Using Ambari Core Services

Date of Publish: 2018-04-30

http://docs.hortonworks.com

Contents

Using Ambari Core Services

The Ambari core services enable you to monitor, analyze, and search the operating status of hosts in your cluster.

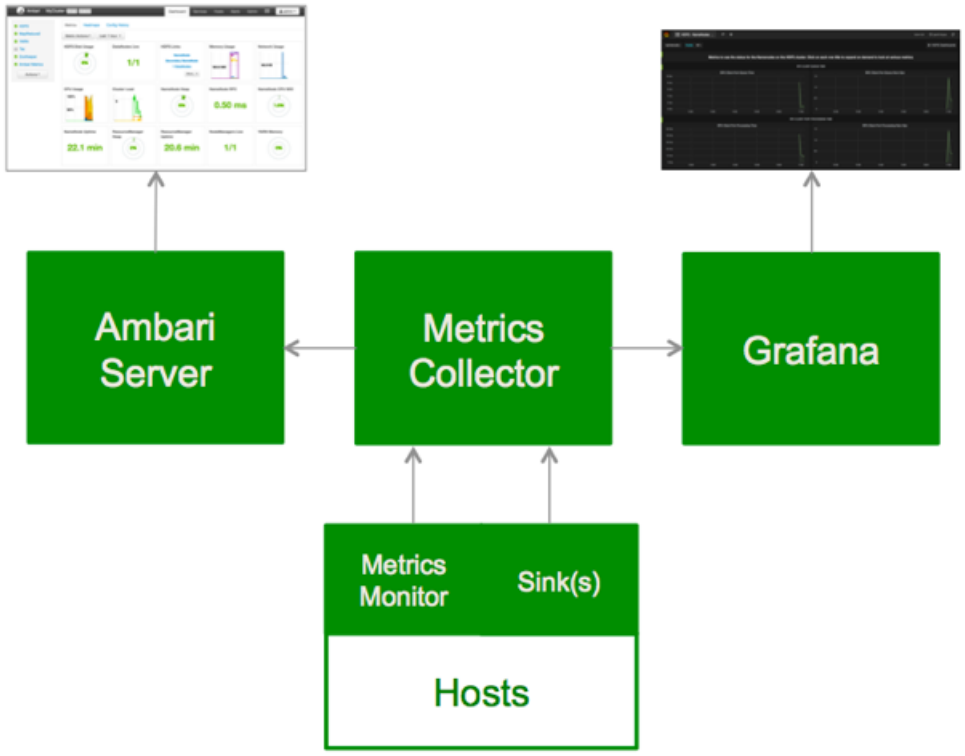
Understanding Ambari Metrics Service

Ambari Metrics System (AMS) collects, aggregates, and serves Hadoop and system metrics in Ambari-managed clusters.

AMS has four components: Metrics Monitors, Hadoop Sinks, Metrics Collector, and Grafana.

- **Metrics Monitors** on each host in the cluster collect system-level metrics and publish to the Metrics Collector.
- **Hadoop Sinks** plug in to Hadoop components to publish Hadoop metrics to the Metrics Collector.
- The **Metrics Collector** is a daemon that runs on a specific host in the cluster and receives data from the registered publishers, the Monitors, and the Sinks.
- **Grafana** is a daemon that runs on a specific host in the cluster and serves pre-built dashboards for visualizing metrics collected in the Metrics Collector.

This conceptual diagram shows how the components of AMS work together to collect metrics and make those metrics available to Ambari.

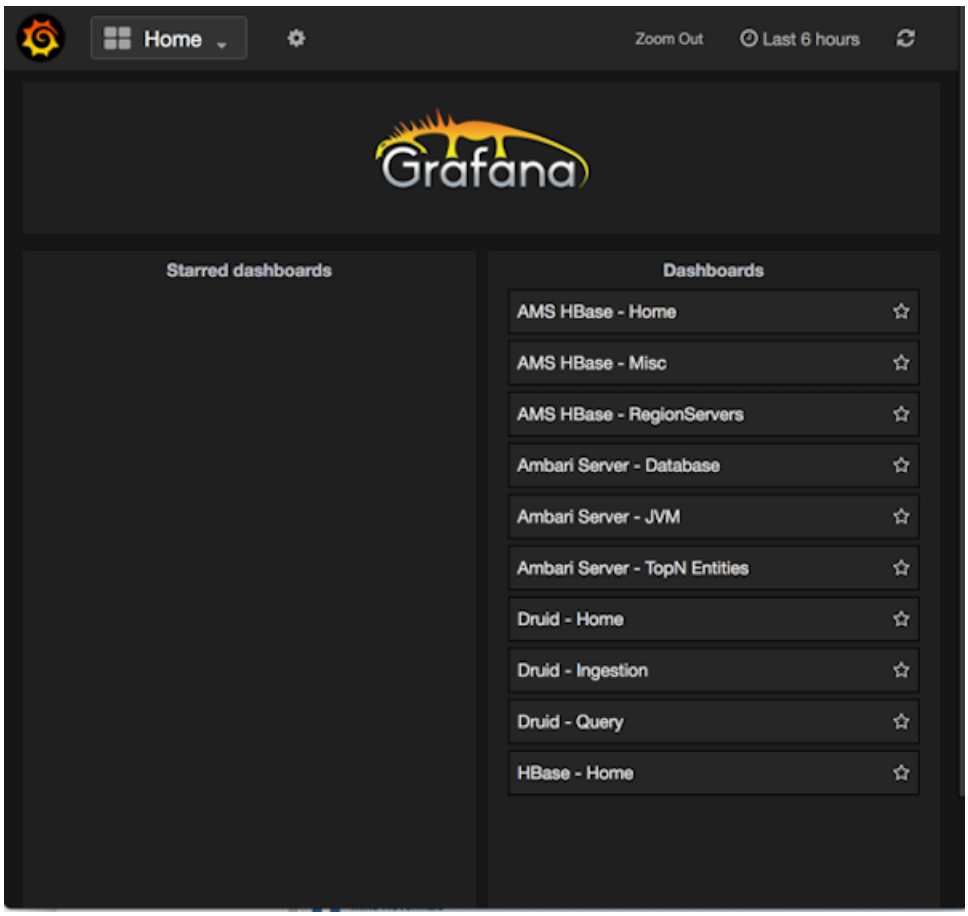


Access Grafana

Use the Grafana user interface to view metrics visualizations.

Procedure

1. In **Ambari Web**, browse to **Services** > **Ambari Metrics** > **Summary**.
2. In **Quick Links**, click **Grafana**.
   A read-only version of the Grafana interface opens in a new tab in your browser.



What to do next

In the Grafana UI, click a link in the **Dashboards** list, or click the **Home** link.

Related Information

http://grafana.org/

View Grafana dashboards

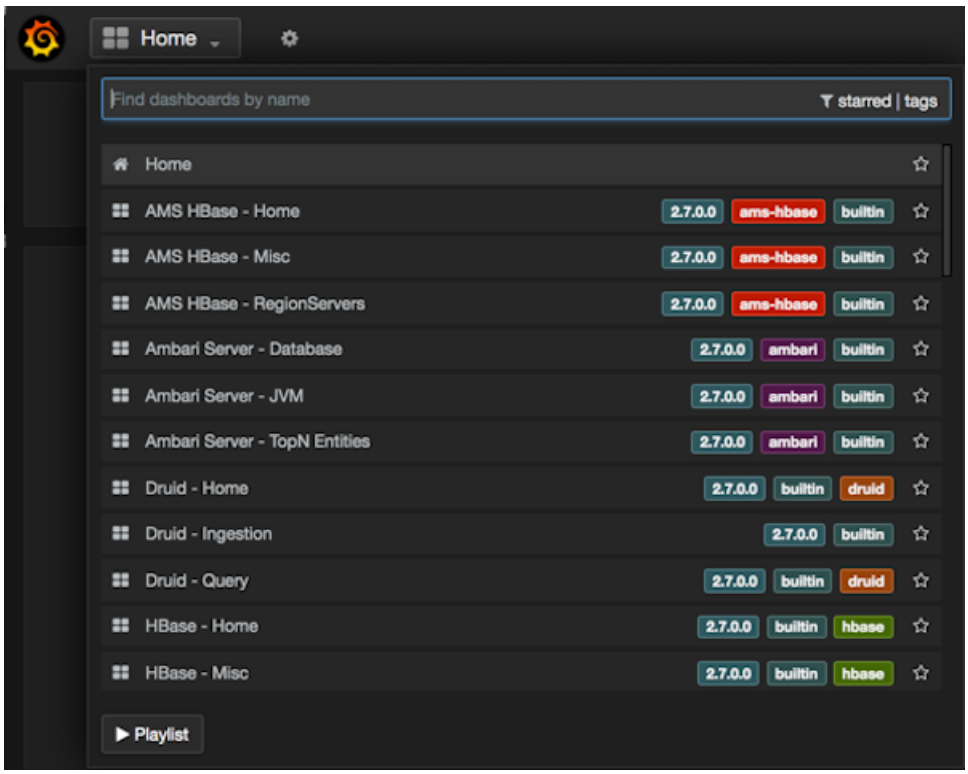
Use **Dashboards** on the Grafana home page to access AMS, Ambari server, Druid and HBase metrics.

About this task

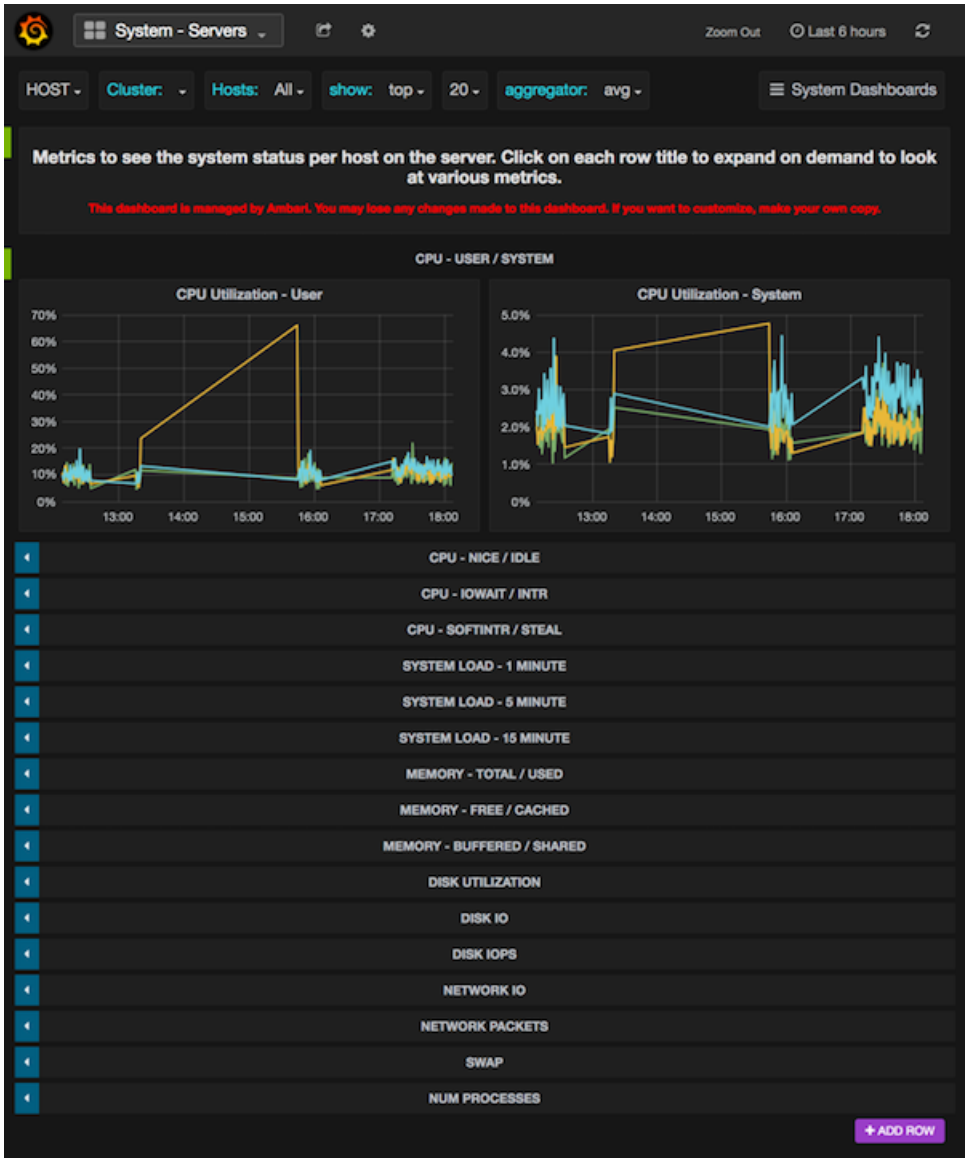
To view specific metrics included in the list:

Procedure

1. In Grafana, browse to **Dashboards**.
2. On **Dashboards** click a dashboard name.
3. To see more available dashboards, click the **Home** list.



4. Scroll down to view all available dashboards.
5. From the list on **Home**, click a dashboard name..
   For example, click **System - Servers**.
   The **System - Servers** dashboard opens.

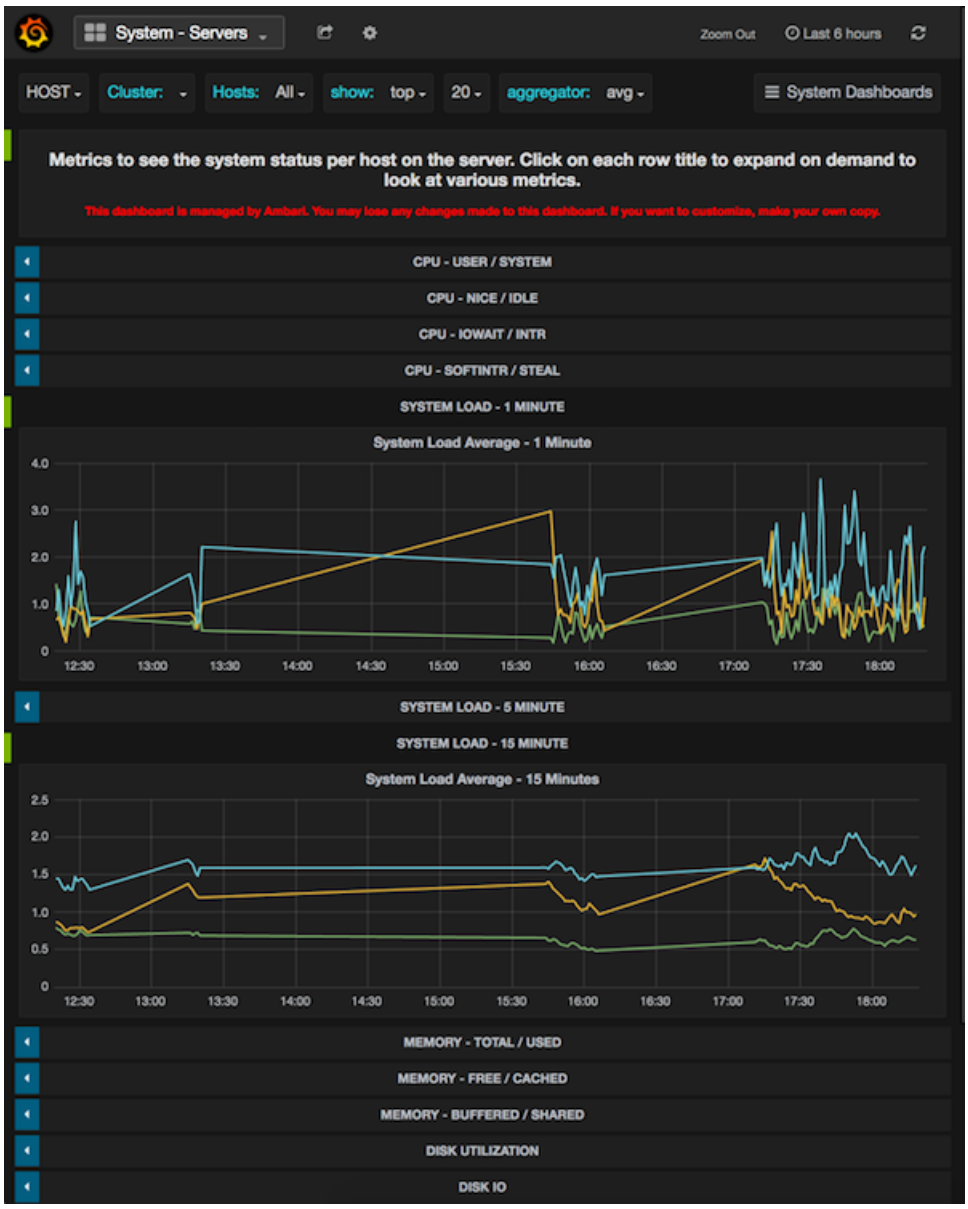


View selected metrics in a Grafana dashboard

Use each Grafana dashboard to visualize multiple metrics

Procedure

- On a dashboard, expand one or more rows to view detailed metrics.
  For example, in the **System - Servers** dashboard, click **System Load Average - 1 Minute**.
  The row expands to display a chart that shows metrics information. This example shows the System Load Average - 1 Minute and the System Load Average - 15 Minute rows expanded. Other rows in the System-Servers dashboard remain collapsed.



View metrics for selected hosts

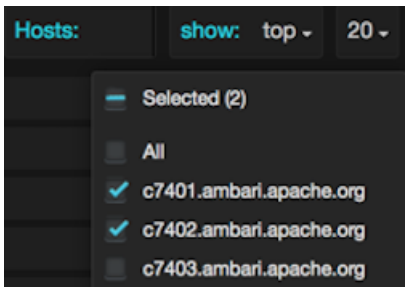
Use **Hosts** to limit the number of hosts for which Grafana dispalys metrics information.

About this task

Grafana shows metrics for all hosts in your cluster by default. You can limit the set of hosts for which metrics display to one or more, by selecting them from the **Hosts** menu.

Procedure

1. On Grafana **Home**, expand **Hosts**.
2. In **Hosts**. select one or more host names.
   A check mark appears next to selected host names.



Results

Selections in the **Hosts** menu apply to all metrics in the current dashboard. Grafana refreshes the current dashboards when you select a new set of hosts.

Grafana dashboards reference

Ambari Metrics System includes Grafana with pre-built dashboards for advanced visualization of cluster metrics.

AMS HBase dashboards

AMS HBase Grafana dashboards track the same metrics as the regular HBase dashboard, but for the AMS-owned instance.

AMS HBase refers to the HBase instance managed by Ambari Metrics Service independently. It does not have any connection with the cluster HBase service.

The following Grafana dashboards are available for AMS HBase:

AMS HBase Home

The AMS HBase Home dashboards display basic statistics about an HBase cluster.

These dashboards provide insight to the overall status for the HBase cluster.

Table 1: AMS HBase Home metrics descriptions

Row	Metrics	Description
REGIONSERVERS / REGIONS	Num RegionServers	Total number of RegionServers in the cluster.
	Num Dead RegionServers	Total number of RegionServers that are dead in the cluster.
	Num Regions	Total number of regions in the cluster.
	Avg Num Regions per RegionServer	Average number of regions per RegionServer.
NUM REGIONS/ STORES	Num Regions / Stores - Total	Total number of regions and stores (column families) in the cluster.
	Store File Size / Count - Total	Total data file size and number of store files.
NUM REQUESTS	Num Requests - Total	Total number of requests (read, write and RPCs) in the cluster.
	Num Request - Breakdown - Total	Total number of get,put,mutate,etc requests in the cluster.
REGIONSERVER MEMORY	RegionServer Memory - Average	Average used, max or committed on-heap and offheap memory for RegionServers.
	RegionServer Offheap Memory - Average	Average used, free or committed on-heap and offheap memory for RegionServers.
MEMORY - MEMSTORE BLOCKCACHE	Memstore - BlockCache - Average	Average blockcache and memstore sizes for RegionServers.
	Num Blocks in BlockCache - Total	Total number of (hfile) blocks in the blockcaches across all RegionServers.
BLOCKCACHE	BlockCache Hit/Miss/s Total	Total number of blockcache hits misses and evictions across all RegionServers.

Row	Metrics	Description
	BlockCache Hit Percent - Average	Average blockcache hit percentage across all RegionServers.
OPERATION LATENCIES - GET/ MUTATE	Get Latencies - Average	Average min, median, max, 75th, 95th, 99th percentile latencies for Get operation across all RegionServers.
	Mutate Latencies - Average	Average min, median, max, 75th, 95th, 99th percentile latencies for Mutate operation across all RegionServers.
OPERATION LATENCIES - DELETE/ INCREMENT	Delete Latencies - Average	Average min, median, max, 75th, 95th, 99th percentile latencies for Delete operation across all RegionServers.
	Increment Latencies - Average	Average min, median, max, 75th, 95th, 99th percentile latencies for Increment operation across all RegionServers.
OPERATION LATENCIES - APPEND/REPLAY	Append Latencies - Average	Average min, median, max, 75th, 95th, 99th percentile latencies for Append operation across all RegionServers.
	Replay Latencies - Average	Average min, median, max, 75th, 95th, 99th percentile latencies for Replay operation across all RegionServers.
REGIONSERVER RPC	RegionServer RPC -Average	Average number of RPCs, active handler threads and open connections across all RegionServers.
	RegionServer RPC Queues - Average	Average number of calls in different RPC scheduling queues and the size of all requests in the RPC queue across all RegionServers.
REGIONSERVER RPC	RegionServer RPC Throughput - Average	Average sent and received bytes from the RPC across all RegionServers.

AMS HBase RegionServers

The AMS HBase RegionServers dashboards display metrics for RegionServers in the monitored HBase cluster, including some performance-related data.

These dashboards help you view basic I/O data and compare load among RegionServers.

Table 2: AMS HBase RegionServers metrics descritptions

Row	Metrics	Description
NUM REGIONS	Num Regions	Number of regions in the RegionServer.
STORE FILES	Store File Size	Total size of the store files (data files) in the RegionServer.
	Store File Count	Total number of store files in the RegionServer.
NUM REQUESTS	Num Total Requests /s	Total number of requests (both read and write) per second in the RegionServer.
	Num Write Requests /s	Total number of write requests per second in the RegionServer.
	Num Read Requests /s	Total number of read requests per second in the RegionServer.
NUM REQUESTS - GET / SCAN	Num Get Requests /s	Total number of Get requests per second in the RegionServer.
	Num Scan Next Requests /s	Total number of Scan requests per second in the RegionServer.
NUM REQUESTS - MUTATE / DELETE	Num Mutate Requests - /s	Total number of Mutate requests per second in the RegionServer.
	Num Delete Requests /s	Total number of Delete requests per second in the RegionServer.
NUM REQUESTS - APPEND / INCREMENT	Num Append Requests /s	Total number of Append requests per second in the RegionServer.
	Num Increment Requests /s	Total number of Increment requests per second in the RegionServer.
	Num Replay Requests /s	Total number of Replay requests per second in the RegionServer.
MEMORY	RegionServer Memory Used	Heap Memory used by the RegionServer.
	RegionServer Offheap Memory Used	Offheap Memory used by the RegionServer.
MEMSTORE	Memstore Size	Total Memstore memory size of the RegionServer.

Row	Metrics	Description
BLOCKCACHE - OVERVIEW	BlockCache - Size	Total BlockCache size of the RegionServer.
	BlockCache - Free Size	Total free space in the BlockCache of the RegionServer.
	Num Blocks in Cache	Total number of hfile blocks in the BlockCache of the RegionServer.
BLOCKCACHE - HITS/MISSES	Num BlockCache Hits /s	Number of BlockCache hits per second in the RegionServer.
	Num BlockCache Misses /s	Number of BlockCache misses per second in the RegionServer.
	Num BlockCache Evictions /s	Number of BlockCache evictions per second in the RegionServer.
	BlockCache Caching Hit Percent	Percentage of BlockCache hits per second for requests that requested cache blocks in the RegionServer.
	BlockCache Hit Percent	Percentage of BlockCache hits per second in the RegionServer.
OPERATION LATENCIES - GET	Get Latencies - Mean	Mean latency for Get operation in the RegionServer.
	Get Latencies - Median	Median latency for Get operation in the RegionServer.
	Get Latencies - 75th Percentile	75th percentile latency for Get operation in the RegionServer
	Get Latencies - 95th Percentile	95th percentile latency for Get operation in the RegionServer.
	Get Latencies - 99th Percentile	99th percentile latency for Get operation in the RegionServer.
	Get Latencies - Max	Max latency for Get operation in the RegionServer.
OPERATION LATENCIES - SCAN NEXT	Scan Next Latencies - Mean	Mean latency for Scan operation in the RegionServer.
	Scan Next Latencies - Median	Median latency for Scan operation in the RegionServer.
	Scan Next Latencies - 75th Percentile	75th percentile latency for Scan operation in the RegionServer.
	Scan Next Latencies - 95th Percentile	95th percentile latency for Scan operation in the RegionServer.
	Scan Next Latencies - 99th Percentile	99th percentile latency for Scan operation in the RegionServer.
	Scan Next Latencies - Max	Max latency for Scan operation in the RegionServer.
OPERATION LATENCIES - MUTATE	Mutate Latencies - Mean	Mean latency for Mutate operation in the RegionServer.
	Mutate Latencies - Median	Median latency for Mutate operation in the RegionServer.
	Mutate Latencies - 75th Percentile	75th percentile latency for Mutate operation in the RegionServer.
	Mutate Latencies - 95th Percentile	95th percentile latency for Mutate operation in the RegionServer.
	Mutate Latencies - 99th Percentile	99th percentile latency for Mutate operation in the RegionServer.
	Mutate Latencies - Max	Max latency for Mutate operation in the RegionServer.
OPERATION LATENCIES - DELETE	Delete Latencies - Mean	Mean latency for Delete operation in the RegionServer.
	Delete Latencies - Median	Median latency for Delete operation in the RegionServer.
	Delete Latencies - 75th Percentile	75th percentile latency for Delete operation in the RegionServer.
	Delete Latencies - 95th Percentile	95th percentile latency for Delete operation in the RegionServer.
	Delete Latencies - 99th Percentile	99th percentile latency for Delete operation in the RegionServer.
	Delete Latencies - Max	Max latency for Delete operation in the RegionServer.
OPERATION LATENCIES - INCREMENT	Increment Latencies - Mean	Mean latency for Increment operation in the RegionServer.
	Increment Latencies - Median	Median latency for Increment operation in the RegionServer.
	Increment Latencies - 75th Percentile	75th percentile latency for Increment operation in the RegionServer.
	Increment Latencies - 95th Percentile	95th percentile latency for Increment operation in the RegionServer.
	Increment Latencies - 99th Percentile	99th percentile latency for Increment operation in the RegionServer.
	Increment Latencies - Max	Max latency for Increment operation in the RegionServer.

Row	Metrics	Description
OPERATION LATENCIES - APPEND	Append Latencies - Mean	Mean latency for Append operation in the RegionServer.
	Append Latencies - Median	Median latency for Append operation in the RegionServer.
	Append Latencies - 75th Percentile	75th percentile latency for Append operation in the RegionServer.
	Append Latencies - 95th Percentile	95th percentile latency for Append operation in the RegionServer.
	Append Latencies - 99th Percentile	99th percentile latency for Append operation in the RegionServer.
	Append Latencies - Max	Max latency for Append operation in the RegionServer.
OPERATION LATENCIES - REPLAY	Replay Latencies - Mean	Mean latency for Replay operation in the RegionServer.
	Replay Latencies - Median	Median latency for Replay operation in the RegionServer.
	Replay Latencies - 75th Percentile	75th percentile latency for Replay operation in the RegionServer.
	Replay Latencies - 95th Percentile	95th percentile latency for Replay operation in the RegionServer.
	Replay Latencies - 99th Percentile	99th percentile latency for Replay operation in the RegionServer.
	Replay Latencies - Max	Max latency for Replay operation in the RegionServer.
RPC - OVERVIEW	Num RPC /s	Number of RPCs per second in the RegionServer.
	Num Active Handler Threads	Number of active RPC handler threads (to process requests) in the RegionServer.
	Num Connections	Number of connections to the RegionServer.
RPC - QUEUES	Num RPC Calls in General Queue	Number of RPC calls in the general processing queue in the RegionServer.
	Num RPC Calls in Priority Queue	Number of RPC calls in the high priority (for system tables) processing queue in the RegionServer.
	Num RPC Calls in Replication Queue	Number of RPC calls in the replication processing queue in the RegionServer.
	RPC - Total Call Queue Size	Total data size of all RPC calls in the RPC queues in the RegionServer.
RPC - CALL QUEUED TIMES	RPC - Call Queued Time - Mean	Mean latency for RPC calls to stay in the RPC queue in the RegionServer.
	RPC - Call Queued Time - Median	Median latency for RPC calls to stay in the RPC queue in the RegionServer.
	RPC - Call Queued Time - 75th Percentile	75th percentile latency for RPC calls to stay in the RPC queue in the RegionServer.
	RPC - Call Queued Time - 95th Percentile	95th percentile latency for RPC calls to stay in the RPC queue in the RegionServer.
	RPC - Call Queued Time - 99th Percentile	99th percentile latency for RPC calls to stay in the RPC queue in the RegionServer.
	RPC - Call Queued Time - Max	Max latency for RPC calls to stay in the RPC queue in the RegionServer.
RPC - CALL PROCESS TIMES	RPC - Call Process Time - Mean	Mean latency for RPC calls to be processed in the RegionServer.
	RPC - Call Process Time - Median	Median latency for RPC calls to be processed in the RegionServer.
	RPC - Call Process Time - 75th Percentile	75th percentile latency for RPC calls to be processed in the RegionServer.
	RPC - Call Process Time - 95th Percentile	95th percentile latency for RPC calls to be processed in the RegionServer.
	RPC - Call Process Time - 99th Percentile	99th percentile latency for RPC calls to be processed in the RegionServer.
	RPC - Call Process Time - Max	Max latency for RPC calls to be processed in the RegionServer.

Row	Metrics	Description
RPC - THROUGHPUT	RPC - Received bytes /s	Received bytes from the RPC in the RegionServer.
	RPC - Sent bytes /s	Sent bytes from the RPC in the RegionServer.
WAL - FILES	Num WAL - Files	Number of Write-Ahead-Log files in the RegionServer.
	Total WAL File Size	Total files sized of Write-Ahead-Logs in the RegionServer.
WAL - THROUGHPUT	WAL - Num Appends /s	Number of append operations per second to the filesystem in the RegionServer.
	WAL - Num Sync /s	Number of sync operations per second to the filesystem in the RegionServer.
WAL - SYNC LATENCIES	WAL - Sync Latencies - Mean	Mean latency for Write-Ahead-Log sync operation to the filesystem in the RegionServer.
	WAL - Sync Latencies - Median	Median latency for Write-Ahead-Log sync operation to the filesystem in the RegionServer.
	WAL - Sync Latencies - 75th Percentile	75th percentile latency for Write-Ahead-Log sync operation to the filesystem in the RegionServer.
	WAL - Sync Latencies - 95th Percentile	95th percentile latency for Write-Ahead-Log sync operation to the filesystem in the RegionServer.
	WAL - Sync Latencies - 99th Percentile	99th percentile latency for Write-Ahead-Log sync operation to the filesystem in the RegionServer.
	WAL - Sync Latencies - Max	Max latency for Write-Ahead-Log sync operation to the filesystem in the RegionServer.
WAL - APPEND LATENCIES	WAL - Append Latencies - Mean	Mean latency for Write-Ahead-Log append operation to the filesystem in the RegionServer.
	WAL - Append Latencies - Median	Median latency for Write-Ahead-Log append operation to the filesystem in the RegionServer.
	WAL - Append Latencies - 75th Percentile	95th percentile latency for Write-Ahead-Log append operation to the filesystem in the RegionServer.
	WAL - Append Latencies - 95th Percentile	95th percentile latency for Write-Ahead-Log append operation to the filesystem in the RegionServer.
	WAL - Append Latencies - 99th Percentile	99th percentile latency for Write-Ahead-Log append operation to the filesystem in the RegionServer.
	WAL - Append Latencies - Max	Max latency for Write-Ahead-Log append operation to the filesystem in the RegionServer.
WAL - APPEND SIZES	WAL - Append Sizes - Mean	Mean data size for Write-Ahead-Log append operation to the filesystem in the RegionServer.
	WAL - Append Sizes - Median	Median data size for Write-Ahead-Log append operation to the filesystem in the RegionServer.
	WAL - Append Sizes - 75th Percentile	75th percentile data size for Write-Ahead-Log append operation to the filesystem in the RegionServer.
	WAL - Append Sizes - 95th Percentile	95th percentile data size for Write-Ahead-Log append operation to the filesystem in the RegionServer.
	WAL - Append Sizes - 99th Percentile	99th percentile data size for Write-Ahead-Log append operation to the filesystem in the RegionServer.
	WAL - Append Sizes - Max	Max data size for Write-Ahead-Log append operation to the filesystem in the RegionServer.
SLOW OPERATIONS	WAL Num Slow Append /s	Number of append operations per second to the filesystem that took more than 1 second in the RegionServer.
	Num Slow Gets /s	Number of Get requests per second that took more than 1 second in the RegionServer.

Row	Metrics	Description
	Num Slow Puts /s	Number of Put requests per second that took more than 1 second in the RegionServer.
	Num Slow Deletes /s	Number of Delete requests per second that took more than 1 second in the RegionServer.
FLUSH/ COMPACTION QUEUES	Flush Queue Length	Number of Flush operations waiting to be processed in the RegionServer. A higher number indicates flush operations being slow.
	Compaction Queue Length	Number of Compaction operations waiting to be processed in the RegionServer. A higher number indicates compaction operations being slow.
	Split Queue Length	Number of Region Split operations waiting to be processed in the RegionServer. A higher number indicates split operations being slow.
JVM - GC COUNTS	GC Count /s	Number of Java Garbage Collections per second.
	GC Count ParNew /s	Number of Java ParNew (YoungGen) Garbage Collections per second.
	GC Count CMS /s	Number of Java CMS Garbage Collections per second.
JVM - GC TIMES	GC Times /s	Total time spend in Java Garbage Collections per second.
	GC Times ParNew /s	Total time spend in Java ParNew(YoungGen) Garbage Collections per second.
	GC Times CMS /s	Total time spend in Java CMS Garbage Collections per second.
LOCALITY	Percent Files Local	Percentage of files served from the local DataNode for the RegionServer.

AMS HBase Misc

The AMS HBase Misc dashboards display miscellaneous metrics related to the HBase cluster.

You can use these metrics for tasks like debugging authentication and authorization issues and exceptions raised by RegionServers.

Table 3: AMS HBase Misc metrics descriptions

Row	Metrics	Description
REGIONS IN TRANSITION	Master - Regions in Transition	Number of regions in transition in the cluster.
	Master - Regions in Transition Longer Than Threshold Time	Number of regions in transition that are in transition state for longer than 1 minute in the cluster.
	Regions in Transition Oldest Age	Maximum time that a region stayed in transition state.
NUM THREADS - RUNNABLE	Master Num Threads - Runnable	Number of threads in the Master.
	RegionServer Num Threads - Runnable	Number of threads in the RegionServer.
NUM THREADS - BLOCKED	Master Num Threads - Blocked	Number of threads in the Blocked State in the Master.
	RegionServer Num Threads - Blocked	Number of threads in the Blocked State in the RegionServer.
NUM THREADS - WAITING	Master Num Threads - Waiting	Number of threads in the Waiting State in the Master.
	RegionServer Num Threads - Waiting	Number of threads in the Waiting State in the RegionServer.
NUM THREADS - TIMED WAITING	Master Num Threads - Timed Waiting	Number of threads in the Timed-Waiting State in the Master.
	RegionServer Num Threads - Timed Waiting	Number of threads in the Timed-Waiting State in the RegionServer.
NUM THREADS - NEW	Master Num Threads - New	Number of threads in the New State in the Master.
	RegionServer Num Threads - New	Number of threads in the New State in the RegionServer.
NUM THREADS - TERMINATED	Master Num Threads - Terminated	Number of threads in the Terminated State in the Master.
	RegionServer Num Threads - Terminated	Number of threads in the Terminated State in the RegionServer.

Row	Metrics	Description
RPC AUTHENTICATION	RegionServer RPC Authentication Successes / s	Number of RPC successful authentications per second in the RegionServer.
	RegionServer RPC Authentication Failures /s	Number of RPC failed authentications per second in the RegionServer.
RPC Authorization	RegionServer RPC Authorization Successes /s	Number of RPC successful autorizations per second in the RegionServer.
	RegionServer RPC Authorization Failures /s	Number of RPC failed autorizations per second in the RegionServer.
EXCEPTIONS	Master Exceptions /s	Number of exceptions in the Master.
	RegionServer Exceptions /s	Number of exceptions in the RegionServer.

Ambari dashboards

The following Grafana dashboards are available for Ambari:

Ambari Server database

Metrics that show operating status for the Ambari server database.

Table 4: Ambari Server database metrics descriptions

Row	Metrics	Description
TOTAL READ ALL QUERY	Total Read All Query Counter (Rate)	Total ReadAllQuery operations performed.
	Total Read All Query Timer (Rate)	Total time spent on ReadAllQuery.
TOTAL CACHE HITS & MISSES	Total Cache Hits (Rate)	Total cache hits on Ambari Server with respect to EclipseLink cache.
	Total Cache Misses (Rate)	Total cache misses on Ambari Server with respect to EclipseLink cache.
QUERY	Query Stages Timings	Average time spent on every query sub stage by Ambari Server
	Query Types Avg. Timings	Average time spent on every query type by Ambari Server.
HOST ROLE COMMAND ENTITY	Counter.ReadAllQuery.HostRoleCommandEntity (Rate)	Rate (num operations per second) in which ReadAllQuery operation on HostRoleCommandEntity is performed.
	Timer.ReadAllQuery.HostRoleCommandEntity (Rate)	Rate in which ReadAllQuery operation on HostRoleCommandEntity is performed.
	ReadAllQuery.HostRoleCommandEntity	Average time taken for a ReadAllQuery operation on HostRoleCommandEntity (Timer / Counter).

Ambari Server JVM

Metrics to see status for the Ambari Server Java virtual machine.

Table 5: Ambari Server JVM metrics descriptions

Row	Metrics	Description
JVM - MEMORY PRESSURE	Heap Usage	Used, max or committed on-heap memory for Ambari Server.
	Off-Heap Usage	Used, max or committed off-heap memory for Ambari Server.
JVM GC COUNT	GC Count Par new /s	Number of Java ParNew (YoungGen) Garbage Collections per second.
	GC Time Par new /s	Total time spend in Java ParNew(YoungGen) Garbage Collections per second.
	GC Count CMS /s	Number of Java Garbage Collections per second.
	GC Time Par CMS /s	Total time spend in Java CMS Garbage Collections per second.
JVM THREAD COUNT	Thread Count	Number of active, daemon, deadlock, blocked and runnable threads.

Ambari Server top n

Metrics to see top performing users and operations for Ambari.

Table 6: Ambari Server top n metrics descriptions

Row	Metrics	Description
READ ALL QUERY	Top ReadAllQuery Counters	Top N Ambari Server entities by number of ReadAllQuery operations performed.
	Top ReadAllQuery Timers	Top N Ambari Server entities by time spent on ReadAllQuery operations.
CACHE MISSES	Cache Misses	Top N Ambari Server entities by number of Cache Misses.

Druid Dashboards

The following Grafana dashboards are available for Druid:

Druid Home

Metrics that show operating status for Druid.

Table 7: Druid home metrics descriptions

Row	Metrics	Description
DRUID BROKER	JVM Heap	JVM Heap used by the Druid Broker Node.
	JVM GCM Time	Time spent by the Druid Broker Node in JVM Garbage collection.
DRUID HISTORICAL	JVM Heap	JVM Heap used by the Druid Historical Node.
	JVM GCM Time	Time spent by the Druid Historical Node in JVM Garbage collection.
DRUID COORDINATER	JVM Heap	JVM Heap used by the Druid Coordinator Node.
	JVM GCM Time	Time spent by the Druid Coordinator Node in JVM Garbage collection.
DRUID OVERLORD	JVM Heap	JVM Heap used by the Druid Overlord Node.
	JVM GCM Time	Time spent by the Druid Overlord Node in JVM Garbage collection.
DRUID MIDDLEMANAGER	JVM Heap	JVM Heap used by the Druid Middlemanager Node.
	JVM GCM Time	Time spent by the Druid Middlemanager Node in JVM Garbage collection.

Druid Ingestion

Metrics to see status for Druid data ingestion rates.

Table 8: Druid Ingestion metrics descriptions

Row	Metrics	Description
INGESTION METRICS	Ingested Events	Number of events ingested on real time nodes.
	Events Thrown Away	Number of events rejected because they are outside the windowPeriod.
	Unparseable Events	Number of events rejected because they did not parse.
INTERMEDIATE PERSISTS METRICS	Persisted Rows	Number of Druid rows persisted on disk.
	Average Persist Time	Average time taken to persist intermediate segments to disk.
	Intermediate Persist Count	Number of times that intermediate segments were persisted.
SEGMENT SIZE METRICS	Ave Segment Size	Average size of added Druid segments.
	Total Segment Size	Total size of added Druid segments.

Druid Query

Metrics to see status of Druid queries.

Table 9: Druid Query metrics descriptions

Row	Metrics	Description
QUERY TIME METRICS	Broker Query Time	Average Time taken by Druid Broker node to process queries.
	Historical Query Time	Average time taken by Druid historical nodes to process queries.
	Realtime Query Time	Average time taken by Druid real time nodes to process queries.
SEGMENT SCAN METRICS	Historical Segment Scan Time	Average time taken by Druid historical nodes to scan individual segments.
	Realtime Segment Scan Time	Average time taken by Druid real time nodes to scan individual segments.
	Historical Query Wait Time	Average time spent waiting for a segment to be scanned on historical node.
	Realtime Query Wait Time	Average time spent waiting for a segment to be scanned on real time node.
	Pending Historical Segment Scans	Average Number of pending segment scans on historical nodes.
	Pending Realtime Segment Scans	Average Number of pending segment scans on real time nodes.

HDFS Dashboards

The following Grafana dashboards are available for Hadoop Distributed File System (HDFS) components:

HDFS Home

The HDFS - Home dashboard displays metrics that show operating status for HDFS.

In a NameNode HA setup, metrics are collected from and displayed for both the active and the standby NameNode.

Table 10: HDFS Home metrics descriptions

Row	Metrics	Description
NUMBER OF FILES UNDER CONSTRUCTION & RPC CLIENT CONNECTIONS	Number of Files Under Construction	Number of HDFS files that are still being written.
	PC Client Connections	Number of open RPC connections from clients on NameNode(s).
TOTAL FILE OPERATIONS & CAPACITY USED	Total File Operations	Total number of operations on HDFS files, including file creation/ deletion/rename/truncation, directory/file/block information retrieval, and snapshot related operations.
	Capacity Used	CapacityTotalGB shows total HDFS storage capacity, in GB. CapacityUsedGB indicates total used HDFS storage capacity, in GB.
RPC CLIENT PORT SLOW CALLS & HDFS TOTAL LOAD	RPC Client Port Slow Calls	Number of slow RPC requests on NameNode. A "slow" RPC request is one that takes more time to complete than 99.7% of other requests.
	HDFS Total Load	Total number of connections on all the DataNodes sending/receiving data.
ADD BLOCK STATUS	Add Block Time	The average time (in ms) serving addBlock RPC request on NameNode(s).
	Add Block Num Ops	The rate of addBlock RPC requests on NameNode(s).

HDFS NameNodes

Metrics to see status for the NameNodes.

Table 11: HDFS NameNodes metrics descriptions

Row	Metrics	Description
RPC CLIENT QUEUE TIME	RPC Client Port Queue Time	Average time that a RPC request (on the RPC port facing to the HDFS clients) waits in the queue.
	RPC Client Port Queue Num Ops	Total number of RPC requests in the client port queue.
RPC CLIENT PORT PROCESSING TIME	RPC Client Port Processing Time	Average RPC request processing time in milliseconds, on the client port.
	RPC Client Port Processing Num Ops	Total number of RPC active requests through the client port.
GC COUNT & GC TIME	GC Count	Shows the JVM garbage collection rate on the NameNode.
	GC Time	Shows the garbage collection time in milliseconds.
GC PAR NEW	GC Count Par New	The number of times young generation garbage collection happened.
	GC Time Par New	Indicates the duration of young generation garbage collection.
GC EXTRA SLEEP & WARNING THRESHOLD EXCEEDED	GC Extra Sleep Time	Indicates total garbage collection extra sleep time.
	GC Warning Threshold Exceeded Count	Indicates number of times that the garbage collection warning threshold is exceeded.
RPC CLIENT PORT QUEUE & BACKOFF	RPC Client Port Queue Length	Indicates the current length of the RPC call queue.
	RPC Client Port Backoff	Indicates number of client backoff requests.
RPC SERVICE PORT QUEUE & NUM OPS	RPC Service Port Queue Time	Average time a RPC request waiting in the queue, in milliseconds. These requests are on the RPC port facing to the HDFS services, including DataNodes and the other NameNode.
	RPC Service Port Queue Num Ops	Total number of RPC requests waiting in the queue. These requests are on the RPC port facing to the HDFS services, including DataNodes and the other NameNode.
RPC SERVICE PORT PROCESSING TIME & NUM OPS	RPC Service Port Processing Time	Average RPC request processing time in milliseconds, for the service port.
	RPC Service Port Processing Num Ops	Number of RPC requests processed for the service port.
RPC SERVICE PORT CALL QUEUE LENGTH & SLOW CALLS	RPC Service Port Call Queue Length	The current length of the RPC call queue.
	RPC Service Port Slow Calls	The number of slow RPC requests, for the service port.
TRANSACTIONS SINCE LAST EDIT & CHECKPOINT	Transactions Since Last Edit Roll	Total number of transactions since the last editlog segment.
	Transactions Since Last Checkpoint	Total number of transactions since the last editlog segment checkpoint.
LOCK QUEUE LENGTH & EXPIRED HEARTBEATS	Lock Queue Length	Shows the length of the wait Queue for the FSNameSystemLock.
	Expired Heartbeats	Indicates the number of times expired heartbeats are detected on NameNode.
THREADS BLOCKED / WAITING	Threads Blocked	Indicates the number of threads in a BLOCKED state, which means they are waiting for a lock.
	Threads Waiting	Indicates the number of threads in a WAITING state, which means they are waiting for another thread to perform an action.

HDFS DataNodes

Metrics to see status for the DataNodes.

Table 12: HDFS DataNodes metrics descriptions

Row	Metrics	Description
BLOCKS WRITTEN / READ	Blocks Written	The rate or number of blocks written to a DataNode.
	Blocks Read	The rate or number of blocks read from a DataNode.
FSYNCH TIME / NUM OPS	Fsynch Time	Average fsync time.
	Fsynch Num Ops	Total number of fsync operations.
DATA PACKETS BLOCKED / NUM OPS	Data Packet Blocked Time	Indicates the average waiting time of transfering a data packet on a DataNode.
	Data Packet Blocked Num Ops	Indicates the number of data packets transferred on a DataNode.
PACKET TRANSFER BLOCKED / NUM OPS	Packet Transfer Time	Average transfer time of sending data packets on a DataNode.
	Packet Transfer Num Ops	Indicates the number of data packets blocked on a DataNode.
NETWORK ERRORS / GC COUNT	Network Errors	Rate of network errors on JVM.
	GC Count	Garbage collection DataNode hits.
GC TIME / GC TIME PARNEW	GC Time	JVM garbage collection time on a DataNode.
	GC Time ParNew	Young generation (ParNew) garbage collection time on a DataNode.

HDFS top n

Metrics that show top-level usage in HDFS.

Metrics that show

- Which users perform most HDFS operations on the cluster
- Which HDFS operations run most often on the cluster.

Table 13: HDFS top n metrics descriptions

Row	Metrics	Description
TOP N - OPERATIONS COUNT	Top N Total Operations Count 1 min sliding window	Represents the metrics that show the total operation count per operation for all users. Shown for 1-minute interval.
	Top N Total Operations Count 5 min sliding window	Represents the metrics that show the total operation count per operation for all users. Shown for 5-minute interval.
	Top N Total Operations Count 25 min sliding window	Represents the metrics that show the total operation count per operation for all users. Shown for 25-minute interval.
TOP N - TOTAL OPERATIONS COUNT BY USER	Top N Total Operations Count by User 1 min sliding window	Represents the metrics that show the total operation count per user. Shown for 1-minute intervals.
	Top N Total Operations Count by User 5 min sliding window	Represents the metrics that show the total operation count per user. Shown for 5-minute intervals.
	Top N Total Operations Count by User 25 min sliding window	Represents the metrics that show the total operation count per user. Shown for 25-minute intervals.

Row	Metrics	Description
TOP N - OPERATIONS BY USER	TOP N - Operations by User 1 min sliding window	Represents the drilled down User x Op metrics against the TotalCount Shown for 1-minute intervals.
	TOP N - Operations by User 5 min sliding window	Represents the drilled down User x Op metrics against the TotalCount. Shown for 5-minute intervals.
	TOP N - Operations by User 25 min sliding window	Represents the drilled down User x Op metrics against the TotalCount. Shown for 25-minute intervals.

HDFS Users

Metrics to see status for HDFS users.

Table 14: HDFS Users metrics descriptions

Row	Metrics	Description
Namenode Rpc Caller Volume	Namenode Rpc Caller Volume	Number of RPC calls made by top(10) users.
Namenode Rpc Caller Priority	Namenode Rpc Caller Priority	Priority assignment for incoming calls from top(10) users.

YARN Dashboards

The following Grafana dashboards are available for YARN:

YARN Home

Metrics to see the overall status for the YARN cluster.

Table 15: YARN Home metrics descriptions

Metrics	Description
Nodes	The number of (active, unhealthy, lost) nodes in the cluster.
Apps	The number of (running, pending, completed, failed) apps in the cluster.
Cluster Memory Available	Total available memory in the cluster.

YARN Applications

Metrics to see status of Applications on the YARN Cluster.

Table 16: YARN Applications metrices descriptions

Metrics	Description
Applications By Running Time	Number of apps by running time in 4 categories by default (< 1 hour, 1 ~ 5 hours, 5 ~ 24 hours, > 24 hours).
Apps Running vs Pending	The number of running apps vs the number of pending apps in the cluster.
Apps Submitted vs Completed	The number of submitted apps vs the number of completed apps in the cluster.
Avg AM Launch Delay	The average time taken from allocating an AM container to launching an AM container.
Avg AM Register Delay	The average time taken from RM launches an AM container to AM registers back with RM.

YARN MR JobHistory Server

Metrics to see status of the Job History Server.

Table 17: YARN MR JobHistory Server metrics descriptions

Row	Metrics	Description
JVM METRICS	GC Count	Accumulated GC count over time.
	GC Time	Accumulated GC time over time.
	Heap Mem Usage	Current heap memory usage.
	NonHeap Mem Usage	Current non-heap memory usage.

YARN NodeManagers

Metrics to see status of YARN NodeManagers on the YARN cluster.

Table 18: YARN NodeManagers metrics descriptions

Row	Metrics	Description
NUM CONTAINERS	Containers Running	Current number of running containers.
	Containers Failed	Accumulated number of failed containers.
	Containers Killed	Accumulated number of killed containers.
	Containers Completed	Accumulated number of completed containers.
MEMORY UTILIZATION	Memory Available	Available memory for allocating containers on this node.
	Used Memory	Used memory by containers on this node.
DISK UTILIZATION	Disk Utilization for Good Log Dirs	Disk utilization percentage across all good log directories.
	Disk Utilization for Good Local Dirs	Disk utilization percentage across all good local directories.
	Bad Log Dirs	Number of bad log directories.
	Bad Local Dirs	Number of bad local directories.
AVE CONTAINER LAUNCH DELAY	Ave Container Launch Delay	Average time taken for a NM to launch a container.
RPC METRICS	RPC Avg Processing Time	Average time for processing a RPC call.
	RPC Avg Queue Time	Average time for queuing a PRC call.
	RPC Call Queue Length	The length of the RPC call queue.
	RPC Slow Calls	Number of slow RPC calls.
JVM METRICS	Heap Mem Usage	Current heap memory usage.
	NonHeap Mem Usage	Current non-heap memory usage.
	GC Count	Accumulated GC count over time.
	GC Time	Accumulated GC time over time.
LOG4J METRICS	LOG ERROR	Number of ERROR logs.
	LOG FATAL	Number of FATAL logs.

YARN Queues

Metrics to see status of Queues on the YARN cluster.

Table 19: YARN Queues metrics descriptions

Row	Metrics	Description
NUM APPS	Apps Runnning	Current number of running applications.
	Apps Pending	Current number of pending applications.
	Apps Completed	Accumulated number of completed applications over time.

Row	Metrics	Description
	Apps Failed	Accumulated number of failed applications over time.
	Apps Killed	Accumulated number of killed applications over time.
	Apps Submitted	Accumulated number of submitted applications over time.
NUM CONTAINERS	Containers Running	Current number of running containers.
	Containers Pending	Current number of pending containers.
	Containers Reserved	Current number of Reserved containers.
	Total Containers Allocated	Accumulated number of containers allocated over time.
	Total Node Local Containers Allocated	Accumulated number of node-local containers allocated over time.
	Total Rack Local Containers Allocated	Accumulated number of rack-local containers allocated over time.
	Total OffSwitch Containers Allocated	Accumulated number of off-switch containers allocated over time.
MEMORY UTILIZATION	Allocated Memory	Current amount of memory allocated for containers.
	Pending Memory	Current amount of memory asked by applications for allocating containers.
	Available Memory	Current amount of memory available for allocating containers.
	Reserved Memory	Current amount of memory reserved for containers.
	Memory Used by AM	Current amount of memory used by AM containers.
CONTAINER ALLOCATION DELAY	Ave AM Container Allocation Delay	Average time taken to allocate an AM container since the AM container is requested.

YARN ResourceManager

Metrics to see status of ResourceManagers on the YARN cluster.

Table 20: YARN ResourceManager metrics descriptions

Row	Metrics	Description
RPC STATS	RPC Avg Processing / Queue Time	Average time for processing/queuing a RPC call.
	RPC Call Queue Length	The length of the RPC call queue.
	RPC Slow calls	Number of slow RPC calls.
MEMORY USAGE	Heap Mem Usage	Current heap memory usage.
	NonHeap Mem Usage	Current non-heap memory usage.
GC STATS	GC count	Accumulated GC count over time.
	GcTime	Accumulated GC time over time.
LOG ERRORS	Log Error / Fatal	Number of ERROR/FATAL logs.
AUTHORIZATION & AUTHENTICATION FAILURES	RPC Authorization Failures	Number of authorization failures.
	RPC Authentication Failures	Number of authentication failures.

YARN TimelineServer

Metrics to see the overall status for TimelineServer.

Table 21: YARN Timeline Server metrics descriptions

Row	Metrics	Description
DATA READS	Timeline Entity Data Reads	Accumulated number of read operations.

Row	Metrics	Description
	Timeline Entity Data Read time	Average time for reading a timeline entity.
DATA WRITES	Timeline Entity Data Write	Accumulated number of write operations.
	Timeline Entity Data Write Time	Average time for writing a timeline entity.
JVM METRICS	GC Count	Accumulated GC count over time.
	GC Time	Accumulated GC time over time.
	Heap Usage	Current heap memory usage.
	NonHeap Usage	Current non-heap memory usage.

Hive Dashboards

The following Grafana dashboards are available for Hive:

Hive Home

Metrics that show the overall status for Hive service.

Table 22: Hive Home metrics descriptions

Row	Metrics	Description
WAREHOUSE SIZE - AT A GLANCE	DB count at startup	Number of databases present at the last warehouse service startup time.
	Table count at startup	Number of tables present at the last warehouse service startup time.
	Partition count at startup	Number of partitions present at the last warehouse service startup time.
WAREHOUSE SIZE - REALTIME GROWTH	#tables created (ongoing)	Number of tables created since the last warehouse service startup.
	#partitions created (ongoing)	Number of partitions created since the last warehouse service startup.
MEMORY PRESSURE	HiveMetaStore Memory - Max	Heap memory usage by Hive MetaStores. If applicable, indicates max usage across multiple instances.
	HiveServer2 Memory - Max	Heap memory usage by HiveServer2. If applicable, indicates max usage across multiple instances.
	HiveMetaStore Offheap Memory - Max	Non-heap memory usage by Hive MetaStores. If applicable, indicates max usage across multiple instances.
	HiveServer2 Offheap Memory - Max	Non-heap memory usage by HiveServer2. If applicable, indicates max across multiple instances.
	HiveMetaStore app stop times (due to GC stops)	Total time spent in application pauses caused by garbage collection across Hive MetaStores.
	HiveServer2 app stop times (due to GC stops)	Total time spent in application pauses caused by garbage collection across HiveServer2.
METASTORE - CALL TIMES	API call times - Health Check roundtrip (get_all_databases)	Time taken to process a low-cost call made by health checks to all metastores.
	API call times - Moderate size call (get_partitions_by_names)	Time taken to process a moderate-cost call made by queries/exports/ etc to all metastores. Data for this metric may not be available in a less active warehouse.

Hive HiveMetaStore

Metrics that show operating status for HiveMetaStore hosts.

Select a HiveMetaStore and a host to view relevant metrics.

Table 23: Hive HiveMetaStore metrics descriptions

Row	Metrics	Description
API TIMES	API call times - Health Check roundtrip (get_all_databases)	Time taken to process a low-cost call made by health checks to this metastore.
	API call times - Moderate size call (get_partitions_by_names)	Time taken to process a moderate-cost call made by queries/exports/ etc to this metastore. Data for this metric may not be available in a less active warehouse.
MEMORY PRESSURE	App Stop times (due to GC)	Time spent in application pauses caused by garbage collection.
	Heap Usage	Current heap memory usage.
	Off-Heap Usage	Current non-heap memory usage.

Hive HiveServer2

Metrics that show operating status for HiveServer2 hosts.

Select a HiveServer2 and a host to view relevant metrics.

Table 24: Hive HiveServer2 metrics descriptions

Row	Metrics	Description
API TIMES	API call times - Health Check roundtrip (get_all_databases)	Time taken to process a low-cost cal made by health checks to the metastore embedded in this HiveServer2. Data for this metric may not be available if HiverServer2 is not running in an embedded-metastore mode.
	API call times - Moderate size call (get_partitions_by_names)	Time taken to process a moderate-cost call made by queries/exports/ etc to the metastore embedded in this HiveServer2. Data for this metric may not be available in a less active warehouse, or if HiveServer2 is not running in an embedded-metastore mode.
MEMORY PRESSURE	App Stop times (due to GC)	Time spent in application pauses caused by garbage collection.
	Heap Usage	Current heap memory usage.
	Off-Heap Usage	Current non-heap memory usage.
THREAD STATES	Active operation count	Current number of active operations in HiveServer2 and their running states.
	Completed operation states	Number of completed operations on HiveServer2 since the last restart. Indicates whether they completed as expected or encountered errors.

Hive LLAP Dashboards

The following Grafana dashboards are available for Apache Hive LLAP.

The following Grafana dashboards are available for Apache Hive LLAP. The LLAP Heat map dashboard and the LLAP Overview dashboard enable you to quickly see the hotspots among the LLAP daemons. If you find an issue and want to navigate to more specific information for a specific system, use the LLAP Daemon dashboard.

Note that all Hive LLAP dashboards show the state of the cluster and are useful for looking at cluster information from the previous hour or day. The dashboards do not show real-time results.

Hive LLAP Heatmap

This dashboard enables you to identify the hotspots in the cluster in terms of executors and cache.

The heat map dashboard shows all the nodes that are running LLAP daemons and includes a percentage summary for available executors and cache. The values in the table are color coded based on threshold: if the threshold is more than 50%, the color is green; between 20% and 50%, the color is yellow; and less than 20%, the color is red.

Table 25: Hive LLAP Heatmap metrics descriptions

Row	Metrics	Description
HEAT MAPS	Remaining Cache Capacity	Shows the percentage of cache capacity remaining across the nodes. For example, if the grid is green, the cache is being under utilized. If the grid is red, there is high utilization of cache.
	Remaining Cache Capacity	Same as above (Remaining Cache Capacity), but shows the cache hit ratio.
	Executor Free Slots	Shows the percentage of executor free slots that are available on each nodes.

Hive LLAP Overview

The overview dashboard shows the aggregated information across all of the clusters: for example, the total cache memory from all the nodes.

This dashboard enables you to see that your cluster is configured and running correctly. For example, you might have configured 10 nodes but you see only 8 nodes running. If you find an issue by viewing this dashboard, you can open the LLAP Daemon dashboard to see which node is having the problem.

Table 26: HIVE LLAP Overview metrics descriptions

Row	Metrics	Description
OVERVIEW	Total Executor Threads	Shows the total number of executors across all nodes.
	Total Executor Memory	Shows the total amount of memory for executors across all nodes.
	Total Cache Memory	Shows the total amount of memory for cache across all nodes.
	Total JVM Memory	Shows the total amount of max Java Virtual Machine (JVM) memory across all nodes.
CACHE METRICS ACROSS ALL NODES	Total Cache Usage	Shows the total amount of cache usage (Total, Remaining, and Used) across all nodes.
	Average Cache Hit Rate	As the data is released from the cache, the curve should increase. For example, the first query should run at 0, the second at 80-90 seconds, and then the third 10% faster. If, instead, it decreases, there might be a problem in the cluster.
	Average Cache Read Requests	Shows how many requests are being made for the cache and how many queries you are able to run that make use of the cache. If it says 0, for example, your cache might not be working properly and this grid might reveal a configuration issue.
CACHE METRICS ACROSS ALL NODES	Total Cache Usage	Shows the total amount of cache usage (Total, Remaining, and Used) across all nodes.
	Average Cache Hit Rate	As the data is released from the cache, the curve should increase. For example, the first query should run at 0, the second at 80-90 seconds, and then the third 10% faster. If, instead, it decreases, there might be a problem in the cluster.
	Average Cache Read Requests	Shows how many requests are being made for the cache and how many queries you are able to run that make use of the cache. If it says 0, for example, your cache might not be working properly and this grid might reveal a configuration issue.

Row	Metrics	Description
EXECUTOR METRICS ACROSS ALL NODES	Total Executor Requests	Shows the total number of task requests that were handled, succeeded, failed, killed, evicted and rejected across all nodes. Handled: Total requests across all sub-groups Succeeded: Total requests that were processed. For example, if you have 8 core machines, the number of total executor requests would be 8 Failed: Did not complete successfully because, for example, you ran out of memory Rejected: If all task priorities are the same, but there are still not enough slots to fulfill the request, the system will reject some tasks Evicted: Lower priority requests are evicted if the slots are filled by higher priority requests
	Total Execution Slots	Shows the total execution slots, the number of free or available slots, and number of slots occupied in the wait queue across all nodes. Ideally, the threads available (blue) result should be the same as the threads that are occupied in the queue result.
	Time to Kill Pre-empted Task (300s interval)	Shows the time that it took to kill a query due to pre-emption in percentile (50th, 90th, 99th) latencies in 300 second intervals.
	Max Time To Kill Task (due to preemption)	Shows the maximum time taken to kill a task due to pre-emption. This grid and the one above show you if you are wasting a lot of time killing queries. Time lost while a task is waiting to be killed is time lost in the cluster. If your max time to kill is high, you might want to disable this feature.
	Pre-emption Time Lost (300s interval)	Shows the time lost due to pre-emption in percentile (50th, 90th, 99th) latencies in 300 second intervals.
	Max Time Lost In Cluster (due to pre-emption)	Shows the maximum time lost due to pre-emption. If your max time to kill is high, you might want to disable this feature.
IO ELEVATOR METRICS ACROSS ALL NODES	Column Decoding Time (30s interval)	Shows the percentile (50th, 90th, 99th) latencies for time it takes to decode the column chunk (convert encoded column chunk to column vector batches for processing) in 30 second intervals. The cache comes from IO Elevator. It loads data from HDFS to the cache, and then from the cache to the executor. This metric shows how well the threads are performing and is useful to see that the threads are running.
	Max Column Decoding Time	Shows the maximum time taken to decode column chunk (convert encoded column chunk to column vector batches for processing).
JVM METRICS ACROSS ALL NODES	Average JVM Heap Usage	Shows the average amount of Java Virtual Machine (JVM) heap memory used across all nodes. If the heap usage keeps increasing, you might run out of memory and the task failure count would also increase.
	Average JVM Non-Heap Usage	Shows the average amount of JVM non-heap memory used across all nodes.
	Max GcTotalExtraSleepTime	Shows the maximum garbage collection extra sleep time in milliseconds across all nodes. Garbage collection extra sleep time measures when the garbage collection monitoring is delayed (for example, the thread does not wake up after 500 milliseconds).
	Max GcTimeMillis	Shows the total maximum GC time in milliseconds across all nodes.
	Total JVM Threads	Shows the total number of JVM threads that are in a NEW, RUNNABLE, WAITING, TIMED_WAITING, and TERMINATED state across all nodes.

Row	Metrics	Description
JVM METRICS	Total JVM Heap Used	Shows the total amount of Java Virtual Machine (JVM) heap memory used in the daemon. If the heap usage keeps increasing, you might run out of memory and the task failure count would also increase.
	Total JVM Non-Heap Used	Shows the total amount of JVM non-heap memory used in the LLAP daemon. If the non-heap memory is over-allocated, you might run out of memory and the task failure count would also increase.
	Max GcTotalExtraSleepTime	Shows the maximum garbage collection extra sleep time in milliseconds in the LLAP daemon. Garbage collection extra sleep time measures when the garbage collection monitoring is delayed (for example, the thread does not wake up after 500 milliseconds).
	Max GcTimeMillis	Shows the total maximum GC time in milliseconds in the LLAP daemon.
	Max JVM Threads Runnable	Shows the maximum number of Java Virtual Machine (JVM) threads that are in RUNNABLE state.
	Max JVM Threads Blocked	Shows the maximum number of JVM threads that are in BLOCKED state. If you are seeing spikes in the threads blocked, you might have a problem with your LLAP daemon.
	Max JVM Threads Waiting	Shows the maximum number of JVM threads that are in WAITING state.
	Max JVM Threads Timed Waiting	Shows the maximum number of JVM threads that are in TIMED_WAITING state.

Hive LLAP Daemon

Metrics that show operating status for Hive LLAP daemons.

Table 27: Hive LLAP Daemon metrics descriptions

Row	Metrics	Description
EXECUTOR METRICS	Total Requests Submitted	Shows the total number of task requests handled by the daemon.
	Total Requests Succeeded	Shows the total number of successful task requests handled by the daemon.
	Total Requests Failed	Shows the total number of failed task requests handled by the daemon.
	Total Requests Killed	Shows the total number of killed task requests handled by the daemon.
	Total Requests Evicted From Wait Queue	Shows the total number of task requests handled by the daemon that were evicted from the wait queue. Tasks are evicted if all of the executor threads are in use by higher priority tasks.
	Total Requests Rejected	Shows the total number of task requests handled by the daemon that were rejected by the task executor service. Task are rejected if all of the executor threads are in use and the wait queue is full of tasks that are not eligible for eviction.
	Available Execution Slots	Shows the total number of free slots that are available for execution including free executor threads and free slots in the wait queue.
	95th Percentile Pre-emption Time Lost (300s interval)	Shows the 95th percentile latencies for time lost due to pre-emption in 300 second intervals.
	Max Pre-emption Time Lost	Shows the maximum time lost due to pre-emption.
	95th Percentile Time to Kill Pre-empted Task (300s interval)	Shows the 95th percentile latencies for time taken to kill tasks due to pre-emption in 300 second intervals.
	Max Time To Kill Task Pre-empted Task	Shows the maximum time taken to kill a task due to pre-emption.

Row	Metrics	Description
CACHE METRICS	Total Cache Used	Shows the total amount of cache usage (Total, Remaining, and Used) in LLAP daemon cache.
	Heap Usage	Shows the amount of memory remaining in LLAP daemon cache.
	Average Cache Hit Rate	As the data is released from the cache, the curve should increase. For example, the first query should run at 0, the second at 80-90 seconds, and then the third 10% faster. If, instead, it decreases, there might be a problem in the LLAP daemon.
	Total Cache Read Requests	Shows the total number of read requests received by LLAP daemon cache.
THREAD STATES	95th Percentile Column Decoding Time (30s interval)	Shows the 95th percentile latencies for time it takes to decode the column chunk (convert encoded column chunk to column vector batches for processing) in 30 second intervals. The cache comes from IO Elevator. It loads data from HDFS to the cache, and then from the cache to the executor. This metric shows how well the threads are performing and is useful to see that the threads are running.
	Max Column Decoding Time	Shows the maximum time taken to decode column chunk (convert encoded column chunk to column vector batches for processing).

HBase Dashboards

The following Grafana dashboards are available for HBase:

Monitoring an HBase cluster is essential for maintaining a high-performance and stable system.

Important:

Ambari disables per-region, per-table, and per-user metrics for HBase by default. See Enable specific HBase metrics if you want the Ambari Metrics System to display the more granular metrics of HBase system performance on the individual region, table, or user level.

HBase Home

The HBase - Home dashboards display basic statistics about an HBase cluster.

These dashboards provide insight to the overall status for the HBase cluster.

Table 28: HBase Home metrics descriptions

Row	Metrics	Description
REGIONSERVERS / REGIONS	Num RegionServers	Total number of RegionServers in the cluster.
	Num Dead RegionServers	Total number of RegionServers that are dead in the cluster.
	Num Regions	Total number of regions in the cluster.
	Avg Num Regions per RegionServer	Average number of regions per RegionServer.
NUM REGIONS/ STORES	Num Regions / Stores - Total	Total number of regions and stores (column families) in the cluster.
	Store File Size / Count - Total	Total data file size and number of store files.
NUM REQUESTS	Num Requests - Total	Total number of requests (read, write and RPCs) in the cluster.
	Num Request - Breakdown - Total	Total number of get,put,mutate,etc requests in the cluster.
REGIONSERVER MEMORY	RegionServer Memory - Average	Average used, max or committed on-heap and offheap memory for RegionServers.
	RegionServer Offheap Memory - Average	Average used, free or committed on-heap and offheap memory for RegionServers.
MEMORY - MEMSTORE BLOCKCACHE	Memstore - BlockCache - Average	Average blockcache and memstore sizes for RegionServers.
	Num Blocks in BlockCache - Total	Total number of (hfile) blocks in the blockcaches across all RegionServers.

Row	Metrics	Description
BLOCKCACHE	BlockCache Hit/Miss/s Tota	Total number of blockcache hits misses and evictions across all RegionServers.
	BlockCache Hit Percent - Average	Average blockcache hit percentage across all RegionServers.
OPERATION LATENCIES - GET/ MUTATE	Get Latencies - Average	Average min, median, max, 75th, 95th, 99th percentile latencies for Get operation across all RegionServers.
	Mutate Latencies - Average	Average min, median, max, 75th, 95th, 99th percentile latencies for Mutate operation across all RegionServers.
OPERATION LATENCIES - DELETE/ INCREMENT	Delete Latencies - Average	Average min, median, max, 75th, 95th, 99th percentile latencies for Delete operation across all RegionServers.
	Increment Latencies - Average	Average min, median, max, 75th, 95th, 99th percentile latencies for Increment operation across all RegionServers.
OPERATION LATENCIES - APPEND/REPLAY	Append Latencies - Average	Average min, median, max, 75th, 95th, 99th percentile latencies for Append operation across all RegionServers.
	Replay Latencies - Average	Average min, median, max, 75th, 95th, 99th percentile latencies for Replay operation across all RegionServers.
REGIONSERVER RPC	RegionServer RPC -Average	Average number of RPCs, active handler threads and open connections across all RegionServers.
	RegionServer RPC Queues - Average	Average number of calls in different RPC scheduling queues and the size of all requests in the RPC queue across all RegionServers.
REGIONSERVER RPC	RegionServer RPC Throughput - Average	Average sent and received bytes from the RPC across all RegionServers.

HBase RegionServers

The HBase - RegionServers dashboards display metrics for RegionServers in the monitored HBase cluster, including some performance-related data.

These dashboards help you view basic I/O data and compare load among RegionServers.

Table 29: HBAse RegionServes metrics descriptions

Row	Metrics	Description
NUM REGIONS	Num Regions	Number of regions in the RegionServer.
STORE FILES	Store File Size	Total size of the store files (data files) in the RegionServer.
	Store File Count	Total number of store files in the RegionServer.
NUM REQUESTS	Num Total Requests /s	Total number of requests (both read and write) per second in the RegionServer.
	Num Write Requests /s	Total number of write requests per second in the RegionServer.
	Num Read Requests /s	Total number of read requests per second in the RegionServer.
NUM REQUESTS - GET / SCAN	Num Get Requests /s	Total number of Get requests per second in the RegionServer.
	Num Scan Next Requests /s	Total number of Scan requests per second in the RegionServer.
NUM REQUESTS - MUTATE / DELETE	Num Mutate Requests - /s	Total number of Mutate requests per second in the RegionServer.
	Num Delete Requests /s	Total number of Delete requests per second in the RegionServer.
NUM REQUESTS - APPEND / INCREMENT	Num Append Requests /s	Total number of Append requests per second in the RegionServer.
	Num Increment Requests /s	Total number of Increment requests per second in the RegionServer.
	Num Replay Requests /s	Total number of Replay requests per second in the RegionServer.
MEMORY	RegionServer Memory Used	Heap Memory used by the RegionServer.
	RegionServer Offheap Memory Used	Offheap Memory used by the RegionServer.

Row	Metrics	Description
MEMSTORE	Memstore Size	Total Memstore memory size of the RegionServer.
BLOCKCACHE - OVERVIEW	BlockCache - Size	Total BlockCache size of the RegionServer.
	BlockCache - Free Size	Total free space in the BlockCache of the RegionServer.
	Num Blocks in Cache	Total number of hfile blocks in the BlockCache of the RegionServer.
BLOCKCACHE - HITS/MISSES	Num BlockCache Hits /s	Number of BlockCache hits per second in the RegionServer.
	Num BlockCache Misses /s	Number of BlockCache misses per second in the RegionServer.
	Num BlockCache Evictions /s	Number of BlockCache evictions per second in the RegionServer.
	BlockCache Caching Hit Percent	Percentage of BlockCache hits per second for requests that requested cache blocks in the RegionServer.
	BlockCache Hit Percent	Percentage of BlockCache hits per second in the RegionServer.
OPERATION LATENCIES - GET	Get Latencies - Mean	Mean latency for Get operation in the RegionServer.
	Get Latencies - Median	Median latency for Get operation in the RegionServer.
	Get Latencies - 75th Percentile	75th percentile latency for Get operation in the RegionServer
	Get Latencies - 95th Percentile	95th percentile latency for Get operation in the RegionServer.
	Get Latencies - 99th Percentile	99th percentile latency for Get operation in the RegionServer.
	Get Latencies - Max	Max latency for Get operation in the RegionServer.
OPERATION LATENCIES - SCAN NEXT	Scan Next Latencies - Mean	Mean latency for Scan operation in the RegionServer.
	Scan Next Latencies - Median	Median latency for Scan operation in the RegionServer.
	Scan Next Latencies - 75th Percentile	75th percentile latency for Scan operation in the RegionServer.
	Scan Next Latencies - 95th Percentile	95th percentile latency for Scan operation in the RegionServer.
	Scan Next Latencies - 99th Percentile	99th percentile latency for Scan operation in the RegionServer.
	Scan Next Latencies - Max	Max latency for Scan operation in the RegionServer.
OPERATION LATENCIES - MUTATE	Mutate Latencies - Mean	Mean latency for Mutate operation in the RegionServer.
	Mutate Latencies - Median	Median latency for Mutate operation in the RegionServer.
	Mutate Latencies - 75th Percentile	75th percentile latency for Mutate operation in the RegionServer.
	Mutate Latencies - 95th Percentile	95th percentile latency for Mutate operation in the RegionServer.
	Mutate Latencies - 99th Percentile	99th percentile latency for Mutate operation in the RegionServer.
	Mutate Latencies - Max	Max latency for Mutate operation in the RegionServer.
OPERATION LATENCIES - DELETE	Delete Latencies - Mean	Mean latency for Delete operation in the RegionServer.
	Delete Latencies - Median	Median latency for Delete operation in the RegionServer.
	Delete Latencies - 75th Percentile	75th percentile latency for Delete operation in the RegionServer.
	Delete Latencies - 95th Percentile	95th percentile latency for Delete operation in the RegionServer.
	Delete Latencies - 99th Percentile	99th percentile latency for Delete operation in the RegionServer.
	Delete Latencies - Max	Max latency for Delete operation in the RegionServer.
OPERATION LATENCIES - INCREMENT	Increment Latencies - Mean	Mean latency for Increment operation in the RegionServer.
	Increment Latencies - Median	Median latency for Increment operation in the RegionServer.
	Increment Latencies - 75th Percentile	75th percentile latency for Increment operation in the RegionServer.
	Increment Latencies - 95th Percentile	95th percentile latency for Increment operation in the RegionServer.
	Increment Latencies - 99th Percentile	99th percentile latency for Increment operation in the RegionServer.
	Increment Latencies - Max	Max latency for Increment operation in the RegionServer.

Row	Metrics	Description
OPERATION LATENCIES - APPEND	Append Latencies - Mean	Mean latency for Append operation in the RegionServer.
	Append Latencies - Median	Median latency for Append operation in the RegionServer.
	Append Latencies - 75th Percentile	75th percentile latency for Append operation in the RegionServer.
	Append Latencies - 95th Percentile	95th percentile latency for Append operation in the RegionServer.
	Append Latencies - 99th Percentile	99th percentile latency for Append operation in the RegionServer.
	Append Latencies - Max	Max latency for Append operation in the RegionServer.
OPERATION LATENCIES - REPLAY	Replay Latencies - Mean	Mean latency for Replay operation in the RegionServer.
	Replay Latencies - Median	Median latency for Replay operation in the RegionServer.
	Replay Latencies - 75th Percentile	75th percentile latency for Replay operation in the RegionServer.
	Replay Latencies - 95th Percentile	95th percentile latency for Replay operation in the RegionServer.
	Replay Latencies - 99th Percentile	99th percentile latency for Replay operation in the RegionServer.
	Replay Latencies - Max	Max latency for Replay operation in the RegionServer.
RPC - OVERVIEW	Num RPC /s	Number of RPCs per second in the RegionServer.
	Num Active Handler Threads	Number of active RPC handler threads (to process requests) in the RegionServer.
	Num Connections	Number of connections to the RegionServer.
RPC - QUEUES	Num RPC Calls in General Queue	Number of RPC calls in the general processing queue in the RegionServer.
	Num RPC Calls in Priority Queue	Number of RPC calls in the high priority (for system tables) processing queue in the RegionServer.
	Num RPC Calls in Replication Queue	Number of RPC calls in the replication processing queue in the RegionServer.
	RPC - Total Call Queue Size	Total data size of all RPC calls in the RPC queues in the RegionServer.
RPC - CALL QUEUED TIMES	RPC - Call Queued Time - Mean	Mean latency for RPC calls to stay in the RPC queue in the RegionServer.
	RPC - Call Queued Time - Median	Median latency for RPC calls to stay in the RPC queue in the RegionServer.
	RPC - Call Queued Time - 75th Percentile	75th percentile latency for RPC calls to stay in the RPC queue in the RegionServer.
	RPC - Call Queued Time - 95th Percentile	95th percentile latency for RPC calls to stay in the RPC queue in the RegionServer.
	RPC - Call Queued Time - 99th Percentile	99th percentile latency for RPC calls to stay in the RPC queue in the RegionServer.
	RPC - Call Queued Time - Max	Max latency for RPC calls to stay in the RPC queue in the RegionServer.
RPC - CALL PROCESS TIMES	RPC - Call Process Time - Mean	Mean latency for RPC calls to be processed in the RegionServer.
	RPC - Call Process Time - Median	Median latency for RPC calls to be processed in the RegionServer.
	RPC - Call Process Time - 75th Percentile	75th percentile latency for RPC calls to be processed in the RegionServer.
	RPC - Call Process Time - 95th Percentile	95th percentile latency for RPC calls to be processed in the RegionServer.
	RPC - Call Process Time - 99th Percentile	99th percentile latency for RPC calls to be processed in the RegionServer.
	RPC - Call Process Time - Max	Max latency for RPC calls to be processed in the RegionServer.

Row	Metrics	Description
RPC - THROUGHPUT	RPC - Received bytes /s	Received bytes from the RPC in the RegionServer.
	RPC - Sent bytes /s	Sent bytes from the RPC in the RegionServer.
WAL - FILES	Num WAL - Files	Number of Write-Ahead-Log files in the RegionServer.
	Total WAL File Size	Total files sized of Write-Ahead-Logs in the RegionServer.
WAL - THROUGHPUT	WAL - Num Appends /s	Number of append operations per second to the filesystem in the RegionServer.
	WAL - Num Sync /s	Number of sync operations per second to the filesystem in the RegionServer.
WAL - SYNC LATENCIES	WAL - Sync Latencies - Mean	Mean latency for Write-Ahead-Log sync operation to the filesystem in the RegionServer.
	WAL - Sync Latencies - Median	Median latency for Write-Ahead-Log sync operation to the filesystem in the RegionServer.
	WAL - Sync Latencies - 75th Percentile	75th percentile latency for Write-Ahead-Log sync operation to the filesystem in the RegionServer.
	WAL - Sync Latencies - 95th Percentile	95th percentile latency for Write-Ahead-Log sync operation to the filesystem in the RegionServer.
	WAL - Sync Latencies - 99th Percentile	99th percentile latency for Write-Ahead-Log sync operation to the filesystem in the RegionServer.
	WAL - Sync Latencies - Max	Max latency for Write-Ahead-Log sync operation to the filesystem in the RegionServer.
WAL - APPEND LATENCIES	WAL - Append Latencies - Mean	Mean latency for Write-Ahead-Log append operation to the filesystem in the RegionServer.
	WAL - Append Latencies - Median	Median latency for Write-Ahead-Log append operation to the filesystem in the RegionServer.
	WAL - Append Latencies - 75th Percentile	95th percentile latency for Write-Ahead-Log append operation to the filesystem in the RegionServer.
	WAL - Append Latencies - 95th Percentile	95th percentile latency for Write-Ahead-Log append operation to the filesystem in the RegionServer.
	WAL - Append Latencies - 99th Percentile	99th percentile latency for Write-Ahead-Log append operation to the filesystem in the RegionServer.
	WAL - Append Latencies - Max	Max latency for Write-Ahead-Log append operation to the filesystem in the RegionServer.
WAL - APPEND SIZES	WAL - Append Sizes - Mean	Mean data size for Write-Ahead-Log append operation to the filesystem in the RegionServer.
	WAL - Append Sizes - Median	Median data size for Write-Ahead-Log append operation to the filesystem in the RegionServer.
	WAL - Append Sizes - 75th Percentile	75th percentile data size for Write-Ahead-Log append operation to the filesystem in the RegionServer.
	WAL - Append Sizes - 95th Percentile	95th percentile data size for Write-Ahead-Log append operation to the filesystem in the RegionServer.
	WAL - Append Sizes - 99th Percentile	99th percentile data size for Write-Ahead-Log append operation to the filesystem in the RegionServer.
	WAL - Append Sizes - Max	Max data size for Write-Ahead-Log append operation to the filesystem in the RegionServer.
SLOW OPERATIONS	WAL Num Slow Append /s	Number of append operations per second to the filesystem that took more than 1 second in the RegionServer.
	Num Slow Gets /s	Number of Get requests per second that took more than 1 second in the RegionServer.

Row	Metrics	Description
	Num Slow Puts /s	Number of Put requests per second that took more than 1 second in the RegionServer.
	Num Slow Deletes /s	Number of Delete requests per second that took more than 1 second in the RegionServer.
FLUSH/ COMPACTION QUEUES	Flush Queue Length	Number of Flush operations waiting to be processed in the RegionServer. A higher number indicates flush operations being slow.
	Compaction Queue Length	Number of Compaction operations waiting to be processed in the RegionServer. A higher number indicates compaction operations being slow.
	Split Queue Length	Number of Region Split operations waiting to be processed in the RegionServer. A higher number indicates split operations being slow.
JVM - GC COUNTS	GC Count /s	Number of Java Garbage Collections per second.
	GC Count ParNew /s	Number of Java ParNew (YoungGen) Garbage Collections per second.
	GC Count CMS /s	Number of Java CMS Garbage Collections per second.
JVM - GC TIMES	GC Times /s	Total time spend in Java Garbage Collections per second.
	GC Times ParNew /s	Total time spend in Java ParNew(YoungGen) Garbage Collections per second.
	GC Times CMS /s	Total time spend in Java CMS Garbage Collections per second.
LOCALITY	Percent Files Local	Percentage of files served from the local DataNode for the RegionServer.

HBase Misc

The HBase - Misc dashboards display miscellaneous metrics related to the HBase cluster.

You can use these metrics for tasks like debugging authentication and authorization issues and exceptions raised by RegionServers.

Table 30: HBase Misc metrics descriptions

Row	Metrics	Description
REGIONS IN TRANSITION	Master - Regions in Transition	Number of regions in transition in the cluster.
	Master - Regions in Transition Longer Than Threshold Time	Number of regions in transition that are in transition state for longer than 1 minute in the cluster.
	Regions in Transition Oldest Age	Maximum time that a region stayed in transition state.
NUM THREADS - RUNNABLE	Master Num Threads - Runnable	Number of threads in the Master.
	RegionServer Num Threads - Runnable	Number of threads in the RegionServer.
NUM THREADS - BLOCKED	Master Num Threads - Blocked	Number of threads in the Blocked State in the Master.
	RegionServer Num Threads - Blocked	Number of threads in the Blocked State in the RegionServer.
NUM THREADS - WAITING	Master Num Threads - Waiting	Number of threads in the Waiting State in the Master.
	RegionServer Num Threads - Waiting	Number of threads in the Waiting State in the RegionServer.
NUM THREADS - TIMED WAITING	Master Num Threads - Timed Waiting	Number of threads in the Timed-Waiting State in the Master.
	RegionServer Num Threads - Timed Waiting	Number of threads in the Timed-Waiting State in the RegionServer.
NUM THREADS - NEW	Master Num Threads - New	Number of threads in the New State in the Master.
	RegionServer Num Threads - New	Number of threads in the New State in the RegionServer.
NUM THREADS - TERMINATED	Master Num Threads - Terminated	Number of threads in the Terminated State in the Master.
	RegionServer Num Threads - Terminated	Number of threads in the Terminated State in the RegionServer.

Row	Metrics	Description
RPC AUTHENTICATION	RegionServer RPC Authentication Successes / s	Number of RPC successful authentications per second in the RegionServer.
	RegionServer RPC Authentication Failures /s	Number of RPC failed authentications per second in the RegionServer.
RPC Authorization	RegionServer RPC Authorization Successes /s	Number of RPC successful autorizations per second in the RegionServer.
	RegionServer RPC Authorization Failures /s	Number of RPC failed autorizations per second in the RegionServer.
EXCEPTIONS	Master Exceptions /s	Number of exceptions in the Master.
	RegionServer Exceptions /s	Number of exceptions in the RegionServer.

HBase Tables

HBase - Tables metrics reflect data on the table level.

The dashboards and data help you compare load distribution and resource use among tables in a cluster at different times.

Table 31: HBase Tables metrics descriptions

Row	Metrics	Description
NUM REGIONS/ STORES	Num Regions	Number of regions for the table(s).
	Num Stores	Number of stores for the table(s).
TABLE SIZE	Table Size	Total size of the data (store files and MemStore) for the table(s).
	Average Region Size	Average size of the region for the table(s). Average Region Size is calculated from average of average region sizes reported by each RegionServer (may not be the true average).
MEMSTORE SIZE	MemStore Size	Total MemStore size of the table(s).
STORE FILES	Store File Size	Total size of the store files (data files) for the table(s).
	Num Store Files	Total number of store files for the table(s).
STORE FILE AGE	Max Store File Age	Maximum age of store files for the table(s). As compactions rewrite data, store files are also rewritten. Max Store File Age is calculated from the maximum of all maximum store file ages reported by each RegionServer.
	Min Store File Age	Minimum age of store files for the table(s). As compactions rewrite data, store files are also rewritten. Min Store File Age is calculated from the minimum of all minimum store file ages reported by each RegionServer.
	Average Store File Age	Average age of store files for the table(s). As compactions rewrite data, store files are also rewritten. Average Store File Age is calculated from the average of average store file ages reported by each RegionServer.
	Num Reference Files - Total on All	Total number of reference files for the table(s).
NUM TOTAL REQUESTS	Num Total Requests /s on Tables	Total number of requests (both read and write) per second for the table(s).
NUM READ REQUESTS	Num Read Requests /s	Total number of read requests per second for the table(s).
NUM WRITE REQUESTS	Num Write Requests /s	Total number of write requests per second for the table(s).
NUM FLUSHES	Num Flushes /s	Total number of flushes per second for the table(s).
FLUSHED BYTES	Flushed MemStore Bytes	Total number of flushed MemStore bytes for the table(s).
	Flushed Output Bytes	Total number of flushed output bytes for the table(s).

Row	Metrics	Description
FLUSH TIME HISTOGRAM	Flush Time Mean	Mean latency for Flush operation for the table(s).
	Flush Time Median	Median latency for Flush operation for the table(s).
	Flush Time 95th Percentile	95th percentile latency for Flush operation for the table(s).
	Flush Time Max	Maximum latency for Flush operation for the table(s).
FLUSH MEMSTORE SIZE HISTOGRAM	Flush MemStore Size Mean	Mean size of the MemStore for Flush operation for the table(s).
	Flush MemStore Size Median	Median size of the MemStore for Flush operation for the table(s).
	Flush Output Size 95th Percentile	95th percentile size of the MemStore for Flush operation for the table(s).
	Flush MemStore Size Max	Max size of the MemStore for Flush operation for the table(s).
FLUSH OUTPUT SIZE HISTOGRAM	Flush Output Size Mean	Mean size of the output file for Flush operation for the table(s).
	Flush Output Size Median	Median size of the output file for Flush operation for the table(s).
	Flush Output Size 95th Percentile	95th percentile size of the output file for Flush operation for the table(s).
	Flush Output Size Max	Max size of the output file for Flush operation for the table(s).

HBase Users

The HBase - Users dashboards display metrics and detailed data on a per-user basis across the cluster.

You can click the second drop-down arrow in the upper-left corner to select a single user, a group of users, or all users, and you can change your user selection at any time.

Table 32: HBase Users metrics descriptions

Row	Metrics	Description
NUM REQUESTS - GET/SCAN	Num Get Requests /s	Total number of Get requests per second for the user(s).
	Num Scan Next Requests /s	Total number of Scan requests per second for the user(s).
NUM REQUESTS - MUTATE/DELETE	Num Mutate Requests /s	Total number of Mutate requests per second for the user(s).
	Num Delete Requests /s	Total number of Delete requests per second for the user(s).
NUM REQUESTS - APPEND/ INCREMENT	Num Append Requests /s	Total number of Append requests per second for the user(s).
	Num Increment Requests /s	Total number of Increment requests per second for the user(s).

Kafka Dashboards

The following Grafana dashboards are available for Kafka:

Kafka Home

Metrics that show overall status for the Kafka cluster.

Table 33: Kafka Home metrics descriptions

Row	Metrics	Description
BYTES IN & OUT / MESSAGES IN	Bytes In & Bytes Out /sec	Rate at which bytes are produced into the Kafka cluster and the rate at which bytes are being consumed from the Kafka cluster.
	Messages In /sec	Number of messages produced into the Kafka cluster.
CONTROLLER/ LEADER COUNT & REPLICA MAXLAG	Active Controller Count	Number of active controllers in the Kafka cluster. This should always equal one.
	Replica MaxLag	Shows the lag of each replica from the leader.
	Leader Count	Number of partitions for which a particular host is the leader.

Row	Metrics	Description
UNDER REPLICATED PATRITIONS & OFFLINE PARTITONS COUNT	Under Replicated Partitions	Indicates if any partitions in the cluster are under-replicated.
	Offline Partitions Count	Indicates if any partitions are offline (which means that no leaders or replicas are available for producing or consuming).
PRODUCER & CONSUMER REQUESTS	Producer Req /sec	Rate at which producer requests are made to the Kafka cluster.
	Consumer Req /sec	Rate at which consumer requests are made from the Kafka cluster.
LEADER ELECTION AND UNCLEAN LEADER ELECTIONS	Leader Election Rate	Rate at which leader election is happening in the Kafka cluster.
	Unclean Leader Elections	Indicates if there are any unclean leader elections. Unclean leader election indicates that a replica which is not part of ISR is elected as a leader.
ISR SHRINKS / ISR EXPANDED	IsrShrinksPerSec	If the broker goes down, ISR shrinks. In such case, this metric indicates if any of the partitions are not part of ISR.
	IsrExpandsPerSec	Once the broker comes back up and catches up with the leader, this metric indicates if any partitions rejoined ISR.
REPLICA FETCHER MANAGER	ReplicaFetcherManager MaxLag	The maximum lag in messages between the follower and leader replicas.

Kafka Hosts

Metrics that show operating status for Kafka cluster on a per broker level.

Use the drop-down menus to customize your results:

- Kafka broker
- Host
- Whether to view the largest (top) or the smallest (bottom) values
- Number of values that you want to view
- Aggregator to use: average, max value, or the sum of values

Table 34: Kafka Hosts metrics descriptions

Row	Metrics	Description
BYTES IN & OUT / MESSAGES IN / UNDER REPLICATED PARTITIONS	Bytes In & Bytes Out /sec	Rate at which bytes produced into the Kafka broker and rate at which bytes are being consumed from the Kafka broker.
	Messages In /sec	Number of messages produced into the Kafka broker.
	Under Replicated Partitions	Number of under-replicated partitions in the Kafka broker.
PRODUCER & CONSUMER REQUESTS	Producer Req /sec	Rate at which producer requests are made to the Kafka broker.
	Consumer Req /sec	Rate at which consumer requests are made from the Kafka broker.
REPLICA MANAGER PARTITION/ LEADER/FETCHER MANAGER MAX LAG	Replica Manager Partition Count	Number of topic partitions being replicated for the Kafka broker.
	Replica Manager Leader Count	Number of topic partitions for which the Kafka broker is the leader.
	Replica Fetcher Manager MaxLag clientId Replica	Shows the lag in replicating topic partitions.
ISR SHRINKS / ISR EXPANDS	IsrShrinks /sec	Indicates if any replicas failed to be in ISR for the host.
	IsrExpands /sec	Indicates if any replica has caught up with leader and re-joined the ISR for the host.

Kafka Topics

Metrics related to Kafka cluster on a per topic level.

Select a topic (by default, all topics are selected) to view the metrics for that topic.

Table 35: Kafka Topics metrics descriptions

Row	Metrics	Description
MESSAGES IN/OUT & BYTES IN/OUT	MessagesInPerSec	Rate at which messages are being produced into the topic.
	MessagesOutPerSec	Rate at which messages are being consumed from the topic.
TOTAL FETCH REQUESTS	TotalFetchRequestsPerSec	Number of consumer requests coming for the topic.
TOTAL PRODUCE REQUESTS /SEC	TotalProduceRequestsPerSec	Number of producer requests being sent to the topic.
FETCHER LAG METRICS CONSUMER LAG	FetcherLagMetrics ConsumnerLag	Shows the replica fetcher lag for the topic.

Storm Dashboards

The following Grafana dashboards are available for Storm:

Storm Home

Metrics that show the operating status for Storm.

Table 36: Storm Home metrics descriptions

Row	Metrics	Description
Unnamed	Topologies	Number of topologies in the cluster.
	Supervisors	Number of supervisors in the cluster.
	Total Executors	Total number of executors running for all topologies in the cluster.
	Total Tasks	Total number of tasks for all topologies in the cluster.
Unnamed	Free Slots	Number of free slots for all supervisors in the cluster.
	Used Slots	Number of used slots for all supervisors in the cluster.
	Total Slots	Total number of slots for all supervisors in the cluster. Should be more than 0.

Storm Topology

Metrics that show the overall operating status for Storm topologies.

Select a topology (by default, all topologies are selected) to view metrics for that topology.

Table 37: Storm Topology metrics descriptions

Row	Metrics	Description
RECORDS	All Tasks Input/Output	Input Records is the number of input messages executed on all tasks. Output Records is the number of messages emitted on all tasks.
	All Tasks Acked Tuples	Number of messages acked (completed) on all tasks.
	All Tasks Failed Tuples	Number of messages failed on all tasks.
LATENCY / QUEUE	All Spouts Latency	Average latency on all spout tasks.
	All Tasks Queue	Receive Queue Population is the total number of tuples waiting in the receive queue. Send Queue Population is the total number of tuples waiting in the send queue.
MEMORY USAGE	All workers memory usage on heap	Used bytes on heap for all workers in topology.

Row	Metrics	Description
	All workers memory usage on non-heap	Used bytes on non-heap for all workers in topology.
GC	All workers GC count	PSScavenge count is the number of occurrences for parallel scavenge collector. PSMarkSweep count is the number of occurrences for parallel scavenge mark and sweep collector.
	All workers GC time	PSScavenge timeMs is the sum of the time parallel scavenge collector takes (in milliseconds). PSMarkSweep timeMs is the sum of the time parallel scavenge mark and sweep collector takes (in milliseconds). Note that GC metrics are provided based on worker GC setting, so these metrics are only available for default GC option for worker.childopts. If you use another GC option for worker, you need to copy the dashboard and update the metric name manually.

Storm Components

Metrics that show operating status for Storm topologies on a per component level.

Select a topology and a component to view related metrics.

Table 38: Storm components metrics descriptions

Row	Metrics	Description
RECORDS	Input/Output	Input Records is the number of messages executed on the selected component. Output Records is the number of messages emitted on the selected component.
	Acked Tuples	Number of messages acked (completed) on the selected component.
	Failed Tuples	Number of messages failed on the selected component.
LATENCY / QUEUE	Latency	Complete Latency is the average complete latency on the select component (for Spout). Process Latency is the average process latency on the selected component (for Bolt).
	Queue	Receive Queue Population is the total number of tuples waiting in receive queues on the selected component. Send Queue Population is the total number of tuples waiting in send queues on the selected component.

System Dashboards

The following Grafana dashboards are available for System:

System Home

Metrics to see the overall status of the cluster.

Table 39: System Home metrics descriptions

Row	Metrics	Description
OVERVIEW AVERAGES	Logical CPU Count Per Server	Average number of CPUs (including hyperthreading) aggregated for selected hosts.
	Total Memory Per Server	Total system memory available per server aggregated for selected hosts.
	Total Disk Space Per Server	Total disk space per server aggregated for selected hosts.

Row	Metrics	Description
OVERVIEW - TOTALS	Logical CPU Count Total	Total Number of CPUs (including hyperthreading) aggregated for selected hosts.
	Total Memory	Total system memory available per server aggregated for selected hosts.
	Total Disk Space	Total disk space per server aggregated for selected hosts.
CPU	CPU Utilization -Average	CPU utilization aggregated for selected hosts.
SYSTEM LOAD	System Load - Average	Load average (1 min, 5 min and 15 min) aggregated for selected hosts.
MEMORY	Memory - Average	Average system memory utilization aggregated for selected hosts.
	Memory - Total	Total system memory available aggregated for selected hosts.
DISK UTILITZATION	Disk Utilitzation - Average	Average disk usage aggregated for selected hosts.
	Disk Utilitzation - Total	Total disk available for selected hosts.
DISK IO	Disk IO - Average (upper chart)	Disk read/write counts (iops) co-related with bytes aggregated for selected hosts.
	Disk IO - Average (lower chart)	Average Individual read/write statistics as MBps aggregated for selected hosts.
	Disk IO - Total	Sum of read/write bytes/sec aggregated for selected hosts.
NETWORK IO	Network IO - Average	Average Network statistics as MBps aggregated for selected hosts.
NETWORK PACKETS	Network IO - Total	Sum of Network packets as MBps aggregated for selected hosts.
	Network Packets -Average	Average of Network packets as KBps aggregated for selected hosts.
SWAP/NUM PROCESSES	Swap Space - Average	Average swap space statistics aggregated for selected hosts.
	Num Processes - Average	Average number of processes aggregated for selected hosts.

Note:

- Average implies sum/count for values reported by all hosts in the cluster. Example: In a 30 second window, if 98 out of 100 hosts reported 1 or more value, it is the SUM(Avg value from each host + Interpolated value for 2 missing hosts)/100.
- Sum/Total implies the sum of all values in a timeslice (30 seconds) from all hosts in the cluster. The same interpolation rule applies.

System Servers

Metrics to see the system status per host on the server.

Table 40: System Servers metrics descriptions

Row	Metrics	Description
CPU - USER/ SYSTEM	CPU Utilization - User	CPU utilization per user for selected hosts.
	CPU Utilization - System	CPU utilization per system for selected hosts.
CPU - NICE/IDLE	CPU Utilization - Nice	CPU nice (Unix) time spent for selected hosts.
	CPU Utilization - Idle	CPU idle time spent for selected hosts.
CPU - IOWAIT/ INTR	CPU Utilization - iowait	CPU IO wait time for selected hosts.
	CPU Utilization - Hardware Interrupt	CPU IO interrupt execute time for selected hosts.
CPU - SOFTINTR/ STEAL	CPU Utilization - Software Interrupt	CPU time spent processing soft irqs for selected hosts.
	CPU Utilization - Steal (VM)	CPU time spent processing steal time (virtual cpu wait) for selected hosts.

Row	Metrics	Description
SYSTEM LOAD - 1 MINUTE	System Load Average - 1 Minute	1 minute load average for selected hosts.
SYSTEM LOAD - 5 MINUTE	System Load Average - 5 Minute	5 minute load average for selected hosts.
SYSTEM LOAD - 15 MINUTE	System Load Average - 15 Minute	15 minute load average for selected hosts.
MEMORY - TOTAL/ USED	Memory - Total	Total memory in GB for selected hosts.
	Memory - Used	Used memory in GB for selected hosts.
MEMORY - FREE/ CACHED	Memory - Free	Total free memory in GB for selected hosts.
	Memory - Cached	Total cached memory in GB for selected hosts.
MEMORY - BUFFERED/ SHARED	Memory - Buffered	Total buffered memory in GB for selected hosts.
	Memory - Shared	Total shared memory in GB for selected hosts.
DISK UTILITZATION	Disk Used	Disk space used in GB for selected hosts.
	Disk Free	Disk space available in GB for selected hosts.
DISK IO	Read Bytes	IOPS as read MBps for selected hosts.
	Write Bytes	IOPS as write MBps for selected hosts.
DISK IOPS	Read Count	IOPS as read count for selected hosts.
	Write Count	IOPS as write count for selected hosts.
NETWORK IO	Network Bytes Received	Network utilization as byte/sec received for selected hosts.
	Network Bytes Sent	Network utilization as byte/sec sent for selected hosts.
NETWORK PACKETS	Network Packets Received	Network utilization as packets received for selected hosts.
	Network Packets Sent	Network utilization as packets sent for selected hosts.
SWAP	Swap Space - Total	Total swap space available for selected hosts.
	Swap Space - Free	Total free swap space for selected hosts.
NUM PROCESSES	Num Processes -Total	Count of processes and total running processes for selected hosts.
	Num Processes - Runnable	Count of processes and total running processes for selected hosts.

NiFi Dashboard

The following Grafana dashboard is available for NiFi:

NiFi Home

You can use the following metrics to assess the general health of your NiFi cluster.

For all metrics available in the NiFi-Home dashboard, the single value you see is the average of the information submitted by each node in your NiFi cluster.

Table 41: NiFi Home metrics descriptions

Row	Metrics	Description
JVM INFO	JVM Heap Usage	Displays the amount of memory being used by the JVM process. For NiFi, the default configuration is 512 MB.
	JVM File Descriptor Usage	Shows the number of connections to the operating system. You can monitor this metric to ensure that your JVM file descriptors, or connections, are opening and closing as tasks complete.
	JVM Uptime	Displays how long a Java process has been running. You can use this metric to monitor Java process longevity, and any unexpected restarts.

Row	Metrics	Description
THREAD INFO	Active Threads	NiFi has two user configurable thread pools: • Maximum timer driven thread count (default 10) • Maximum event driven thread count (default 5) This metric displays the number of active threads from these two pools.
	Thread Count	Displays the total number of threads for the JVM process that is running NiFi. This value is larger than the two pools above, because NiFi uses more than just the timer and event driven threads.
	Daemon Thread Count	Displays the number of daemon threads that are running. A daemon thread is a thread that does not prevent the JVM from exiting when the program finishes, even if the thread is still running.
FLOWFILE INFO	FlowFiles Received	Displays the number of FlowFiles received into NiFi from an external system in the last 5 minutes.
	FlowFiles Sent	Displays the number of FlowFiles sent from NiFi to an external system in the last 5 minutes.
	FlowFiles Queued	Displays the number of FlowFiles queued in a NiFi processor connection.
BYTE INFO	Bytes Received	Displays the number of bytes of FlowFile data received into NiFi from an external system, in the last 5 minutes.
	Bytes Sent	Displays the number of bytes of FlowFile data sent from NiFi to an external system, in the last 5 minutes.
	Bytes Queued	Displays the number of bytes of FlowFile data queued in a NiFi processor connection.

Tuning performance for AMS

To get optimal performance from Ambari Metrics System, review the following Metrics Collector configuration options:

Customize the AMS collector mode

You can change the mode of the metrics collector from the default, embedded mode to distributed mode.

About this task

Metrics Collector is built using Hadoop technologies such as Apache HBase, Apache Phoenix, and Apache Traffic Server (ATS). The Collector can store metrics data on the local file system, referred to as *embedded mode*, or use an external HDFS, referred to as *distributed mode*. By default, the Collector runs in embedded mode. In embedded mode, the Collector captures and writes metrics to the local file system on the host where the Collector is running.

Important: When running in embedded mode, you should confirm that hbase.rootdir and hbase.tmp.dir have adequately sized and lightly used partitions. Directory configurations in **Ambari Metrics** > **Configs** > **Advanced** > **ams-hbase-site** are using a sufficiently-sized and not-heavily-utilized partition, such as: file:///grid/0/var/lib/ambari-metrics-collector/hbase. You should also confirm that the TTL settings are appropriate.

When the Collector is configured for distributed mode, it writes metrics to HDFS, and the components run in distributed processes, which helps to manage CPU and memory consumption. To switch the Metrics Collector from embedded mode to distributed mode:

Procedure

1. In **Ambari Web**, browse to **Services** > **Ambari Metrics** > **Configs**.
2. Change the values of the following properties to the values shown in the following table:

Table 42: AMS Config Properties

Configuration Section	Property	Description	Value
General	Metrics Service operation mode (timeline.metrics.service.operation.mode)	Designates whether to run in distributed or embedded mode.	distributed
Advanced ams-hbase-site	hbase.cluster.distributed	Indicates AMS will run in distributed mode.	true
Advanced ams-hbase-site	hbase.rootdir 1	The HDFS directory location where metrics will be stored.	hdfs://$NAMENODE_FQDN:8020/apps/ams/metrics

3. Using **Ambari Web** > **Hosts** > **Components**, restart the Metrics Collector.

   If your cluster if configured for a highly available NameNode, set the hbase.rootdir value to use the HDFS name service instead of the NameNode host name.

   hdfs://hdfsnameservice/apps/ams/metrics

4. Optionally, you can migrate existing data from the local store to HDFS prior to switching to distributed mode:
   a) Create an HDFS directory for the ams user.
      su - hdfs -c 'hdfs dfs -mkdir -p /apps/ams/metrics'
   b) Stop Metrics Collector.
   c) Copy the metric data from the AMS local directory to an HDFS directory.

      This is the value of hbase.rootdir in Advanced ams-hbase-site used when running in embedded mode.

      For example:

      ```
      su - hdfs -c 'hdfs dfs -copyFromLocal
      /var/lib/ambari-metrics-collector/hbase/* /apps/ams/metrics'
      ```

      ```
      su - hdfs -c 'hdfs dfs -copyFromLocal
      /var/lib/ambari-metrics-collector/hbase/* /apps/ams/metrics'
      ```

      ```
      su - hdfs -c 'hdfs dfs -chown -R ams:hadoop
      /apps/ams/metrics'
      ```

   d) Switch to distributed mode.
   e) Restart the Metrics Collector.

What to do next

If you are working with Apache HBase cluster metrics and want to display the more granular metrics of HBase cluster performance on the individual region, table, or user level, see .

Customize AMS TTL settings

Customize the configuration properties in **Advanced ams-site** to configure time-to-live (TTL) for aggregated metrics.

About this task

AMS enables you to configure time-to-live (TTL) for aggregated metrics by navigating to **Ambari Metrics** > **Configs** > **Advanced ams-site**. Each property name is self explanatory and controls the amount of time to keep metrics (in seconds) before they are purged. TTL values are set in seconds. For example, assume that you are running a single-node sandbox and want to ensure that no values are stored for more than seven days, to reduce local disk space consumption. In this case, you can set to 604800s (seven days) any property ending in .ttl that has a value greater than 604800.

Before you begin

In **Ambari Metrics** > **Configs** > **Advanced ams-site**, reset TTL values for the following properties:

- timeline.metrics.cluster.aggregator.daily.ttl, which controls the daily aggregation TTL and is set by default to two years.
- timeline.metrics.cluster.aggregator.minute.ttl, , which controls minute -level aggregated metrics TTL.
- timeline.metrics.host.aggregator.ttl, which controls host-based precision metrics TTL.

If you are working in an environment prior to Apache Ambari 2.1.2, you should make these settings during installation; otherwise, you must use the HBase shell by running the following command from the Collector host:

/usr/lib/ams-hbase/bin/hbase --config /etc/ams-hbase/conf shell

Procedure

- After you are connected, update each of the following tables with the TTL value hbase(main):000:0> alter 'METRIC_RECORD_DAILY', { NAME => '0', TTL => 604800}

Table 43: AMS TTL Mappings

Map This TTL Property	To This HBase Table
timeline.metrics.cluster.aggregator.daily.ttl	METRIC_AGGREGATE_DAILY
timeline.metrics.cluster.aggregator.hourly.ttl	METRIC_AGGREGATE_HOURLY
timeline.metrics.cluster.aggregator.minute.ttl	METRIC_AGGREGATE
timeline.metrics.host.aggregator.daily.ttl	METRIC_RECORD_DAILY
timeline.metrics.host.aggregator.hourly.ttl	METRIC_RECORD_HOURLY
timeline.metrics.host.aggregator.minute.ttl	METRIC_RECORD_MINUTE
timeline.metrics.host.aggregator.ttl	METRIC_RECORD

Customize AMS memory settings

AMS runs across multiple components, each of which impacts memory use.

About this task

Because AMS uses multiple components, such as Apache HBase and Apache Phoenix to store and query metrics, you must consider multiple properties for tuning AMS memory use.

Procedure

- Tune the AMS memory settings using the following properties:

Table 44: AMS Memory Settings

Configuration	Property	Description
Advanced ams-env	metrics_collector_heapsize	Heap size configuration for the Collector.
Advanced ams-hbase-env	hbase_regionserver_heapsize	Heap size configuration for the single AMS HBase Region Server.
Advanced ams-hbase-env	hbase_master_heapsize	Heap size configuration for the single AMS HBase Master.
Advanced ams-hbase-env	regionserver_xmn_size	Maximum value for the young generation heap size for the single AMS HBase RegionServer.
Advanced ams-hbase-env	hbase_master_xmn_size	Maximum value for the young generation heap size for the single AMS HBase Master.

Customize AMS environment specific settings for a cluster

The Metrics Collector mode, TTL settings, memory settings, and disk space requirements for AMS depend on the number of nodes in the cluster.

Procedure

- Tune your AMS environment based on the following recommendations and tuning guidelines:

Table 45: AMS Environment Tuning Recommendations

Cluster Environment	Host Count	Disk Space	Collector Mode	TTL	Memory Settings
Single-Node Sandbox	1	2GB	embedded	Reduce TTLs to 7 Days	metrics_collector_heap_size=1024 hbase_regionserver_heapsize=512 hbase_master_heapsize=512 hbase_master_xmn_size=128
PoC	1-5	5GB	embedded	Reduce TTLs to 30 Days	metrics_collector_heap_size=1024 hbase_regionserver_heapsize=512 hbase_master_heapsize=512 hbase_master_xmn_size=128
Pre-Production	5-20	20GB	embedded	Reduce TTLs to 3 Months	metrics_collector_heap_size=1024 hbase_regionserver_heapsize=1024 hbase_master_heapsize=512 hbase_master_xmn_size=128
Production	20-50	50GB	embedded	n.a.	metrics_collector_heap_size=1024 hbase_regionserver_heapsize=1024 hbase_master_heapsize=512 hbase_master_xmn_size=128
Production	50-200	100GB	embedded	n.a.	metrics_collector_heap_size=2048 hbase_regionserver_heapsize=2048 hbase_master_heapsize=2048 hbase_master_xmn_size=256
Production	200-400	200GB	embedded	n.a.	metrics_collector_heap_size=2048 hbase_regionserver_heapsize=2048 hbase_master_heapsize=2048 hbase_master_xmn_size=512
Production	400-800	200GB	distributed	n.a.	metrics_collector_heap_size=8192 hbase_regionserver_heapsize=122288 hbase_master_heapsize=1024 hbase_master_xmn_size=1024 regionserver_xmn_size=1024

Cluster Environment	Host Count	Disk Space	Collector Mode	TTL	Memory Settings
Production	800+	500GB	distributed	n.a.	metrics_collector_heap_size=12288 hbase_regionserver_heapsize=16384 hbase_master_heapsize=16384 hbase_master_xmn_size=2048 regionserver_xmn_size=1024

Move the AMS metrics collector

Use this procedure to move the Ambari Metrics Collector to a new host.

Procedure

1. In **Ambari Web**, stop the **Ambari Metrics** service.
2. Execute the following API call to delete the current Metric Collector component:
   curl -u admin:admin -H "X-Requested-By:ambari" - i -X DELETE http:// [AMBARI_SERVER_NAME] >:8080/ api/v1/clusters/ [CLUSTER_NAME] /hosts/ [METRICS_COLLECTOR_HOSTNAME] /host_components/ [METRICS_COLLECTOR]
3. Execute the following API call to add Metrics Collector to a new host:
   curl -u admin:admin -H "X-Requested-By:ambari" - i -X POST http:// [AMBARI_SERVER_NAME]:8080/ api/v1/clusters/ [CLUSTER_NAME]/hosts/ [METRICS_COLLECTOR_HOSTNAME]/host_components/ [METRICS_COLLECTOR]
4. In **Ambari Web** > **Hosts**, browse to the page of the host on which you installed the new Metrics Collector and click **Install the Metrics Collector**.
5. In **Ambari Web**, start the **Ambari Metrics** service.

What to do next

Restarting all services is not required after moving the Ambari Metrics Collector, using Ambari 2.5 and later.

Enable specific HBase metrics

HBase metrics other than HBase RegionServer metrics are disabled by default.

About this task

Other than HBase RegionServer metrics, Ambari disables per-region, per-table, and per-user metrics by default. These metrics can be numerous and therefore cause performance issues. If you want Ambari to collect these metrics, you can enable them. You should test this option and confirm that your AMS performance is acceptable before enabling additional HBase metrics in a production environment.

Procedure

1. On the Ambari Server, browse to the following location:
   /var/lib/ambari-server/resources/stacks/HDP/#version/services/HBASE/package/templates/
2. Edit the following template files:
   hadoop-metrics2-hbase.properties-GANGLIA-MASTER.j2
   hadoop-metrics2-hbase.properties-GANGLIA-RS.j2
3. Either comment out or remove the following lines:
   *.source.filter.class=org.apache.hadoop.metrics2.filter.RegexFilter
   hbase.*.source.filter.exclude=.*(Regions|Users|Tables).*
4. Save the template files and restart Ambari Server for the changes to take effect.

What to do next

If you upgrade Ambari to a newer version, you must re-apply this change to the template file.

Setting up AMS security

Settng secure access to Ambari Metrics System (AMS) includes defining the same password access to both Grafana and AMS and establishing https protocol for both Grafana and AMS.

Change the Grafana admin password

Change the Grafana admin password in the Grafana UI, and in the AMS configuration, using **Ambari Web**.

About this task

If you need to change the Grafana Admin password after you initially install Ambari, you must change the password directly in Grafana, and then make the same change in the Ambari Metrics configuration.

Procedure

1. In **Ambari Web**, browse to **Services** > **Ambari Metrics** > **Quick Links**, and then click **Grafana**.
   The Grafana UI opens in read-only mode.
2. Click **Sign In**, in the left column.
3. Log in as admin, using the unchanged password.
4. Click the **admin** label in the left column.
5. In the admin profile, click **Change password**.
   a) In **Change password**, enter the unchanged password.
   b) Enter and confirm the new password.
   c) Click **Change Password**.
6. Return to **Ambari Web**, browse to **Services** > **Ambari Metrics** > **Configs**.
7. On **Configs**, in the **General** section, update and confirm the Grafana Admin Password with the new password.
8. Click **Save**.

What to do next

Restart services, as prompted.

Set up https for Grafana

Limiting Grafana access to only HTTPS connections requires providing a certificate.

About this task

Using a self-signed certificate for initial trials is possible. Self-signed certificates are not recommended for production environments. After your get your certificate, you must run a special setup command.

Procedure

1. Log in to the host on which Grafana resides.
2. Browse to the Grafana configuration directory.
   cd /etc/ambari-metrics-grafana/conf/
3. Locate your certificate.

   If you want to create a temporary self-signed certificate, you can use this as an example:

```
openssl genrsa -out ams-grafana.key 2048
openssl req -new -key ams-grafana.key -out ams-grafana.csr
openssl x509 -req -days 365 -in ams-grafana.csr -signkey ams-grafana.key -
out ams-grafana.crt
```

4. Set the certificate, key file ownership, and permissions so that they are accessible to Grafana.

```
chown ams:hadoop ams-grafana.crt
chown ams:hadoop ams-grafana.key
chmod 400 ams-grafana.crt
chmod 400 ams-grafana.key
```

   For a non-root Ambari user, use:

```
chmod 444 ams-grafana.crt
```

   to enable the agent user to read the file.
5. In **Ambari Web**, browse to **Services** > **Ambari Metrics** > **Configs**.
6. Update the following properties in the **Advanced ams-grafana-ini** section:

protocol	https
cert_file	/etc/ambari-metrics-grafana/conf/ams-grafana.crt
cert-Key	/etc/ambari-metrics-grafana/conf/ams-grafana.key

7. In **Configs**, click Save.

What to do next

Restart services, as prompted.

Set up https for AMS

Limiting AMS access to only HTTPS connections requires providing a certificate.

About this task

If you want to limit access to AMS to HTTPS connections, you must provide a certificate. While it is possible to use a self-signed certificate for initial trials, it is not suitable for production environments. After your get your certificate, you must run a special setup command.

Procedure

1. Create your own CA certificate.
   openssl req -new -x509 -keyout ca.key -out ca.crt -days 365
2. Import CA certificate into the truststore.
   # keytool -keystore /<path>/truststore.jks -alias CARoot -import -file ca.crt -storepass bigdata
3. Check truststore.

```
# keytool -keystore /<path>/truststore.jks -list
Enter keystore password:

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 2 entries

caroot, Feb 22, 2016, trustedCertEntry,
Certificate fingerprint (SHA1):
 AD:EE:A5:BC:A8:FA:61:2F:4D:B3:53:3D:29:23:58:AB:2E:B1:82:AF
```

   You should see trustedCertEntry for CA.
4. Generate certificate for AMS Collector and store private key in keystore.

keytool -genkey -alias c6401.ambari.apache.org -keyalg RSA -keysize 1024 -dname "CN=c6401.ambari.apache.org,OU=IT,O=Apache,L=US,ST=US,C=US" -keypass bigdata -keystore /<path>/keystore.jks -storepass bigdata

Note: If you use an alias different than the default hostname (c6401.ambari.apache.org), then, in step 12, set the ssl.client.truststore.alias config to use that alias.

5. Create certificate request for AMS collector certificate.
keytool -keystore /<path>/keystore.jks -alias c6401.ambari.apache.org -certreq -file c6401.ambari.apache.org.csr -storepass bigdata
6. Sign the certificate request with the CA certificate.
openssl x509 -req -CA ca.crt -CAkey ca.key -in c6401.ambari.apache.org.csr -out c6401.ambari.apache.org_signed.crt -days 365 -CAcreateserial -passin pass:bigdata
7. Import CA certificate into the keystore.
keytool -keystore /<path>/keystore.jks -alias CARoot -import -file ca.crt -storepass bigdata
8. Import signed certificate into the keystore.
keytool -keystore /<path>/keystore.jks -alias c6401.ambari.apache.org -import -file c6401.ambari.apache.org_signed.crt -storepass bigdata
9. Check keystore.

```
caroot2, Feb 22, 2016, trustedCertEntry,
Certificate fingerprint (SHA1):
 7C:B7:0C:27:8E:0D:31:E7:BE:F8:BE:A1:A4:1E:81:22:FC:E5:37:D7
[root@c6401 tmp]# keytool -keystore /tmp/keystore.jks -list
Enter keystore password:

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 2 entries

caroot, Feb 22, 2016, trustedCertEntry,
Certificate fingerprint (SHA1):
 AD:EE:A5:BC:A8:FA:61:2F:4D:B3:53:3D:29:23:58:AB:2E:B1:82:AF
c6401.ambari.apache.org, Feb 22, 2016, PrivateKeyEntry,
Certificate fingerprint (SHA1):
 A2:F9:BE:56:7A:7A:8B:4C:5E:A6:63:60:B7:70:50:43:34:14:EE:AF
```

You should see PrivateKeyEntry for the ams collector hostname entry and trustedCertEntry for CA.

10. Copy /[PATH]/truststore.jks to all nodes to /[PATH]/truststore.jks and set appropriate access permissions.
11. Copy /[PATH]/keystore.jks to AMS collector node ONLY to /[PATH]/keystore.jks and set appropriate access permissions.

    Recommended: set owner to ams user and access permnissions to 400.
12. In Ambari Web, update the following AMS configs, in Advanced:

- ams-site/timeline.metrics.service.http.policy=HTTPS_ONLY
- ams-ssl-server/ssl.server.keystore.keypassword=bigdata
- ams-ssl-server/ssl.server.keystore.location=/<path>/keystore.jks
- ams-ssl-server/ssl.server.keystore.password=bigdata
- ams-ssl-server/ssl.server.keystore.type=jks
- ams-ssl-server/ssl.server.truststore.location=/<path>/truststore.jks
- ams-ssl-server/ssl.server.truststore.password=bigdata
- ams-ssl-server/ssl.server.truststore.reload.interval=10000
- ams-ssl-server/ssl.server.truststore.type=jks
- ams-ssl-client/ssl.client.truststore.location=/<path>/truststore.jks
- ams-ssl-client/ssl.client.truststore.password=bigdata

- ams-ssl-client/ssl.client.truststore.type=jks

13. In Ambari Web, Add the following AMS config property, using **Custom ams-ssl-client** > **Add Property**:

    [metrics_collector_hostname_fqdn].ssl.client.truststore.alias=[Alias used to create certificate for AMS on the host with the specified FQDN]. Default is hostname fqdn.

14. Restart services with stale configs.
15. Configure Ambari server to use truststore.
    # ambari-server setup-security Using python /usr/bin/python Security setup options...
    ==========
    Choose one of the following options: [1] Enable HTTPS for Ambari server. [2]
    Encrypt passwords stored in ambari.properties file. [3] Setup Ambari kerberos
    JAAS configuration. [4] Setup truststore. [5] Import certificate to truststore.
    ========== Enter choice, (1-5): 4 Do you want to configure a truststore [y/n] (y)? TrustStore type [jks/jceks/pkcs12] (jks):jks Path to TrustStore file :/<path>/keystore.jks Password for TrustStore: Re-enter password: Ambari Server 'setup-security' completed successfully.
16. Configure ambari server to use https instead of http in requests to AMS Collector by adding "server.timeline.metrics.https.enabled=true" to ambari.properties file.
    # echo "server.timeline.metrics.https.enabled=true" >> /etc/ambari-server/conf/ambari.properties
17. Restart ambari server.

Understanding Ambari log search

Ambari Log Search enables you to search for logs generated by Ambari-managed HDP components.

Ambari Log Search relies on the Ambari Infra service to provide Apache Solr indexing services. Two components compose the Log Search solution:

Log Feeder

The Log Feeder component parses component logs. A Log Feeder is deployed to every node in the cluster and interacts with all component logs on that host. When started, the Log Feeder begins to parse all known component logs and sends them to the Apache Solr instances (managed by the Ambari Infra service) to be indexed.

By default, only FATAL, ERROR, and WARN logs are captured by the Log Feeder. You can temporarily or permanently add other log levels using the Log Search UI filter settings

(for temporary log level capture) or through the Log Search configuration control in Ambari.

Log Search Server

The Log Search Server hosts the Log Search UI web application, providing the API that is used by Ambari and the Log Search UI to access the indexed component logs. After logging in as a local or LDAP user, you can use the Log Search UI to visualize, explore, and search indexed component logs.

Install Log Search

Use **Ambari Web** > **+Add Services** to install Ambari Log Search.

About this task

Log Search is a built-in service in Ambari 2.4 and later. You can install Log Search as part of an initial cluster deployment, or add Log Search as a single service to an existing cluster.

Procedure

- During a new cluster installation, click **+Add Services** > **Log Search**.

  Optionally, you can manually place the Log Search Server, on the same host as the Ambari Server.

The Log Feeders are automatically installed on all nodes in the cluster.

Related Information

Add a service

Access log search UI

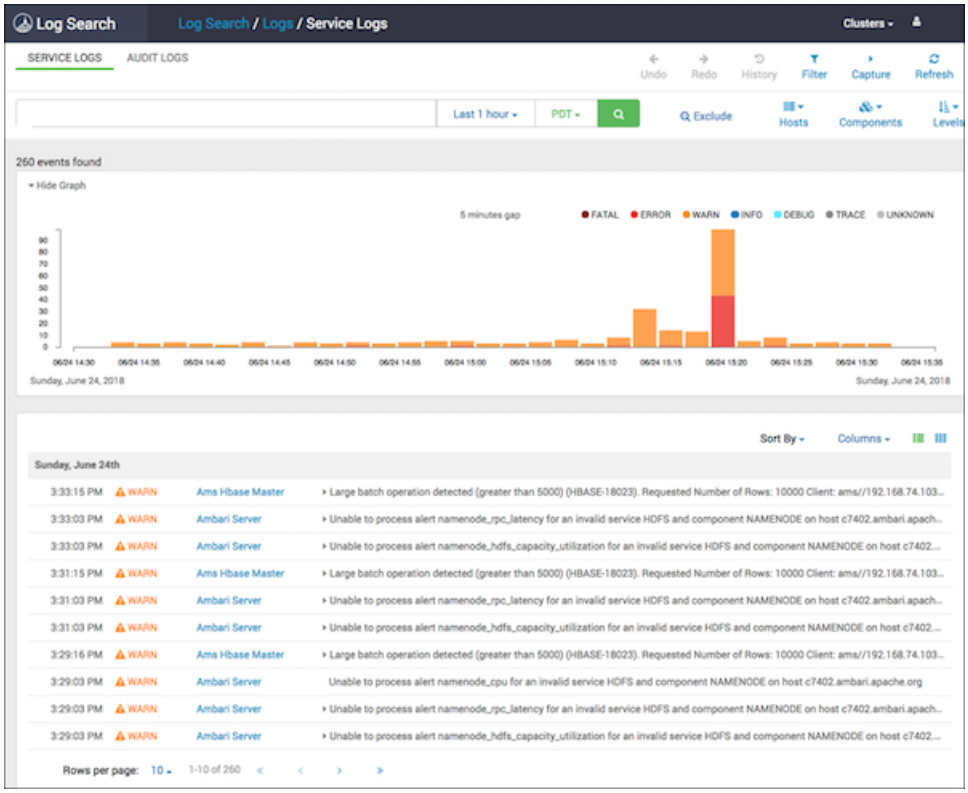
The Log Search UI is a purpose-built web application used to search HDP component logs.

About this task

The UI is focussed on helping operators quickly access and search logs from a single location. Logs can be filtered by log level, time, component, and can be searched by keyword. Helpful tools such as histograms to show number of logs by level for a time period are available, as well as controls to help rewind and fast forward search sessions, contextual click to include/exclude terms in log viewing, and multi-tab displays for troubleshooting multi-component and host issues.

Procedure

- In **Ambari Web** > **Services** > **Log Search** > **Quick Links**, click click **Log Search**.
  The Log Search UI displays **Service Logs**.



View logs for background operations

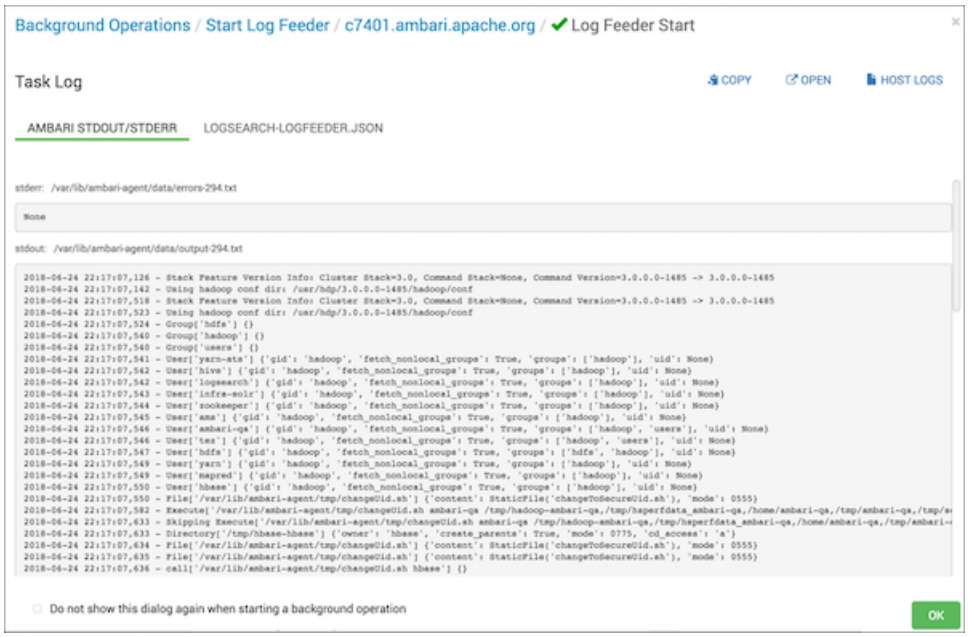
Use **Background Ops** to link to log details.

About this task

When you perform lifecycle operations such as starting or stopping services, it is critical that you have access to logs that can help you recover from potential failures. These logs are now available in **Background Ops**. **Background Ops** also links to the Host Detail Logs tab, which lists all the log files that have been indexed and can be viewed for a specific host.

Procedure

- In **Background Ops**, click **Host Logs**.



View logs for each host

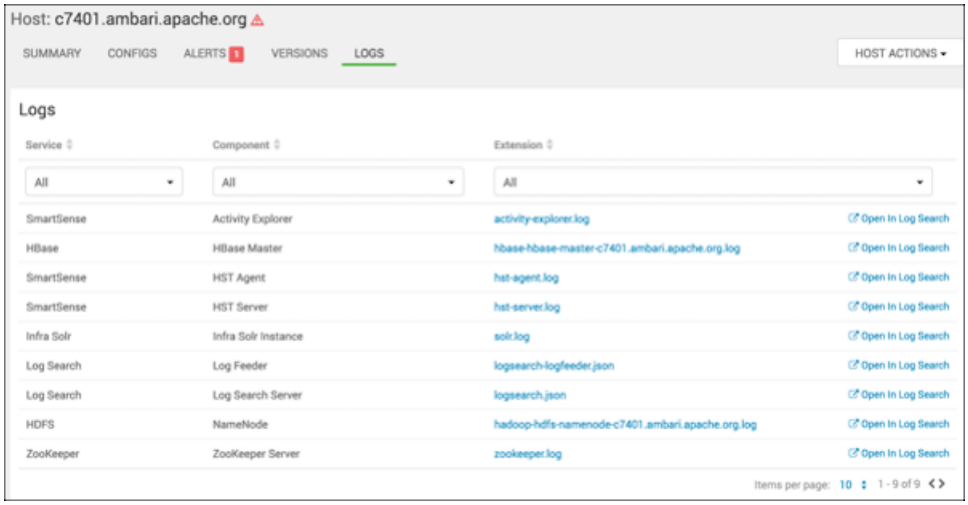
Use **Hosts** > **[HOST_FQDN]** > **Logs** to view current logs for that host.

About this task

A **Logs** tab appears on each host detail page, containing a list of indexed, viewable log files, organized by service, component, and type. You can open and search each of these files using a link to the Log Search UI.

Procedure

- On **Host Details** > **Logs**, for a specific component click **Open in Log Search**.



View service logs

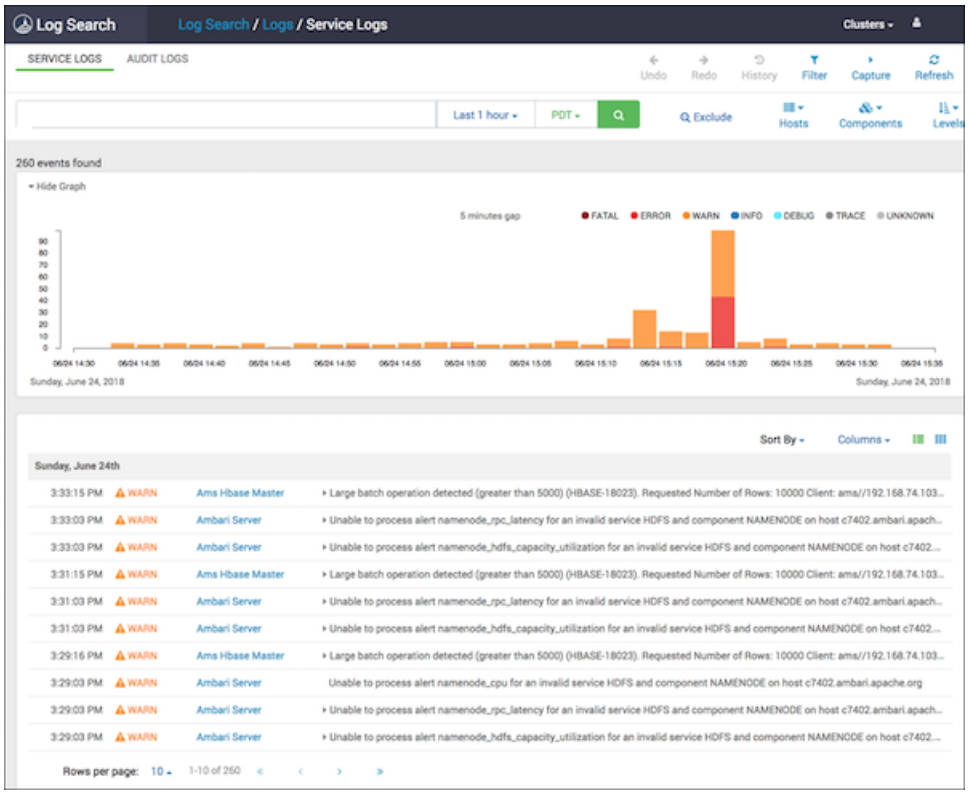
Use **Log Search** > **Service Logs** to search across all component logs.

About this task

The **Service Logs** UI is organized so that you can quickly see how many logs were captured for each log level across the entire cluster, search for keywords, include and exclude components, and match logs to your search query, filtering for specific log levels, components, and time ranges.

Procedure

- In **Log Search**, click **Service Logs**.
  Service Logs displays options to refine your current log search results.



What to do next

View Access Logs.

View audit logs

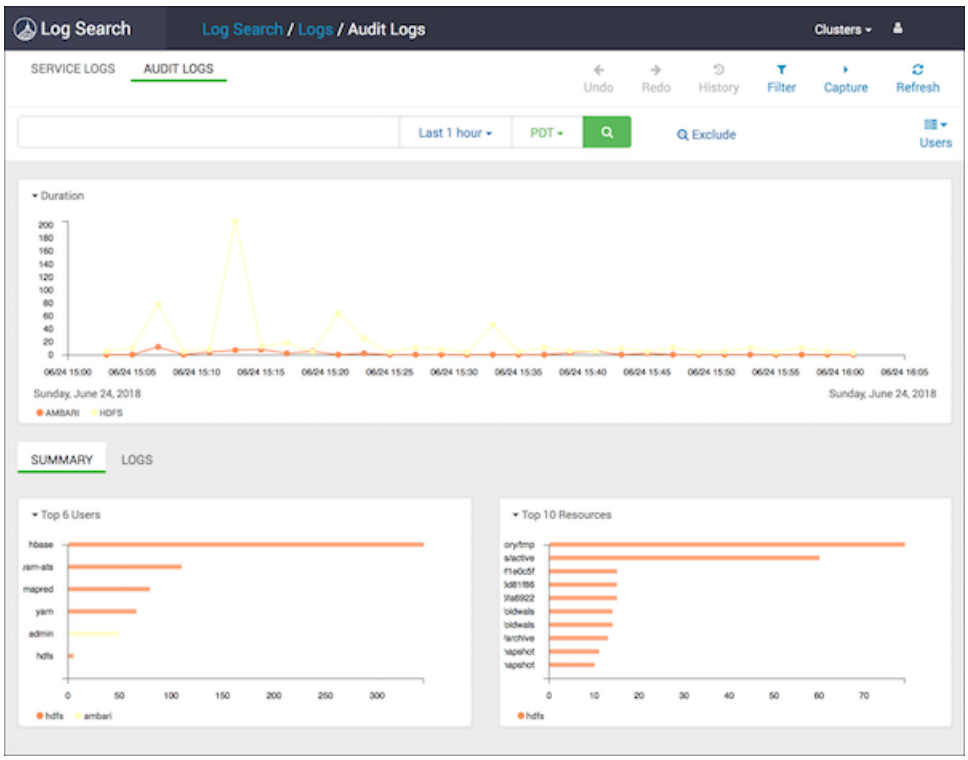
Use **Log Search** > **Audit Logs** to audit logs across all component logs.

About this task

When troubleshooting HDFS-related issues, you might find it helpful to search for and spot trends in HDFS access by users. The **Audit Logs** tab enables you to view HDFS audit log entries for a specific time frame, to see aggregated usage data showing the top ten HDFS users by file system resources accessed, as well as the top ten file system resources accessed across all users. This can help you find anomalies or hot and cold data sets.

Procedure

- Click **Audit Logs**.
  Aggregated metrics about audit log information.



Understanding Ambari Infra

Ambari Infra provides common shared services for stack components.

Many services in HDP depend on core services to index data. For example, Apache Atlas uses indexing services for tagging lineage-free text search, and Apache Ranger uses indexing for audit data. The role of Ambari Infra is to provide these common shared services for stack components.

Currently, the Ambari Infra service has only one component: the Infra Solr Instance. The Infra Solr Instance is a fully managed Apache Solr installation. By default, a single-node SolrCloud installation is deployed when the Ambari Infra Service is chosen for installation; however, you should install multiple Infra Solr Instances so that you have distributed indexing and search for Atlas, Ranger, and LogSearch (Technical Preview).

To install multiple Infra Solr Instances, you simply add them to existing cluster hosts through Ambari's **+Add Service** capability. The number of Infra Solr Instances you deploy depends on the number of nodes in the cluster and the services deployed.

Because one Ambari Infra Solr Instance is used by multiple HDP components, you should be careful when restarting the service, to avoid disrupting those dependent services. In HDP 2.5 and later, Atlas, Ranger, and Log Search depend on the Ambari Infra service.

Note:

Infra Solr Instance is intended for use only by HDP components. Use by third-party components or applications is not supported.

Large clusters produce many log entries, and Ambari Infra provides a convenient utility for archiving and purging logs that are no longer required. This utility is called the Solr Data Manager. The Solr Data Manager is a python program available in /usr/bin/infra-solr-data-manager. This program allows users to quickly archive, delete, or save data from a Solr collection, with the following usage options.

Related Information

Add a service

Operation Modes

Use one of three operation modes, based on what you want to do with your data.

-m MODE, --mode=MODE archive | delete | save

The mode to use depends on the intent. Archive stores data into the desired storage medium and then deletes the data after it has been stored. Delete is self-explanatory. Save works just like Archive, except that data is not deleted after it has been stored.

Connect to Solr

The URL to use to connect to the specific Solr Cloud instance.

-s SOLR_URL, --solr-url=[SOLR_URL]

For example:

http://c7401.ambari.apache.org:8886/solr.

-c COLLECTION, --collection=COLLECTION

The name of the Solr collection. For example: ‘hadoop_logs’

-k SOLR_KEYTAB,--solr-keytab=SOLR_KEYTAB

The keytab file to use when operating against a kerberized Solr instance.

-n SOLR_PRINCIPAL, --solr-principal=SOLR_PRINCIPAL

The principal name to use when operating against a kerberized Solr instance.

Record schema

Descriptions of the Record Schema command-line options.

-i ID_FIELD, --id-field=ID_FIELD

The name of the field in the solr schema to use as the unique identifier for each record.

-f FILTER_FIELD, --filter-field=FILTER_FIELD

The name of the field in the solr schema to filter off of. For example: 'logtime’

-o DATE_FORMAT, --date-format=DATE_FORMAT

The custom date format to use with the -d DAYS field to match log entries that are older than a certain number of days.

-e END

Based on the filter field and date format, this argument conimages the date that should be used as the end of the date range. If you use ‘2018-08-29T12:00:00.000Z’, then any records with a filter field that is after that date will be saved, deleted, or archived depending on the mode.

-d DAYS, --days=DAYS

Based on the filter field and date format, this argument conimages the number days before today should be used as the end of the range. If you use ‘30’, then any records with a filter field that is older than 30 days will be saved, deleted, or archived depending on the mode.

-q ADDITIONAL_FILTER, --additional-filter=ADDITIONAL_FILTER

Any additional filter criteria to use to match records in the collection.

Extract records

Descriptions of the Extract Record command-line options.

-r READ_BLOCK_SIZE, --read-block-size=READ_BLOCK_SIZE

The number of records to read at a time from Solr. For example: ‘10’ to read 10 records at a time.

-w WRITE_BLOCK_SIZE, --write-block-size=WRITE_BLOCK_SIZE

The number of records to write per output file. For example: ‘100’ to write 100 records per file.

-j NAME, --name=NAME name included in result files

Additional name to add to the final filename created in save or archive mode.

--json-file

Default output format is one valid json document per record delimited by a newline. This option will write out a single valid JSON document containing all of the records.

-z COMPRESSION, --compression=COMPRESSION none | tar.gz | tar.bz2 | zip | gz

Depending on how output files will be analyzed, you have the choice to choose the optimal compression and file format to use for output files. Gzip compression is used by default.

Write data to HDFS

Descriptions of the command-line options for writing data to HDFS.

-a HDFS_KEYTAB, --hdfs-keytab=HDFS_KEYTAB

The keytab file to use when writing data to a kerberized HDFS instance.

-l HDFS_PRINCIPAL, --hdfs-principal=HDFS_PRINCIPAL

The principal name to use when writing data to a kerberized HDFS instance.

-u HDFS_USER, --hdfs-user=HDFS_USER

The user to connect to HDFS as.

-p HDFS_PATH, --hdfs-path=HDFS_PATH

The path in HDFS to write data to in save or archive mode.

Write data to S3

Descriptions of the command-line options for writing data to S3.

-t KEY_FILE_PATH, --key-file-path=KEY_FILE_PATH

The path to the file on the local file system that contains the AWS Access and Secret Keys. The file should contain the keys in this format: [ACCESS_KEY],[SECRET_KEY]

-b BUCKET, --bucket=BUCKET

The name of the bucket that data should be uploaded to in save or archive mode.

-y KEY_PREFIX, --key-prefix=KEY_PREFIX

The key prefix allows you to create a logical grouping of the objects in an S3 bucket. The prefix value is similar to a directory name enabling you to store data in the same directory in a bucket. For example, if your Amazon S3 bucket name is logs, and you set prefix to hadoop/, and the file on your storage device is hadoop_logs_-_2017-10-28T01_25_40.693Z.json.gz, then the file would be identified by this URL: http://s3.amazonaws.com/logs/hadoop/hadoop_logs_-_2017-10-28T01_25_40.693Z.json.gz

-g, --ignore-unfinished-uploading

To deal with connectivity issues, uploading extracted data can be retried. If you do not wish to resume uploads, use the -g flag to disable this behaviour.

Write data locally

Descriptions of the command-line options for writing data locally.

-x LOCAL_PATH, --local-path=LOCAL_PATH

The path on the local file system that should be used to write data to in save or archive mode

Example delete indexed data

A working example of deleting indexed data.

In delete mode (-m delete), the program deletes data from the Solr collection. This mode uses the filter field (-f FITLER_FIELD) option to control which data should be removed from the index.

The command below will delete log entries from the hadoop_logs collection, which have been created before August 29, 2017, we'll use the -f option to specify the field in the Solr collection to use as a filter field, and the -e option to denote the end of the range of values to remove.

```
infra-solr-data-manager -m delete -s
://c6401.ambari.apache.org:8886/solr -c hadoop_logs -f logtime -e
 2017-08-29T12:00:00.000Z
```

Example archive indexed data

A working example of archiving indexed data.

In archive mode, the program fetches data from the Solr collection and writes it out to HDFS or S3, then deletes the data.

The program will fetch records from Solr and creates a file once the write block size is reached, or if there are no more matching records found in Solr. The program keeps track of its progress by fetching the records ordered by the filter field, and the id field, and always saves their last values. Once the file is written, it’s is compressed using the configured compression type.

After the compressed file is created the program creates a command file containing instructions with next steps. In case of any interruptions or error during the next run for the same collection the program will start executing the saved command file, so all the data would be consistent. If the error is due to invalid configuration, and failures persist, the -g option can be used to ignore the saved command file. The program supports writing data to HDFS, S3, or Local Disk.

The command below will archive data from the solr collection hadoop_logs accessible at http://c6401.ambari.apache.org:8886/solr, based on the field logtime, and will extract everything older than 1 day, read 10 documents at once, write 100 documents into a file, and copy the zip files into the local directory /tmp.

```
infra-solr-data-manager -m archive -s
http://c6401.ambari.apache.org:8886/solr -c hadoop_logs -f logtime -d
1 -r 10 -w 100 -x /tmp -v
```

Example save indexed data

A working example of saving indexed data.

Saving is similar to Archiving data except that the data is not deleted from Solr after the files are created and uploaded. The Save mode is recommended for testing that the data is written as expected before running the program in Archive mode with the same parameters.

The below example will save the last 3 days of hdfs audit logs into HDFS path "/" with the user hdfs, fetching data from a kerberized Solr.

```
infra-solr-data-manager -m save -s
http://c6401.ambari.apache.org:8886/solr -c audit_logs -f logtime -d 3
-r 10 -w 100 -q type:\”hdfs_audit\” -j hdfs_audit -k
/etc/security/keytabs/ambari-infra-solr.service.keytab -n
infra-solr/c6401.ambari.apache.org@AMBARI.APACHE.ORG -u hdfs -p /
```

Example analyze archived, indexed data with Hive

A working example of analyzing archived, indexed data with Hive.

Once data has been archived or saved to HDFS, Hive tables can be used to quickly access and analyzed stored data. Only line delimited JSON files can be analyzed with Hive. Line delimited JSON files are created by default unless the --json-file argument is passed. Data saved or archived using --json-file cannot be analyzed with Hive. In the following examples, the hive-json-serde.jar is used to process the stored JSON data. Prior to creating the included tables, the jar must be added in the Hive shell:

ADD JAR [PATH_TO_JAR]/hive-json-serde.jar

Here are some examples for table schemes for various log types. Using external tables is recommended, as it has the advantage of keeping the archives in HDFS. First ensure a directory is created to store archived or stored line delimited logs:

hadoop fs -mkdir [SOME_DIRECTORY_PATH]

Example Hadoop logs

A working example of creating an external table of Hadoop logs.

```
CREATE EXTERNAL TABLE hadoop_logs
(
logtime string,
level string,
thread_name string,
logger_name string,
file string,
line_number int,
method string,
log_message string,
cluster string,
type string,
path string,
logfile_line_number int,
host string,
ip string,
id string,
event_md5 string,
message_md5 string,
seq_num int
)
ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'
LOCATION '[SOME_DIRECTORY_PATH]';
```

Example audit logs

Recommendations for archiving audit logs.

As audit logs have a slightly different field set, we suggest to archive them separately using --additional-filter, and we offer separate schemas for HDFS, Ambari, and Ranger audit logs.

Example HDFS audit logs

A working example of creating an external table of HDFS audit logs.

```
CREATE EXTERNAL TABLE audit_logs_hdfs
(
evtTime string,
level string,
logger_name string,
log_message string,
resource string,
```

```
result int,
action string,
cliType string,
req_caller_id string,
ugi string,
reqUser string,
proxyUsers array<string>,
authType string,
proxyAuthType string,
dst string,
perm string,
cluster string,
type string,
path string,
logfile_line_number int,
host string,
ip string,
cliIP string,
id string,
event_md5 string,
message_md5 string,
seq_num int
)
ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'
LOCATION '[SOME_DIRECTORY_PATH]';
```

Example Ambari audit logs

A working example of creating an external table of Ambari audit logs.

```
CREATE EXTERNAL TABLE audit_logs_ambari
(
evtTime string,
log_message string,
resource string,
result int,
action string,
reason string,
ws_base_url string,
ws_command string,
ws_component string,
ws_details string,
ws_display_name string,
ws_os string,
ws_repo_id string,
ws_repo_version string,
ws_repositories string,
ws_request_id string,
ws_roles string,
ws_stack string,
ws_stack_version string,
ws_version_note string,
ws_version_number string,
ws_status string,
ws_result_status string,
cliType string,
reqUser string,
task_id int,
cluster string,
type string,
path string,
logfile_line_number int,
host string,
```

```
cliIP string,
id string,
event_md5 string,
message_md5 string,
seq_num int
)
ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'
LOCATION '[SOME_DIRECTORY_PATH]';
```

Example Ranger audit logs

A working example of creating an external table of Ranger audit logs.

```
CREATE EXTERNAL TABLE audit_logs_ranger
(
evtTime string,
access string,
enforcer string,
resource string,
result int,
action string,
reason string,
resType string,
reqUser string,
cluster string,
cliIP string,
id string,
seq_num int
)
ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'
LOCATION '[SOME_DIRECTORY_PATH]';
```

Tuning performance for Ambari Infra

Recommendations for tuning your operating system and Solr, based on how you use Ambari Infra and Ranger in your environment.

When using Ambari Infra to index and store Ranger audit logs, you should properly tune Solr to handle the number of audit records stored per day. The following topicss describe recommendations for tuning your operating system and Solr, based on how you use Ambari Infra and Ranger in your environment.

Tuning your operating system for use with Solr

To avoid many open network connections and exceptions related to creating new native threads:

Solr clients use many network connections when indexing and searching, and to avoid many open network connections, the following sysctl parameters are recommended:

```
net.ipv4.tcp_max_tw_buckets = 1440000
net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_tw_reuse = 1
```

These settings can be made permanent by placing them in /etc/sysctl.d/net.conf, or they can be set at runtime using the following sysctl command example:

```
sysctl -w net.ipv4.tcp_max_tw_buckets=1440000
sysctl -w net.ipv4.tcp_tw_recycle=1
sysctl -w net.ipv4.tcp_tw_reuse=1
```

Additionally, the number of user processes for solr should be increased to avoid exceptions related to creating new native threads. This can be done by creating a new file named /etc/security/limits.d/infra-solr.conf with the following contents:

```
infra-solr - nproc 6000
```

Tuning JVM settings for Solr

Heap sizing settings are very important for production Solr instances.

About this task

The heap sizing settings are very important for production Solr instances when indexing many Ranger audit logs. For production deployments, we suggest setting the Infra Solr Minimum Heap Size, and Infra Solr Maximum Heap Size to 12 GB. These settings can be found and applied by following the steps below:

Procedure

1. In **Ambari Web**, browse to **Services** > **Ambari Infra** > **Configs**.
2. In **Settings**, you will see two sliders controlling the Infra Solr Heap Size.
3. Set the Infra Solr Minimum Heap Size to 12GB or 12,288MB.
4. Set the Infra Solr Maximum Heap Size to 12GB or 12,288MB.
5. Click **Save** to save the configuration.
6. Restart the affected services as prompted by Ambari.

   The value used for the -XX:G1HeapRegionSize is based on the 12GB Solr Maximum Heap recommended. If you choose to use a different heap size for the Solr server, please consult the following table for recommendations:

Table 46: Garbage Collection Heap Size Requirements for Solr

Heap Size	G1HeapRegionSize
< 4GB	1MB
4-8GB	2MB
8-16GB	4MB
16-32GB	8MB
32-64GB	16MB
>64GB	32MB

Tuning GC settings for Solr

To use the G1 Garbage Collector with the Ambari Infra Solr Instance, follow these steps.

About this task

The garbage collection settings are very important for production Solr instances when indexing many Ranger audit logs. Using the G1 Garbage Collector is also recommended for production deployments.

Procedure

1. In **Ambari Web**, browse to **Services** > **Ambari Infra** > **Configs**.
2. In **Advanced**, expand the section for **Advanced infra-solr-env** .
3. In the **infra-solr-env template**, locate the multi-line GC_TUNE environmental variable definition, and replace it with the following content:

```
GC_TUNE="-XX:+UseG1GC
  -XX:+PerfDisableSharedMem
```

```
  -XX:+ParallelRefProcEnabled
  -XX:G1HeapRegionSize=4m
  -XX:MaxGCPauseMillis=250
  -XX:InitiatingHeapOccupancyPercent=75
  -XX:+UseLargePages
  -XX:+AggressiveOpts"
```

The value used for the -XX:G1HeapRegionSize is based on the 12GB Solr Maximum Heap recommended. If you choose to use a different heap size for the Solr server, please consult the following table for recommendations:

Table 47: Garbage Collection Heap Size Requirements for Solr

Heap Size	G1HeapRegionSize
< 4GB	1MB
4-8GB	2MB
8-16GB	4MB
16-32GB	8MB
32-64GB	16MB
>64GB	32MB

Tuning environment specific parameters

Based on the number of audit records per day.

About this task

Each of the recommendations below are dependent on the number of audit records that are indexed per day. To quickly determine how many audit records are indexed per day, use the command examples below:

Procedure

- Using a HTTP client such as curl, execute the following command:
  curl -g "http://[AMBARI_INFRA_HOSTNAME]:8886/solr/ranger_audits/select?q=(evtTime:[NOW-7DAYS+TO+*])&wt=json&indent=true&rows=0"
  You should receive a message similar to the following:

```
{
  "responseHeader":{
    "status":0,
    "QTime":1,
    "params":{
      "q":"evtTime:[NOW-7DAYS TO *]",
      "indent":"true",
      "rows":"0",
      "wt":"json"}},
  "response":{"numFound":306,"start":0,"docs":[]
  }}
```

- Take the numFound element of the response and divide it by 7 to get the average number of audit records being indexed per day. You can also replace the ‘7DAYS’ in the curl request with a broader time range, if necessary, using the following key words:
  - 1MONTHS
  - 7DAYS
- Just ensure you divide by the appropriate number if you change the event time query. The average number of records per day will be used to identify which recommendations below apply to your environment.

Less Than 50 Million Audit Records Per Day

Based on the Solr REST API call if your average number of documents per day is less than 50 million records per day, the following recommendations apply. In each recommendation, the time to live, or TTL, which controls how long a document should be kept in the index until it is removed is taken into consideration. The default TTL is 90 days, but some customers choose to be more aggressive, and remove documents from the index after 30 days. Due to this, recommendations for both common TTL settings are specified.

These recommendations assume that you are using our recommendation of 12GB heap per Solr server instance. In each situation we have recommendations for co-locating Solr with other master services, and for using dedicated Solr servers. Testing has shown that Solr performance requires different server counts depending on whether Solr is co-located or on dedicated servers. Based on our testing with Ranger, Solr shard sizes should be around 25GB for best overall performance. However, Solr shard sizes can go up to 50GB without a significant performance impact.

This configuration is our best recommendation for just getting started with Ranger and Ambari Infra so the only recommendation is using the default TTL of 90 days.

Default Time To Live (TTL) 90 days:

- Estimated total index size: ~150 GB to 450 GB
- Total number of primary/leader shards: 6
- Total number of shards including 1 replica each: 12
- Total number of co-located Solr nodes: ~3 nodes, up to 2 shards per node

  (does not include replicas)
- Total number of dedicated Solr nodes: ~1 node, up to 12 shards per node

  (does not include replicas)

50 - 100 Million Audit Records Per Day

50 to 100 million records ~ 5 - 10 GB data per day.

Default Time To Live (TTL) 90 days:

- Estimated total index size: ~ 450 - 900 GB for 90 days
- Total number of primary/leader shards: 18-36
- Total number of shards including 1 replica each: 36-72
- Total number of co-located Solr nodes: ~9-18 nodes, up to 2 shards per node

  (does not include replicas)
- Total number of dedicated Solr nodes: ~3-6 nodes, up to 12 shards per node

	(does not include replicas) Custom Time To Live (TTL) 30 days: • Estimated total index size: 150 - 300 GB for 30 days • Total number of primary/leader shards: 6-12 • Total number of shards including 1 replica each: 12-24 • Total number of co-located Solr nodes: ~3-6 nodes, up to 2 shards per node (does not include replicas) • Total number of dedicated Solr nodes: ~1-2 nodes, up to 12 shards per node (does not include replicas)
100 - 200 Million Audit Records Per Day	100 to 200 million records ~ 10 - 20 GB data per day. Default Time To Live (TTL) 90 days: • Estimated total index size: ~ 900 - 1800 GB for 90 days • Total number of primary/leader shards: 36-72 • Total number of shards including 1 replica each: 72-144 • Total number of co-located Solr nodes: ~18-36 nodes, up to 2 shards per node (does not include replicas) • Total number of dedicated Solr nodes: ~3-6 nodes, up to 12 shards per node (does not include replicas) Custom Time To Live (TTL) 30 days: • Estimated total index size: 300 - 600 GB for 30 days • Total number of primary/leader shards: 12-24 • Total number of shards including 1 replica each: 24-48 • Total number of co-located Solr nodes: ~6-12 nodes, up to 2 shards per node (does not include replicas) • Total number of dedicated Solr nodes: ~1-3 nodes, up to 12 shards per node (does not include replicas)

- If you choose to use at least 1 replica for high availability, then increase the number of nodes accordingly. If high availability is a requirement, then consider using no less than 3 Solr nodes in any configuration.
- As illustrated in these examples, a lower TTL requires less resources. If your compliance objectives call for longer data retention, you can use the SolrDataManager to archive data into long term storage (HDFS, or S3) and provides Hive tables allowing you to easily query that data. With this strategy, hot data can be stored in Solr for rapid access through the Ranger UI, and cold data can be archived to HDFS, or S3 with access provided through Ranger.

Adding new shards for Solr

Refer to the Apache Solr reference guide if you must add shards for Solr.

If after reviewing the other recommendations for Solr , you need to add additional shards to your existing deployment, the following Solr documentation will help you understand how to accomplish that task: https://archive.apache.org/dist/lucene/solr/ref-guide/apache-solr-ref-guide-5.5.pdf

Reindexing data to reduce Solr out of memory exception errors

How to enable Doc Values as a solution.

When using Ambari Infra with Ranger Audit, if you are seeing many instances of Solr exiting with Java “Out Of Memory” exceptions, a solution exists to update the Ranger Audit schema to use less heap memory by enabling DocValues. This change requires a re-index of data and is disruptive, but helps tremendously with heap memory consumption. Refer to this HCC article for the instructions on making this change: https://community.hortonworks.com/articles/156933/restore-backup-ranger-audits-to-newly-collection.html