.metrics.percentiles.intervals A comma-separated list of the granularity in seconds for the metrics which describe the 50/75/90/95/99th percentile latency for rpc queue/processing time. The metrics are outputted if rpc.metrics.quantile.enable is set to true. hadoop.security.crypto.codec.classes.EXAMPLECIPHERSUITE The prefix for a given crypto codec, contains a comma-separated list of implementation classes for a given crypto codec (eg EXAMPLECIPHERSUITE). The first implementation will be used if available, others are fallbacks. hadoop.security.crypto.codec.classes.aes.ctr.nopadding org.apache.hadoop.crypto.OpensslAesCtrCryptoCodec, org.apache.hadoop.crypto.JceAesCtrCryptoCodec Comma-separated list of crypto codec implementations for AES/CTR/NoPadding. The first implementation will be used if available, others are fallbacks. hadoop.security.crypto.cipher.suite AES/CTR/NoPadding Cipher suite for crypto codec. hadoop.security.crypto.jce.provider The JCE provider name used in CryptoCodec. hadoop.security.crypto.buffer.size 8192 The buffer size used by CryptoInputStream and CryptoOutputStream. hadoop.security.java.secure.random.algorithm SHA1PRNG The java secure random algorithm. hadoop.security.secure.random.impl Implementation of secure random. hadoop.security.random.device.file.path /dev/urandom OS security random device file path. hadoop.security.key.provider.path The KeyProvider to use when managing zone keys, and interacting with encryption keys when reading and writing to an encryption zone. For hdfs clients, the provider path will be same as namenode's provider path. fs.har.impl.disable.cache true Don't cache 'har' filesystem instances. hadoop.security.kms.client.authentication.retry-count 1 Number of time to retry connecting to KMS on authentication failure hadoop.security.kms.client.encrypted.key.cache.size 500 Size of the EncryptedKeyVersion cache Queue for each key hadoop.security.kms.client.encrypted.key.cache.low-watermark 0.3f If size of the EncryptedKeyVersion cache Queue falls below the low watermark, this cache queue will be scheduled for a refill hadoop.security.kms.client.encrypted.key.cache.num.refill.threads 2 Number of threads to use for refilling depleted EncryptedKeyVersion cache Queues hadoop.security.kms.client.encrypted.key.cache.expiry 43200000 Cache expiry time for a Key, after which the cache Queue for this key will be dropped. Default = 12hrs hadoop.security.kms.client.timeout 60 Sets value for KMS client connection timeout, and the read timeout to KMS servers. hadoop.security.kms.client.failover.sleep.base.millis 100 Expert only. The time to wait, in milliseconds, between failover attempts increases exponentially as a function of the number of attempts made so far, with a random factor of +/- 50%. This option specifies the base value used in the failover calculation. The first failover will retry immediately. The 2nd failover attempt will delay at least hadoop.security.client.failover.sleep.base.millis milliseconds. And so on. hadoop.security.kms.client.failover.sleep.max.millis 2000 Expert only. The time to wait, in milliseconds, between failover attempts increases exponentially as a function of the number of attempts made so far, with a random factor of +/- 50%. This option specifies the maximum value to wait between failovers. Specifically, the time between two failover attempts will not exceed +/- 50% of hadoop.security.client.failover.sleep.max.millis milliseconds. ipc.server.max.connections 0 The maximum

number of concurrent connections a server is allowed to accept. If this limit is exceeded, incoming connections will first fill the listen queue and then may go to an OS-specific listen overflow queue. The client may fail or timeout, but the server can avoid running out of file descriptors using this feature. 0 means no limit. hadoop.registry.rm.enabled false Is the registry enabled in the YARN Resource Manager? If true, the YARN RM will, as needed. create the user and system paths, and purge service records when containers, application attempts and applications complete. If false, the paths must be created by other means, and no automatic cleanup of service records will take place. hadoop.registry.zk.root /registry The root zookeeper node for the registry hadoop.registry.zk.session.timeout.ms 60000 Zookeeper session timeout in milliseconds hadoop.registry.zk.connection.timeout.ms 15000 Zookeeper connection timeout in milliseconds hadoop.registry.zk.retry.times 5 Zookeeper connection retry count before failing hadoop.registry.zk.retry.interval.ms 1000 hadoop.registry.zk.retry.ceiling.ms 60000 Zookeeper retry limit in milliseconds, during exponential backoff. This places a limit even if the retry times and interval limit, combined with the backoff policy, result in a long retry period hadoop.registry.zk.quorum localhost:2181 List of hostname:port pairs defining the zookeeper quorum binding for the registry hadoop.registry.secure false Key to set if the registry is secure. Turning it on changes the permissions policy from "open access" to restrictions on kerberos with the option of a user adding one or more auth key pairs down their own tree. hadoop.registry.system.acls sasl:yarn@, sasl:mapred@, sasl:hdfs@ A comma separated list of Zookeeper ACL identifiers with system access to the registry in a secure cluster. These are given full access to all entries. If there is an "@" at the end of a SASL entry it instructs the registry client to append the default kerberos domain. hadoop.registry.kerberos.realm The kerberos realm: used to set the realm of system principals which do not declare their realm, and any other accounts that need the value. If empty, the default realm of the running process is used. If neither are known and the realm is needed, then the registry service/client will fail. hadoop.registry.jaas.context Client Key to define the JAAS context. Used in secure mode hadoop.shell.missing.defaultFs.warning false Enable hdfs shell commands to display warnings if (fs.defaultFS) property is not set. hadoop.shell.safely.delete.limit.num.files 100 Used by -safely option of hadoop fs shell -rm command to avoid accidental deletion of large directories. When enabled, the -rm command requires confirmation if the number of files to be deleted is greater than this limit. The default limit is 100 files. The warning is disabled if the limit is 0 or the -safely is not specified in -rm command. fs.client.htrace.sampler.classes The class names of the HTrace Samplers to use for Hadoop filesystem clients. hadoop.htrace.span.receiver.classes The class names of the Span Receivers to use for Hadoop. hadoop.http.logs.enabled true Enable the "/logs" endpoint on all Hadoop daemons, which serves local logs, but may be considered a security risk due to it listing the contents of a directory. fs.client.resolve.topology.enabled false Whether the client machine will use the class specified by property net.topology.node.switch.mapping.impl to compute the network distance between itself and remote machines of the FileSystem. Additional properties might need to be configured depending on the class specified in net.topology.node.switch.mapping.impl. For example, if org.apache.hadoop.net.ScriptBasedMapping is used, a valid script file needs to be specified in net.topology.script.file.name. fs.adl.impl org.apache.hadoop.fs.adl.AdlFileSystem fs.AbstractFileSystem.adl.impl org.apache.hadoop.fs.adl.Adl adl.feature.ownerandgroup.enableupn false When true : User and Group in FileStatus/AclStatus response is represented as user friendly name as per Azure AD profile. When false (default) : User and Group in FileStatus/AclStatus response is represented by the unique identifier from Azure AD profile (Object ID as GUID). For optimal performance, false is recommended. fs.adl.oauth2.access.token.provider.type ClientCredential Defines Azure Active Directory OAuth2 access token provider type. Supported types are ClientCredential, RefreshToken, MSI, DeviceCode, and Custom. The ClientCredential type requires property fs.adl.oauth2.client.id, fs.adl.oauth2.credential, and fs.adl.oauth2.refresh.url. The RefreshToken type requires property fs.adl.oauth2.client.id and fs.adl.oauth2.refresh.token. The MSI type reads optional property fs.adl.oauth2.msi.port, if specified. The DeviceCode type requires property fs.adl.oauth2.devicecode.clientapp.id. The Custom type requires property fs.adl.oauth2.access.token.provider. fs.adl.oauth2.client.id The OAuth2 client id. fs.adl.oauth2.credential The OAuth2 access key. fs.adl.oauth2.refresh.url The OAuth2 token endpoint. fs.adl.oauth2.refresh.token The OAuth2 refresh token. fs.adl.oauth2.access.token.provider The class name of the OAuth2 access token provider. fs.adl.oauth2.msi.port The localhost port for the MSI token service. This is the port specified when creating the Azure VM. The default, if this setting is not specified, is 50342. Used by MSI token provider. fs.adl.oauth2.devicecode.clientapp.id The app id of the AAD native app in whose context the auth request

should be made. Used by DeviceCode token provider. hadoop.caller.context.enabled false When the feature is enabled, additional fields are written into name-node audit log records for auditing coarse granularity operations. hadoop.caller.context.max.size 128 The maximum bytes a caller context string can have. If the passed caller context is longer than this maximum bytes, client will truncate it before sending to server. Note that the server may have a different maximum size, and will truncate the caller context to the maximum size it allows. hadoop.caller.context.signature.max.size 40 The caller's signature (optional) is for offline validation. If the signature exceeds the maximum allowed bytes in server, the caller context will be abandoned, in which case the caller context will not be recorded in audit logs. seq.io.sort.mb 100 The total amount of buffer memory to use while sorting files, while using SequenceFile.Sorter, in megabytes. By default, gives each merge stream 1MB, which should minimize seeks. seq.io.sort.factor 100 The number of streams to merge at once while sorting files using SequenceFile.Sorter. This determines the number of open file handles. hadoop.zk.address Host:Port of the ZooKeeper server to be used. hadoop.zk.num-retries 1000 Number of tries to connect to ZooKeeper. hadoop.zk.retry-interval-ms 1000 Retry interval in milliseconds when connecting to ZooKeeper. hadoop.zk.timeout-ms 10000 ZooKeeper session timeout in milliseconds. Session expiration is managed by the ZooKeeper cluster itself, not by the client. This value is used by the cluster to determine when the client's session expires. Expirations happens when the cluster does not hear from the client within the specified session timeout period (i.e. no heartbeat). hadoop.zk.acl world:anyone:rwcda ACL's to be used for ZooKeeper znodes. hadoop.zk.auth Specify the auths to be used for the ACL's specified in hadoop.zk.acl. This takes a comma-separated list of authentication mechanisms, each of the form 'scheme:auth' (the same syntax used for the 'addAuth' command in the ZK CLI). hadoop.treat.subject.external false When creating UGI with UserGroupInformation(Subject), treat the passed subject external if set to true, and assume the owner of the subject should do the credential renewal. When true this property will introduce an incompatible change which may require changes in client code. For more details, see the jiras: HADOOP-13805,HADOOP-13558.