HDInsight Administration and Security

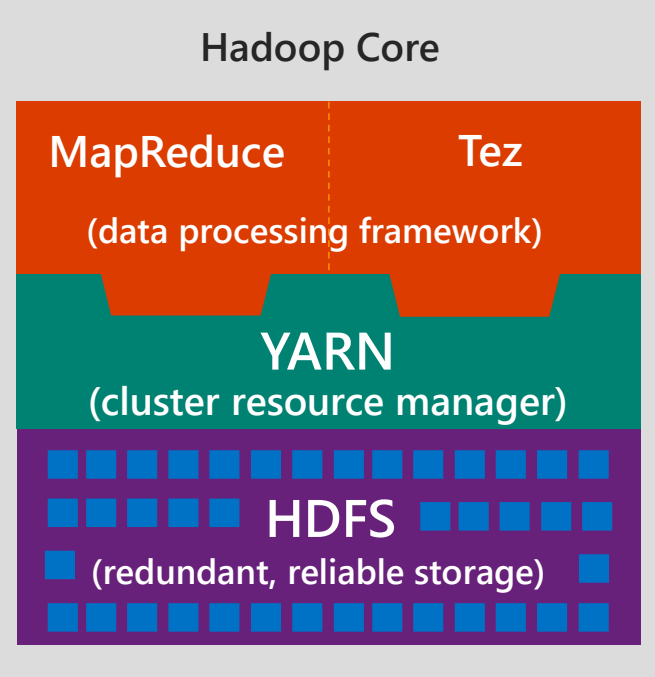
Technical Overview

Manjunath

Hadoop – What is it?

A highly reliable, distributed and parallel programming framework for analyzing big data

- An Java-based, open sourced, Apache project
- Capable of running on variety of hardware platforms, including clusters of commodity hardware
  - Is tolerant to failures of nodes, software components, network
  - Scales with the cluster
- The Hadoop core consists of:
  - A scalable, reliable file system (HDFS)
  - A framework that enables development of programs based on MapReduce (MR) or Directed Acyclic Graph (DAG) model
  - YARN, a distributed resource manager that allocates and controls access to the resource of the cluster manager
- In addition to the core Hadoop has a rich ecosystem that supports SQL/NoSQL, Streaming, Real-time and Interactive applications.



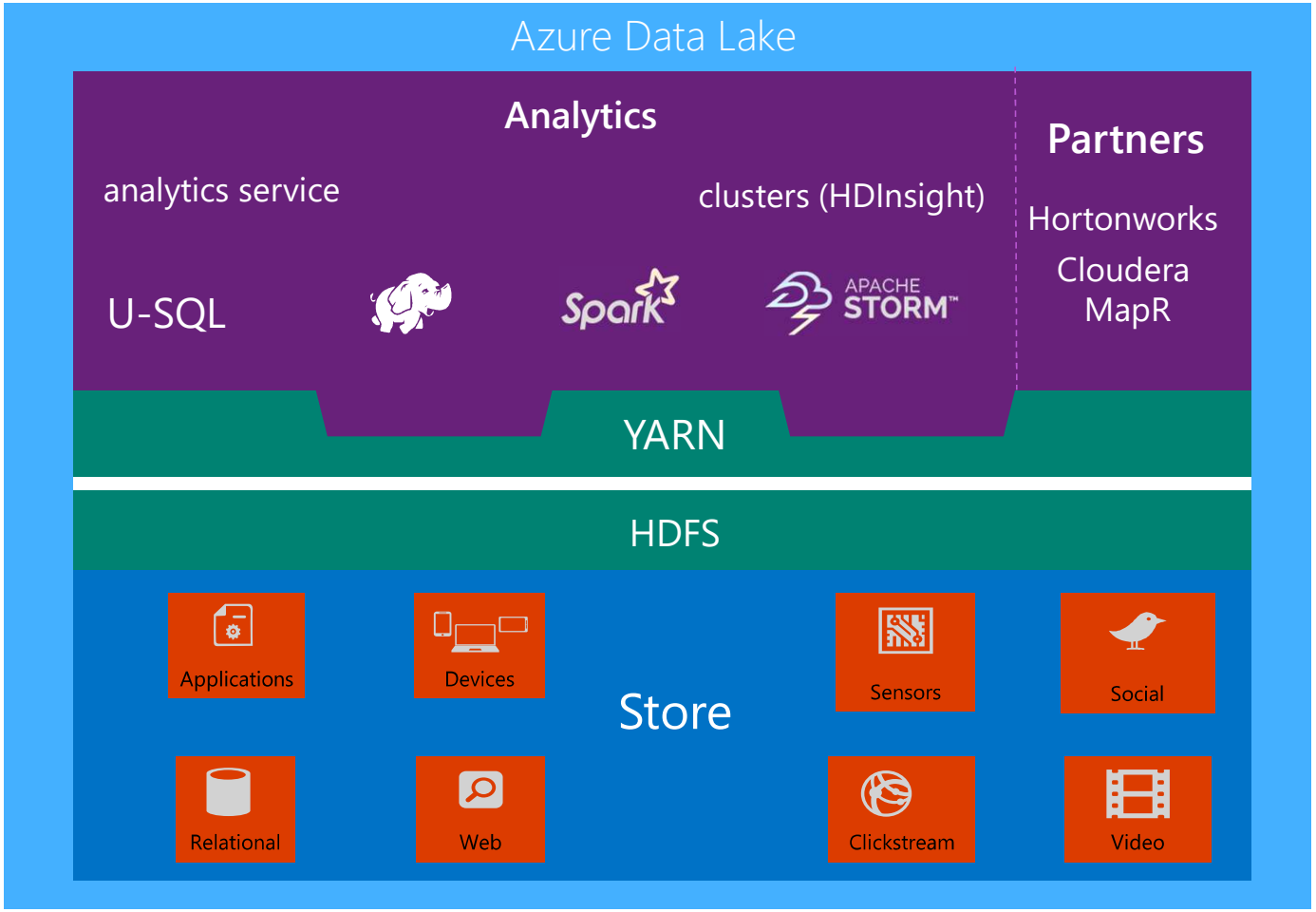
HDInsight – What is it?

A standard Apache Hadoop distribution offered as a managed service on Microsoft Azure

- Based on the Hortonworks Data Platform (HDP)
- Provisioned as clusters on Azure. Clusters can run on Windows or Linux Servers.
- Offers a capacity-on-demand, pay-as-you-go pricing model
- Integrates with:
  - Azure Blob Storage and Azure Data Lake Store for the Hadoop File System (HDFS)
  - Azure Portal for management and administration
  - Visual Studio for application development tooling

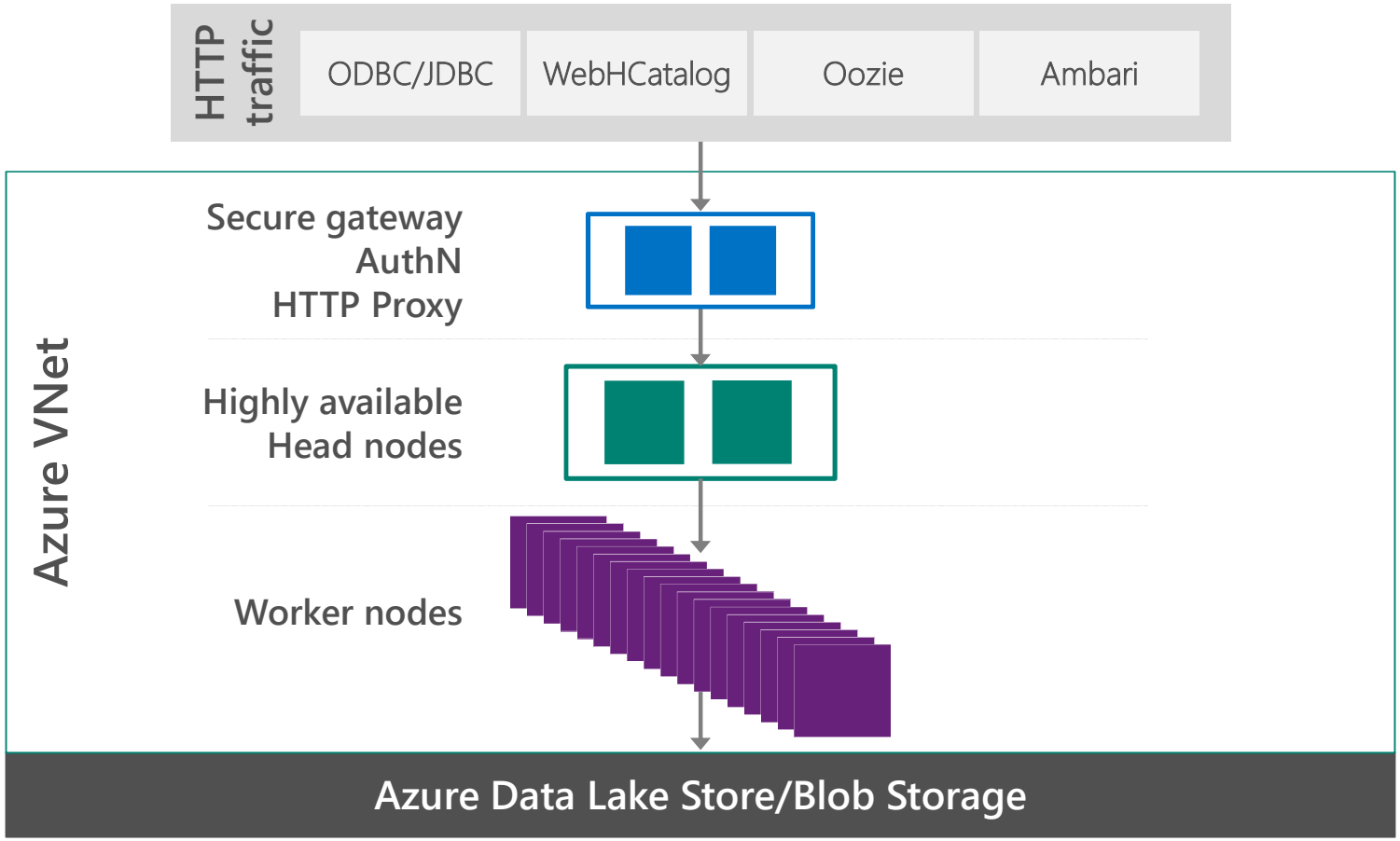
In addition to the core, HDInsight supports the Hadoop Ecosystem

HDInsight: How/Where it fits?



- ✓ Integrated analytics and storage
- ✓ Fully Managed
- ✓ Easy to use – "dial for scale"
- ✓ Proven at scale
- ✓ Analyze data of any size, shape or speed
- ✓ Open-standards based

HDInsight cluster architecture

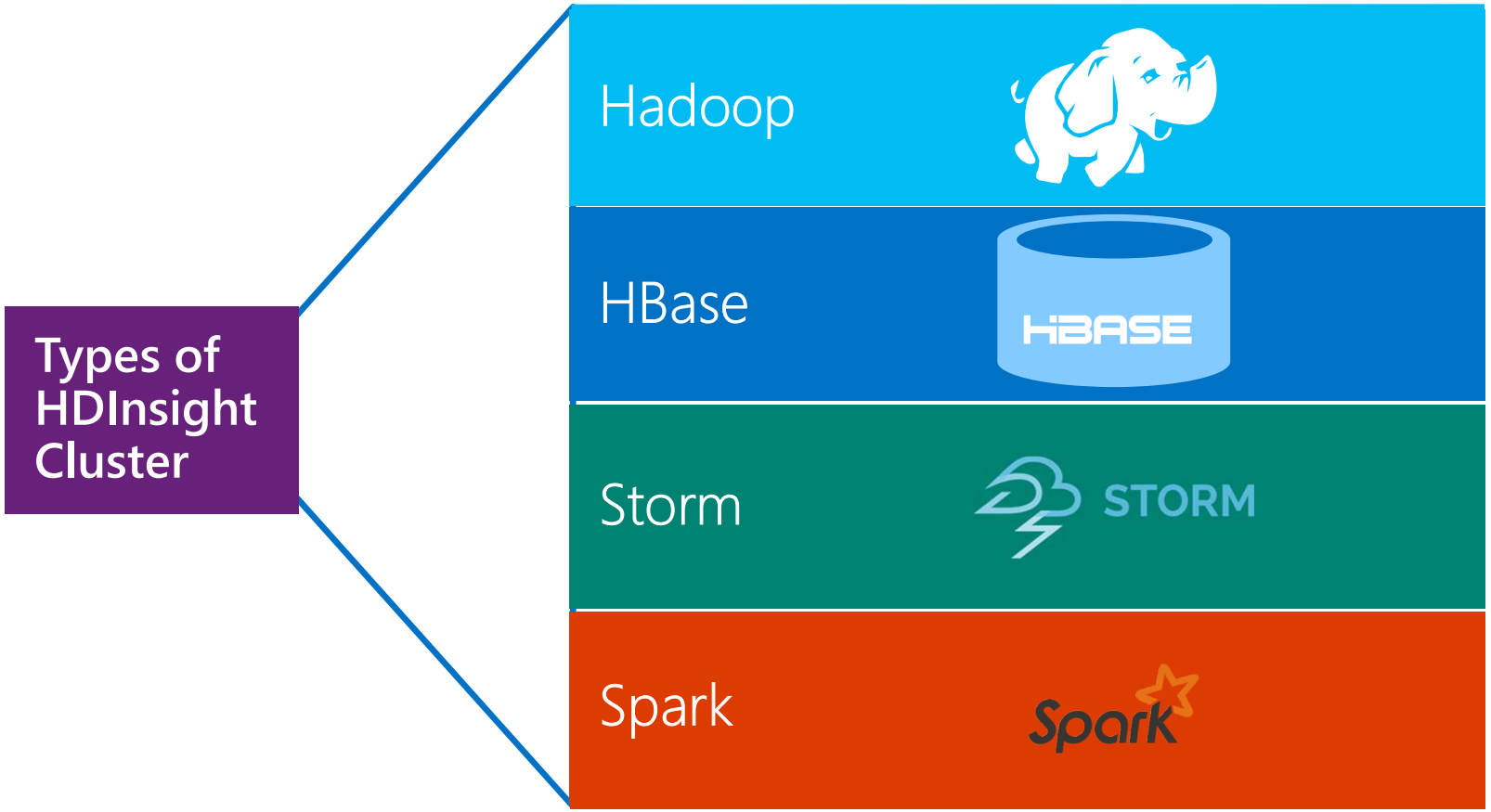


HDInsight on Linux: Administration Overview

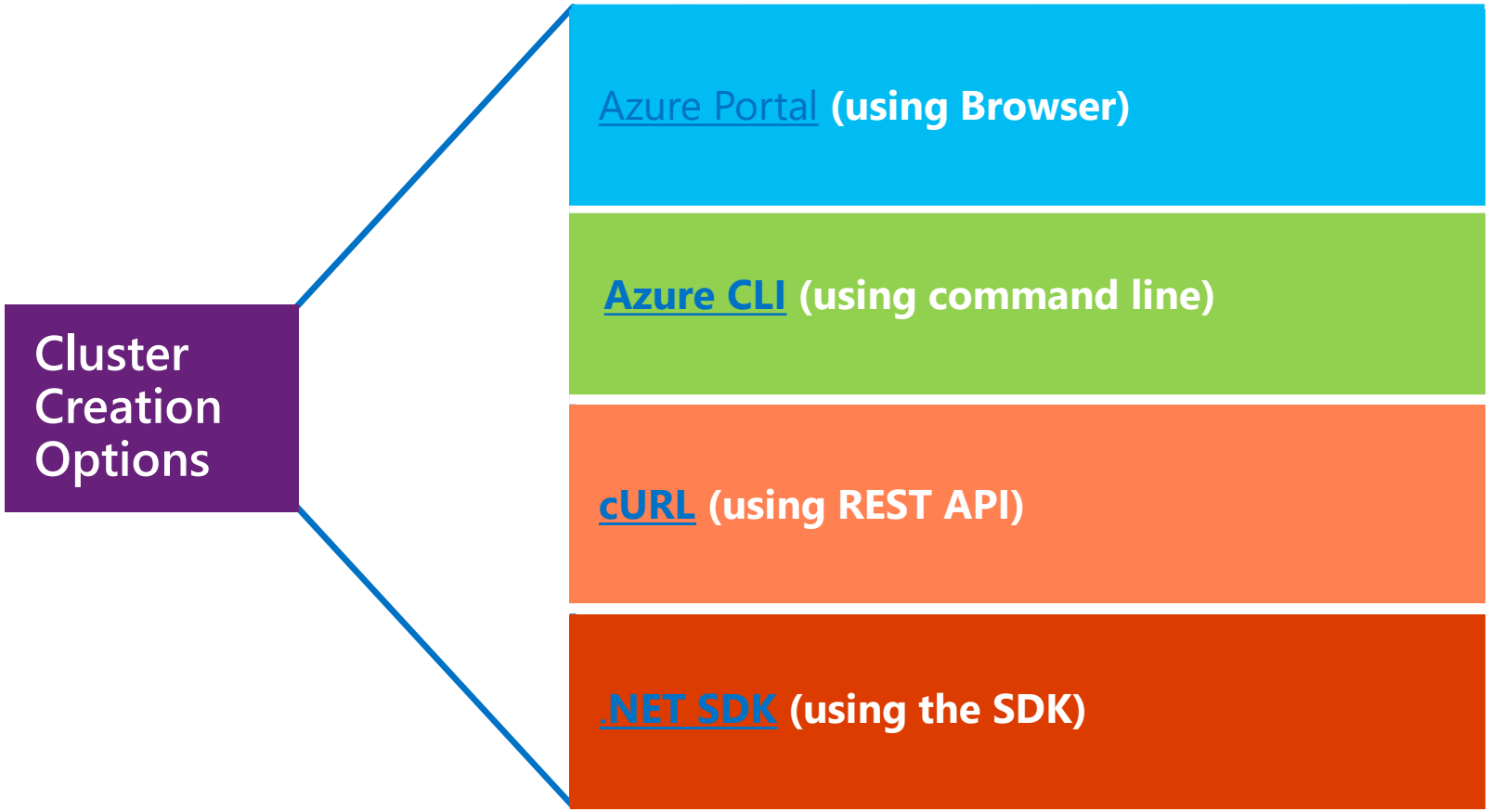
Agenda

- Creating HDI Clusters
  - Script Actions
- Audit Logs
- HDI Configuration
- Ambari Web UI

Cluster Types Overview



Ways to create HDInsight (Linux) Clusters



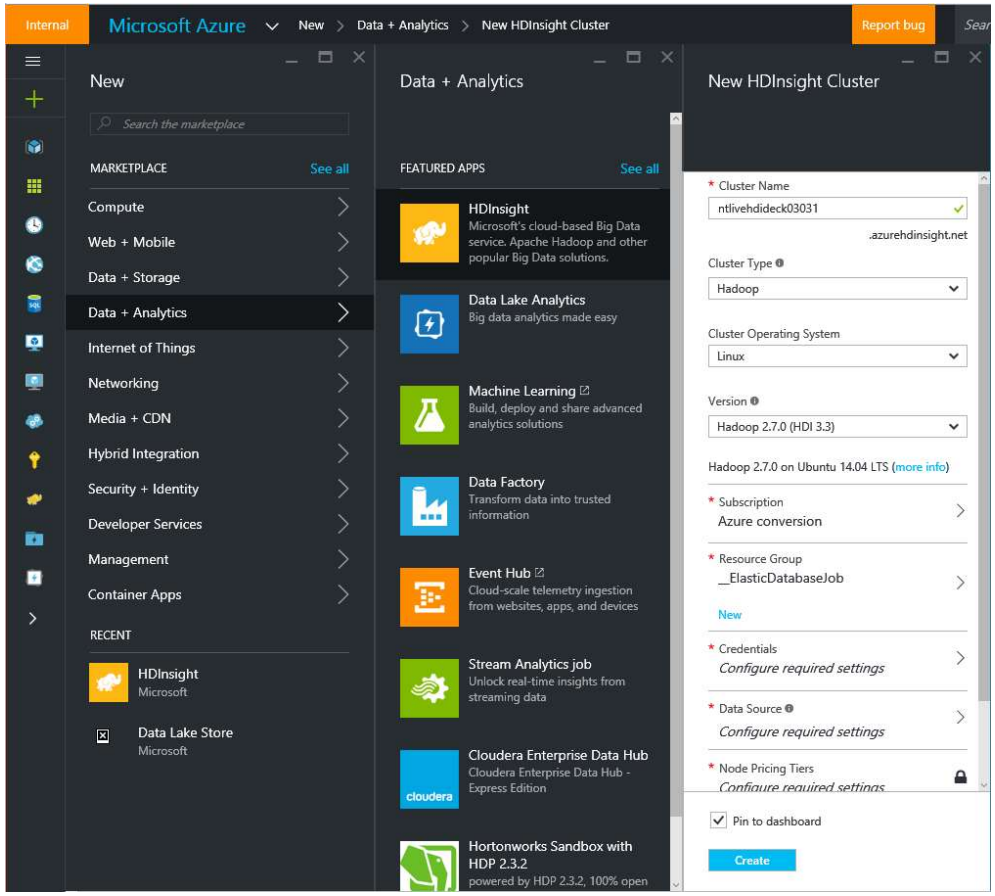
Creating a HDInsight Cluster via the Portal

Azure Portal

Azure Portal provides a guided wizard to create HDInsight clusters.

The key parameters to specify include:

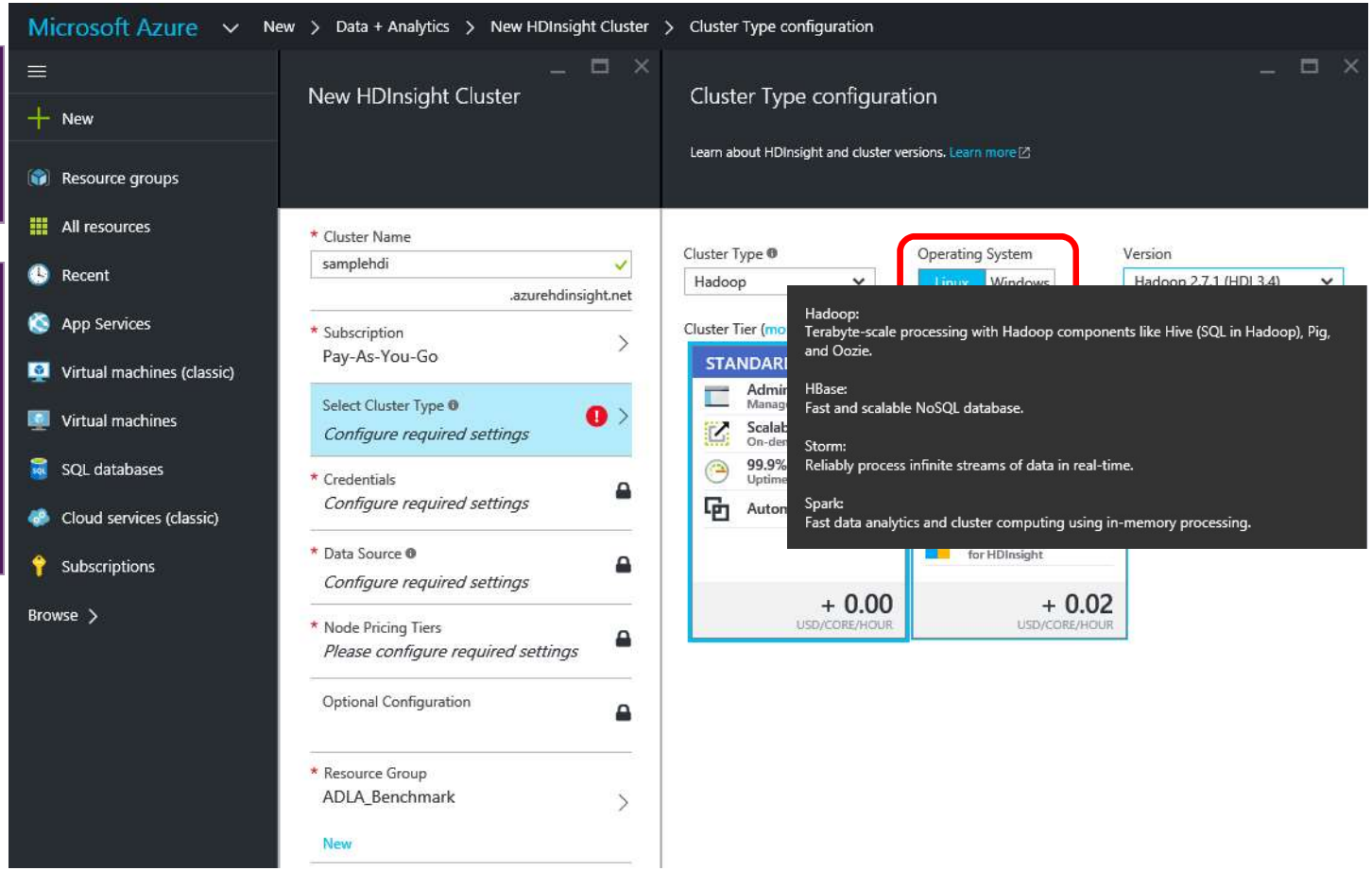
- Type of Hadoop cluster
- OS (Linux or Windows)
- Hadoop Version
- Azure storage data source
- Number and size of nodes i.e. head nodes, worker nodes etc)
  - The actual types of nodes depends on cluster types
- Security credential for accessing web/REST APIs and for SSH
- Optional metadata store
- Azure Virtual Network
- Script Action for customization



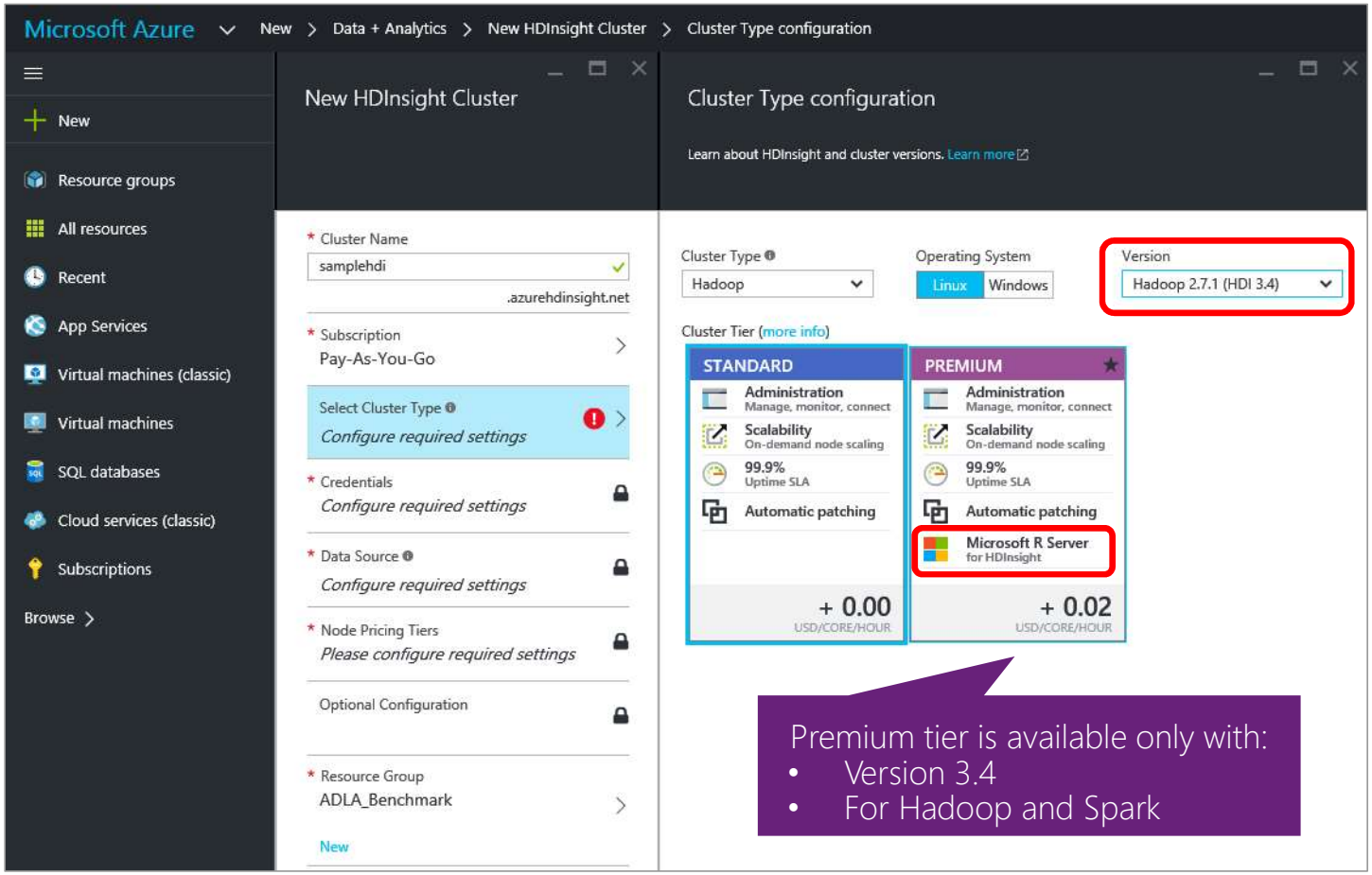
Step 1: Specify Cluster Type and OS

OS choices are:
- Windows
- Linux

Cluster type choices are:
- Hadoop
- HBase
- Storm
- Spark



Step 2 :Specify Version and Cluster Tier

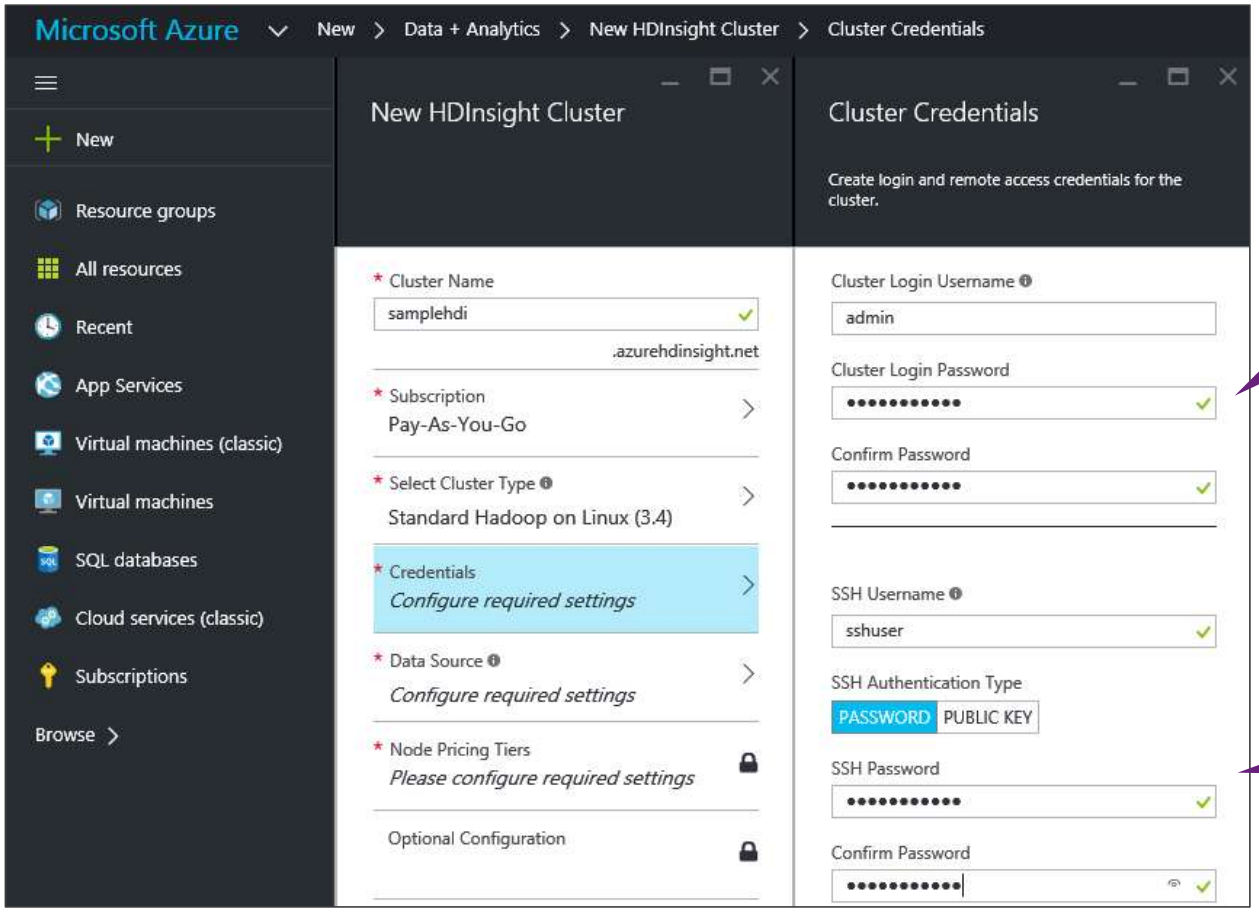


Other supported versions are 3.3 and 3.2

Premium tier is available only with:

- Version 3.4
- For Hadoop and Spark

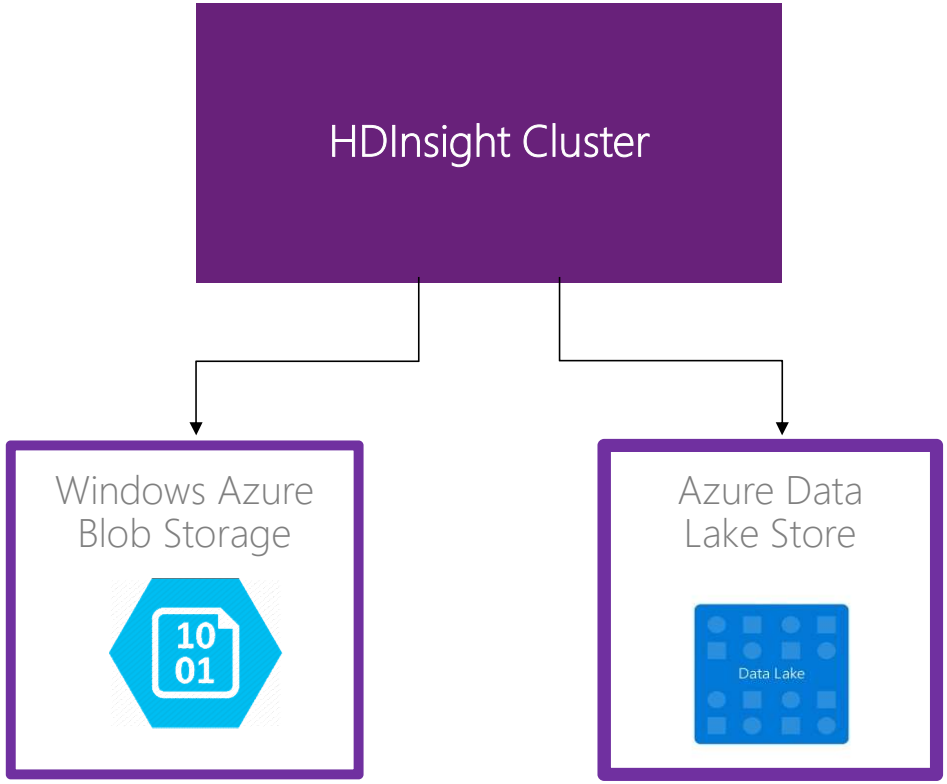
Step 3: Specify SSH and Admin Credentials



Credentials to submit jobs to the cluster and the Ambari Dashboard

Credentials to remotely access the cluster

Two storage options: WASB or ADLS

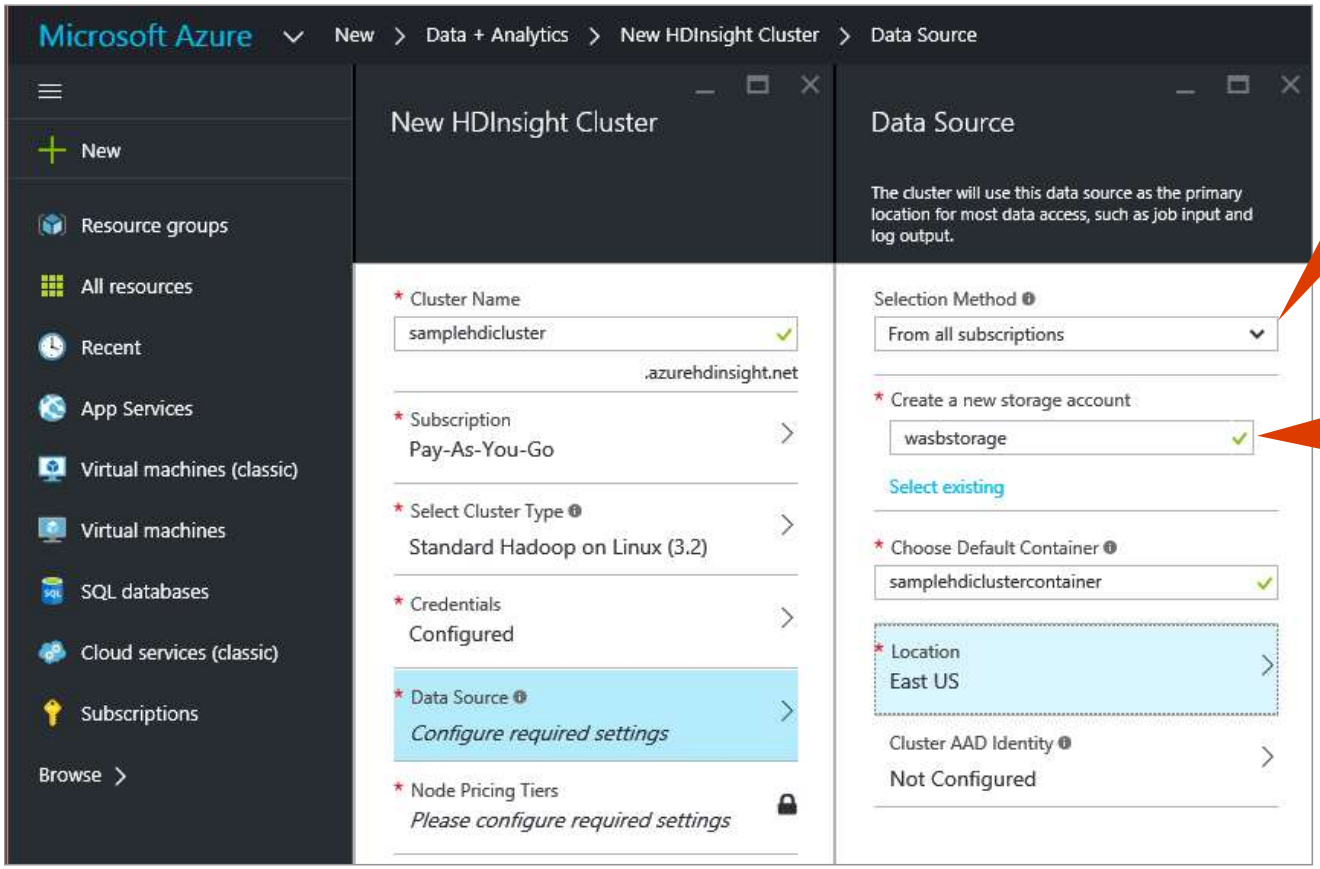


For **Hadoop Clusters**, ADLS can only be used an additional storage account. The default is still WASB.

For **Storm clusters** ADLS can be used to write data from a Storm topology. Data Lake Store can also be used to store reference data that can then be read by a Storm topology.

For HBase clusters ADLS can be used as a default storage or additional storage—available only with HDI version 3.2

Step 4(1): Specifying WASB for Storage

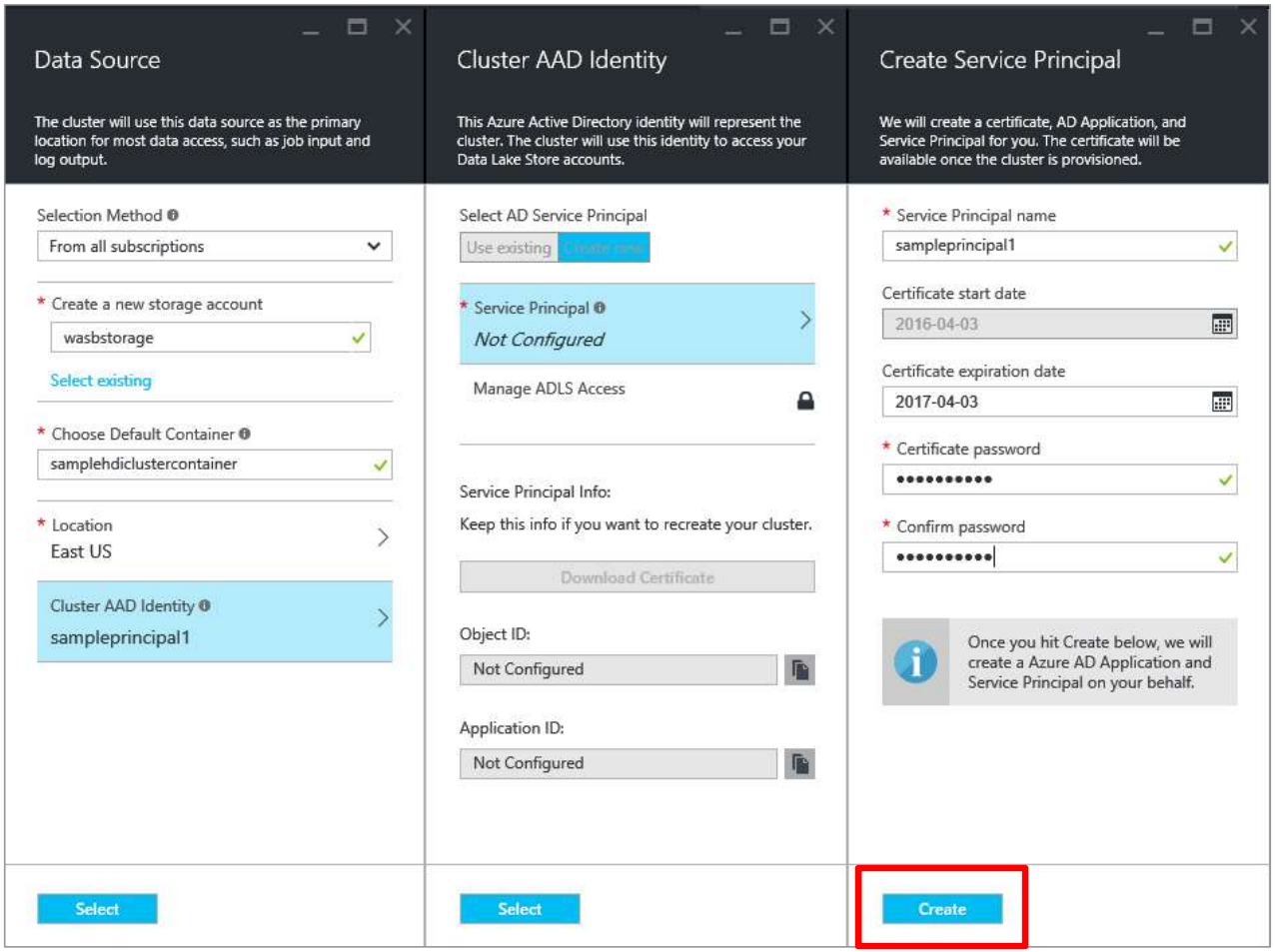


Choose a storage account from all your subscriptions or specify the storage account name and access key

You can specify an existing storage account and container or have a new one created for you.

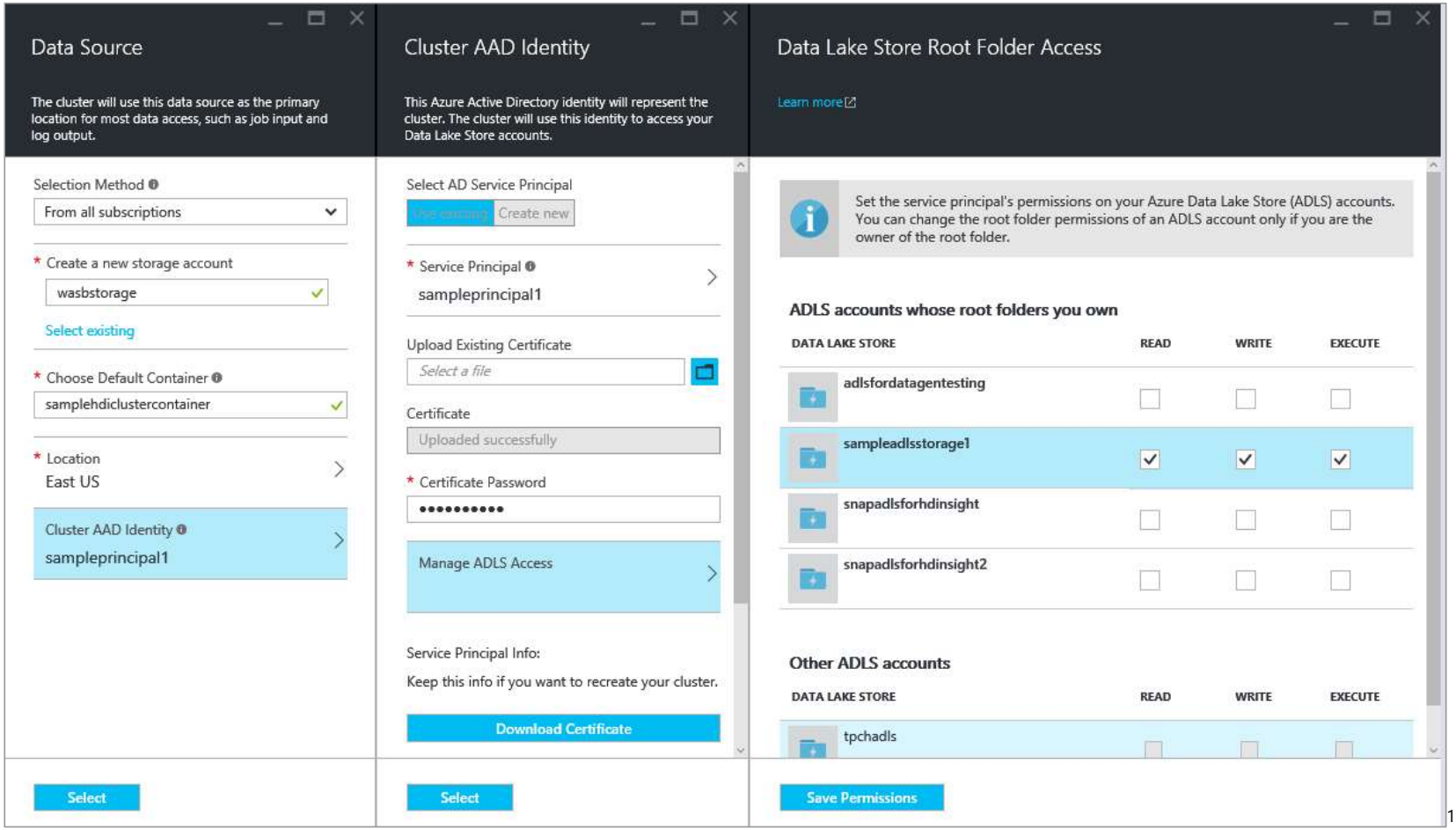
Step 4(2): Specifying ADLS for Storage

Step1: Create a Service Principal (Azure Active Director ([AAD] Identity) that can represent the cluster



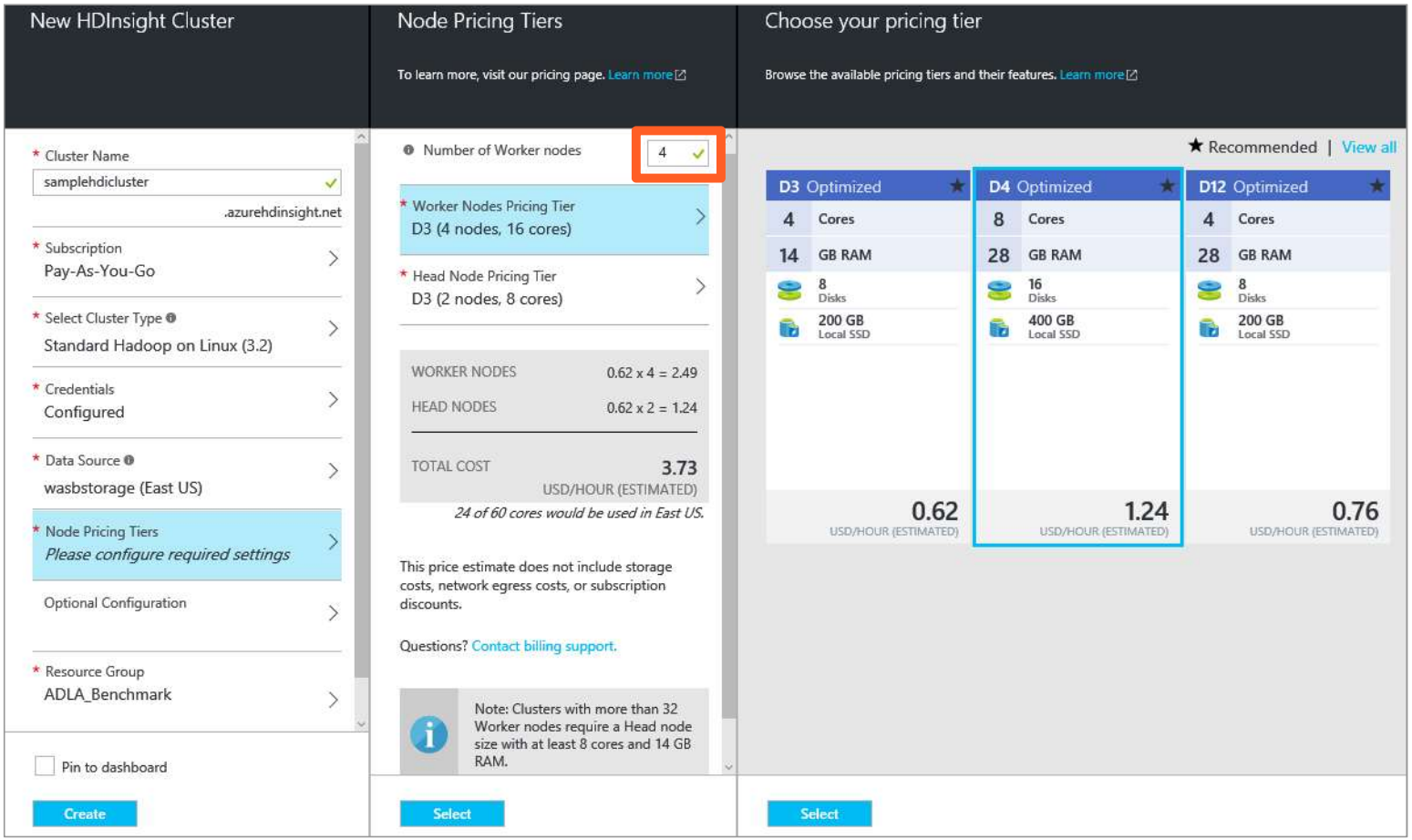
Step 4(2): Specifying ADLS for Storage

Step2: Grant READ, WRITE and EXECUTE permissions to the Service Principal on the desired ADLS storage account.



Step 5: Specify Cluster Configuration

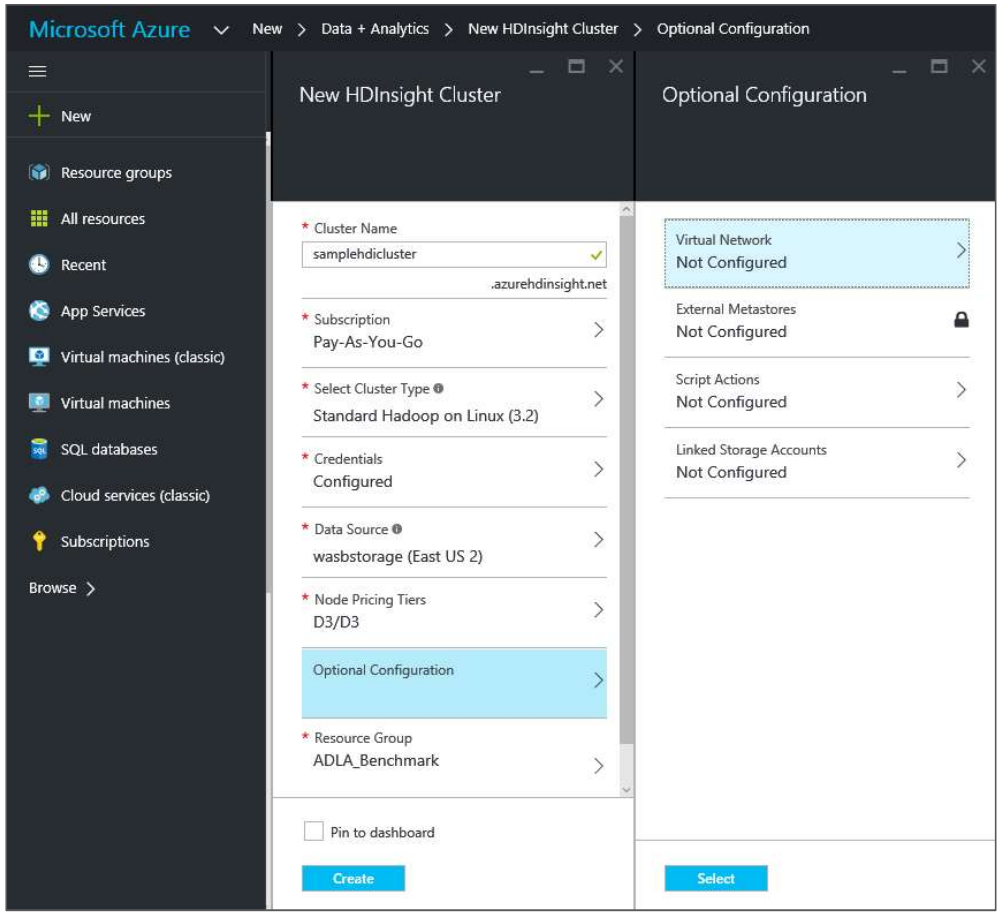
Specify the number of worker nodes and the VM instance type for worker and head nodes



Step 6: Optional Configurations

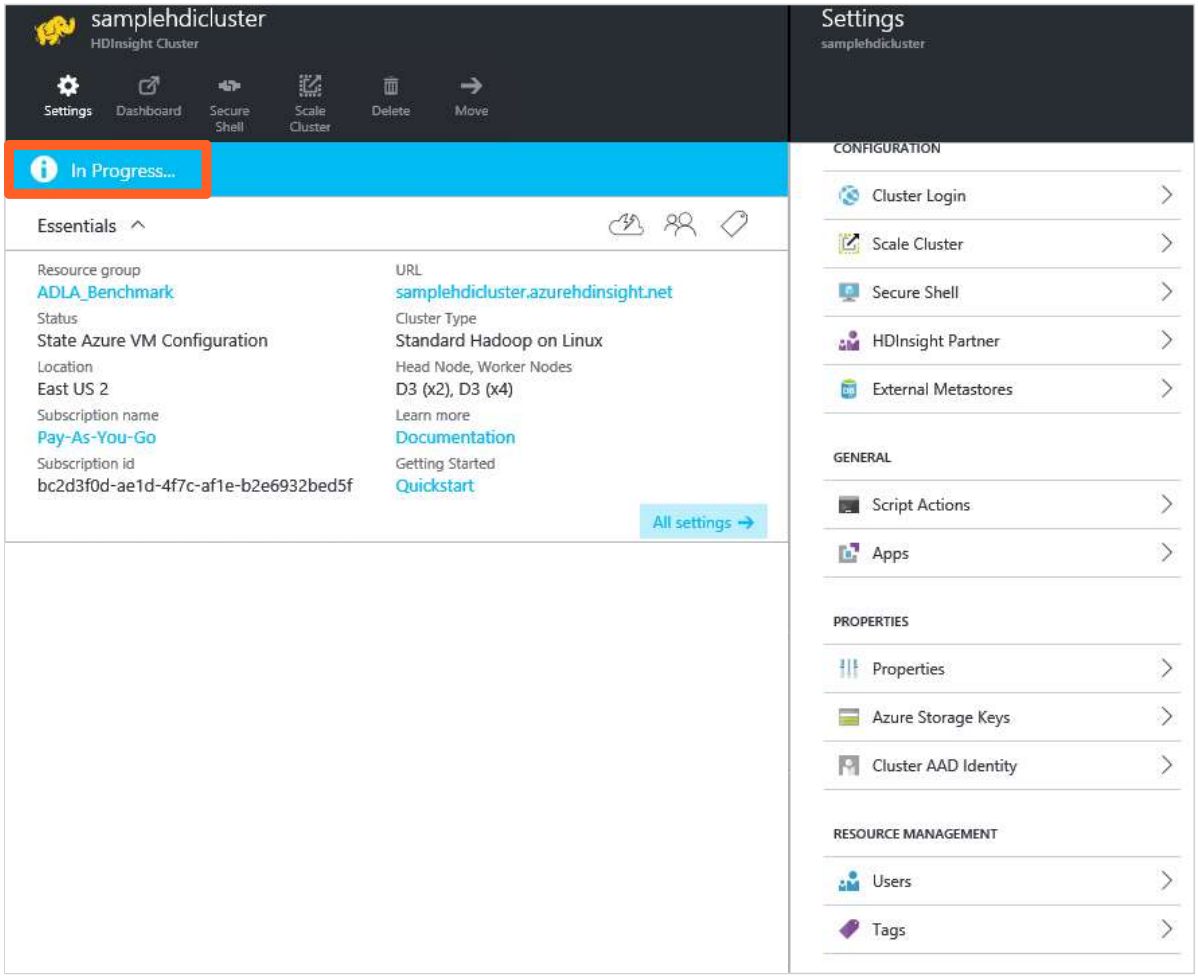
Optionally you can configure:

- Virtual Network
- External Metastores
- Script Actions
- Linked Storage Accounts



Cluster Creation

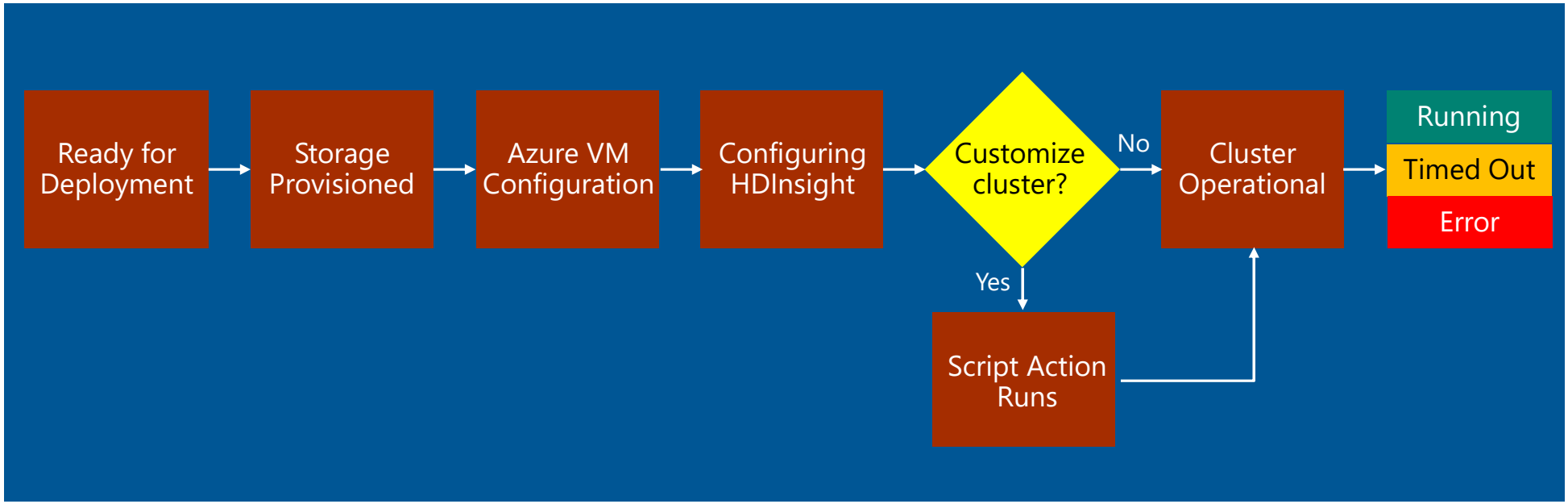
Provisioning and configuring the cluster according to specification can take between 5 and 15 minutes.



Script Actions

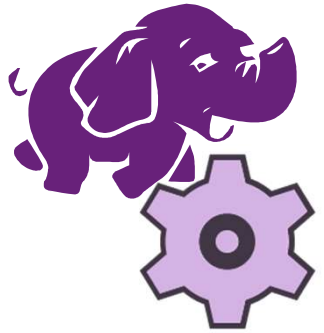
Customize with Script Actions

Script Actions enable clusters to be customized during creation using custom scripts: Clusters configuration can be changed or additional software installed.



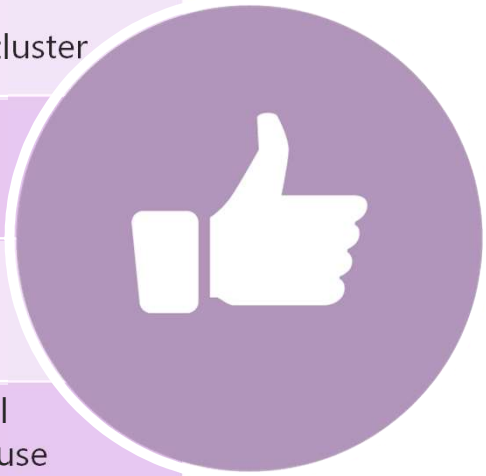
Script Actions: Key concepts

- Script actions are Bash scripts that run when HDInsight is being configured.
- Scripts run in parallel on all the specified nodes in the cluster.
  - A script can be ran on the head nodes, the worker nodes, or both.
- Script actions must complete within 60 minutes, or they will timeout
- Each cluster can accept multiple script actions that are invoked in the order in which they are specified.
- **Script Action scripts can be used from:**
  - **The Azure Portal**
  - **Azure PowerShell**
  - **The HDInsight .NET SDK**



Script Action: Best Practices

Target the right Hadoop version	Different versions of HDInsight have different versions of Hadoop services and components installed
Provide stable links to script resources:	All of the scripts and resources used by the script should remain available throughout the lifetime of the cluster
Use pre-compiled resources:	To minimize the time it takes to run the scripts
Ensure script idempotency	As nodes of an HDInsight cluster will be re-imaged during the cluster lifetime
Configure the custom components to use Azure Blob storage	On a cluster re-image, the HDFS file system gets formatted and all data that is stored there will be lost. Change the configuration to use Azure Blob storage (WASB) instead
Write information to STDOUT and STDERR	So the information is logged, and can be viewed after the cluster has been provisioned by using the Ambari web UI



Provided Scripts

HDInsight provides Script Action scripts to install additional software

Software	Script
Hue	https://hdiconfigactions.blob.core.windows.net/linuxhueconfigactionv01/install-hue-uber-v01.sh [See Install and use Hue on HDInsight clusters]
Spark	https://hdiconfigactions.blob.core.windows.net/linuxsparkconfigactionv02/spark-installer-v02.sh [See Install and use Spark on HDInsight clusters]
R	https://hdiconfigactions.blob.core.windows.net/linuxrconfigactionv01/r-installer-v01.sh [See Install and use R on HDInsight clusters]
Solr	https://hdiconfigactions.blob.core.windows.net/linuxsolrconfigactionv01/solr-installer-v01.sh [See Install and use Solr on HDInsight clusters]
Giraph	https://hdiconfigactions.blob.core.windows.net/linuxgiraphconfigactionv01/giraph-installer-v01.sh [See Install and use Giraph on HDInsight clusters]
Hive libraries	https://hdiconfigactions.blob.core.windows.net/linuxsetupcustomhivelibsv01/setup-customhivelibs-v01.sh [See Add Hive libraries on HDInsight clusters]

Creating a cluster with .NET SDK

Creating a cluster with .NET SDK

Code to create HDI cluster:

- Linux OS
- Hadoop
- 15 worker nodes
- "EAST US 2" location

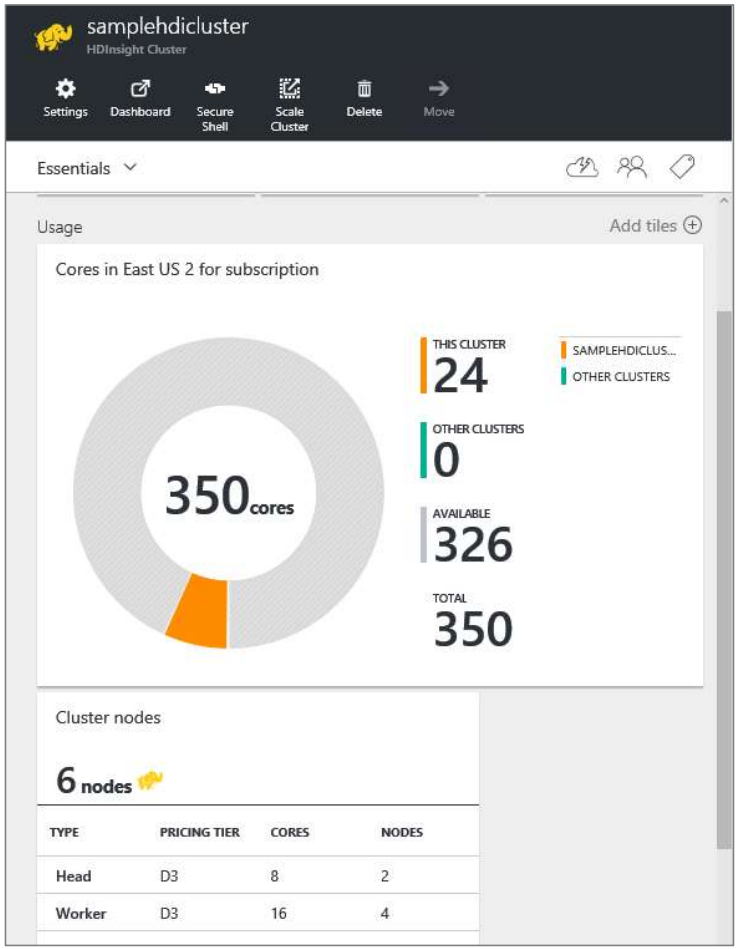
```
var tokenCreds                  = GetTokenCloudCredentials(); //See notes section for definition of this function
var subCloudCredentials         = GetSubscriptionCloudCredentials (tokenCreds, "My Subscription ID");
var resourceManagementClient    = new ResourceManagementClient(subCloudCredentials);
var rpResult                    = resourceManagementClient. Providers.Register("Microsoft.HDInsight");
_hdiManagementClient            = new HDInsightManagementClient (subCloudCredentials);
//specify the cluster configuration details
var parameters                  = new ClusterCreateParameters  {
        ClusterSizeInNodes              = 15,
        ClusterType                     = HDInsightClusterType.Hadop,
        OSType                          = OSType.Linux,
        Version                         = "3.2",
        DefaultStorageAccountName       = "mystorageaccount.blob.core.windows.net",
        DefaultStorageAccountKey        =  "my-storage-key",
        DefaultStorageContainer         =   "HDInsightContainer",
        ClusterUserName                 =  "admin",
        Password                        = "MyPassword"
        Location                        = "EAST US 2",
        SshUserName                     = "sshuser",
        SshPublicKey                    = @"---- BEGIN SSH2 PUBLIC KEY ----
                                          mPCsJVGQLu6O1wqcxRqiKk7keYq8b
                                          P5s30v6bIljsLZYTnyReNUa5LtFw7eauGr
                                          ---- END SSH2 PUBLIC KEY ----";
};
//Now create the cluster
_hdiManagementClient.Clusters.Create("MyResourceGroup", "MySampleCluster", parameters);
```

Microsoft

Cluster: Resource Usage Overview

The Azure Portal provides a report on:

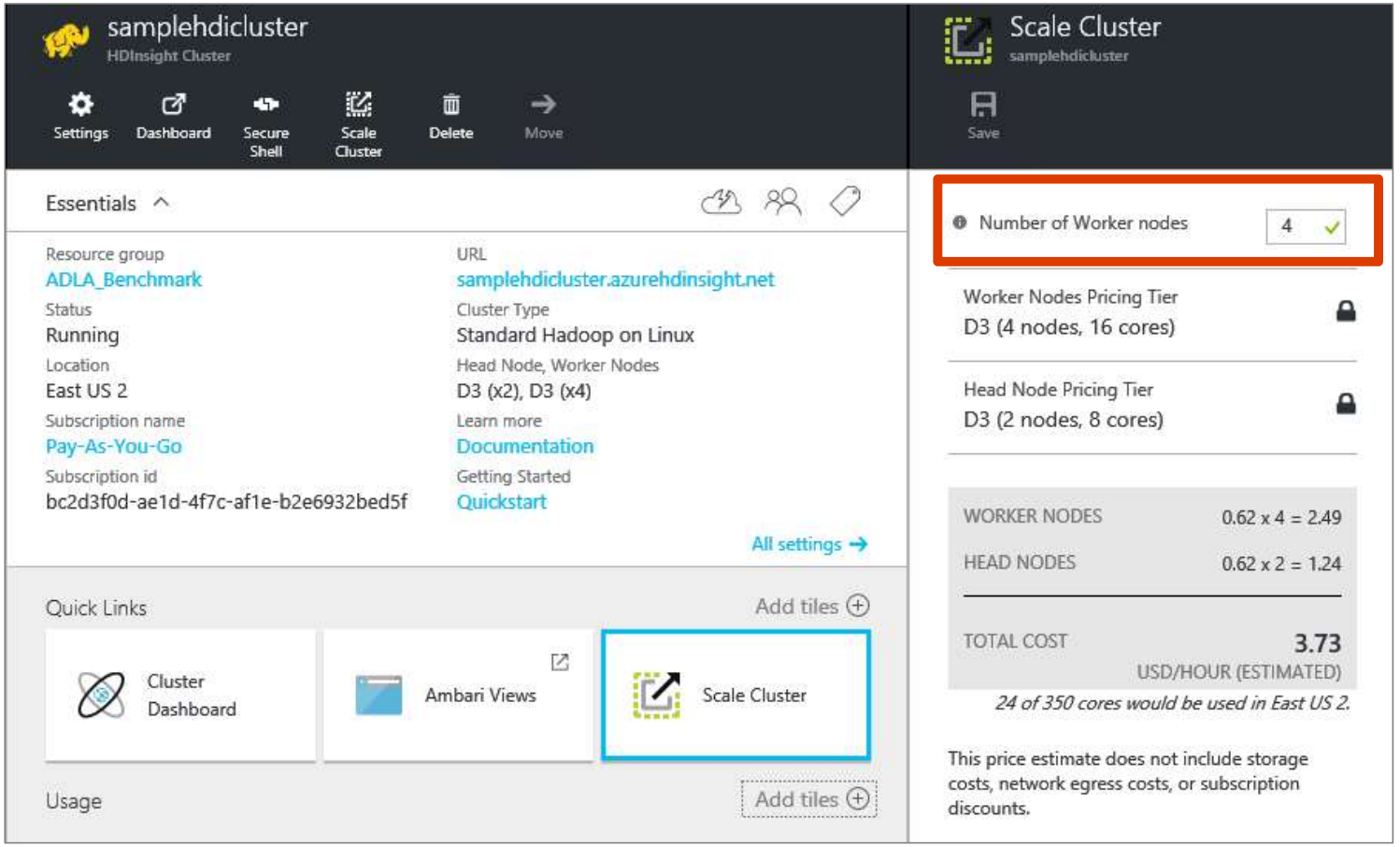
- # of cores consumed by this cluster and other clusters
- # of cores available for additional clusters



Cluster reconfiguration

After the cluster has been created, you can dynamically change (increase or decrease) the number of Worker nodes.

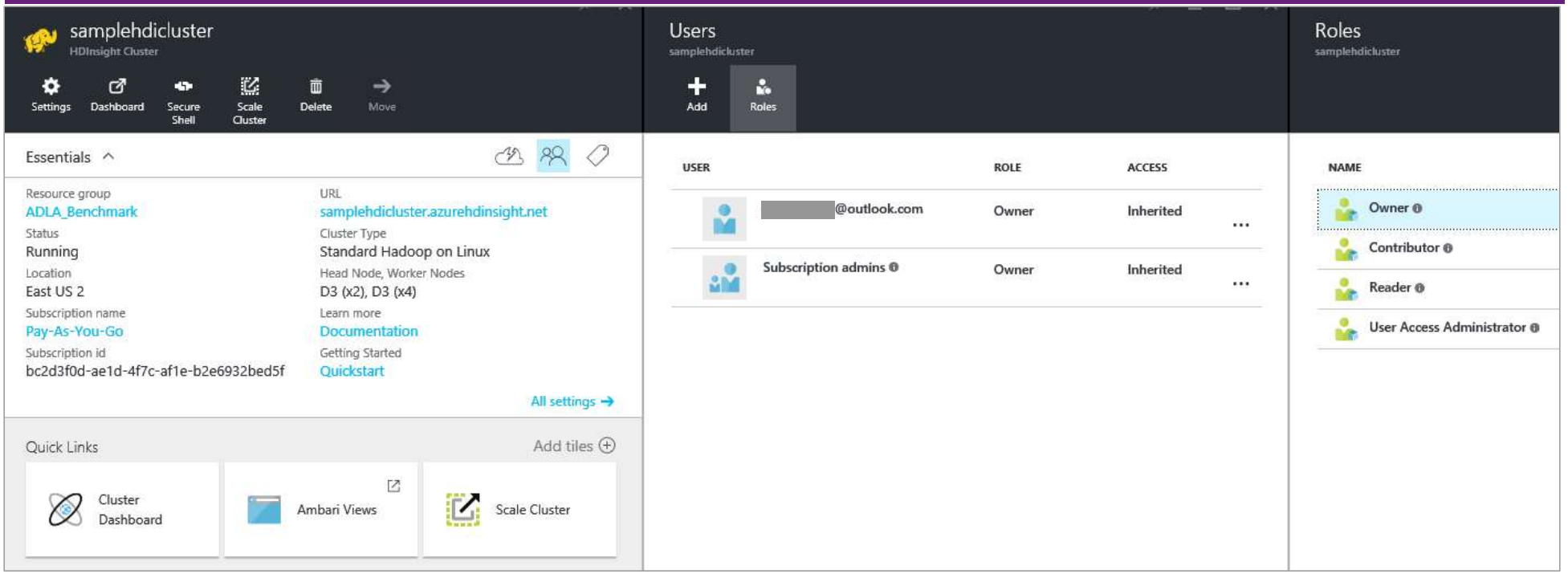
Note: The VM instance type *cannot* be changed.



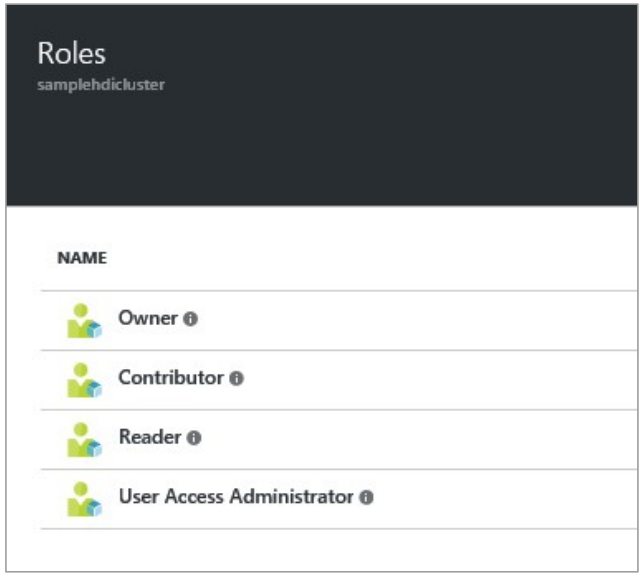
Post-creation Actions

Security: Role-based Access

New users can be added in the role of “Owner”, “Contributor”, Reader or “User Access Administrator”
Users can be added or deleted at anytime

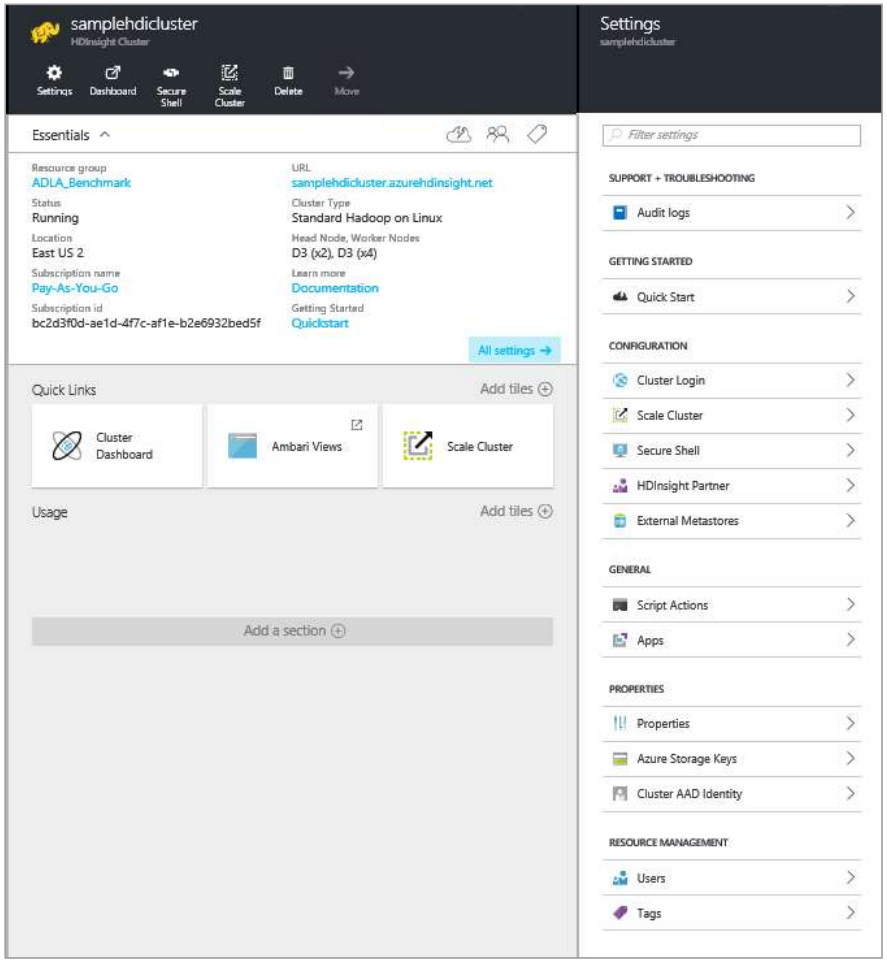


Security: Roles and Privileges



Role	Privilege
Owner	Lets you manage everything
Contributor	Lets you manage everything except access to resources
Reader	Lets you view everything but not make changes
User Access Administrator	Lets you manage user access to Azure resources

HDInsight Cluster Settings



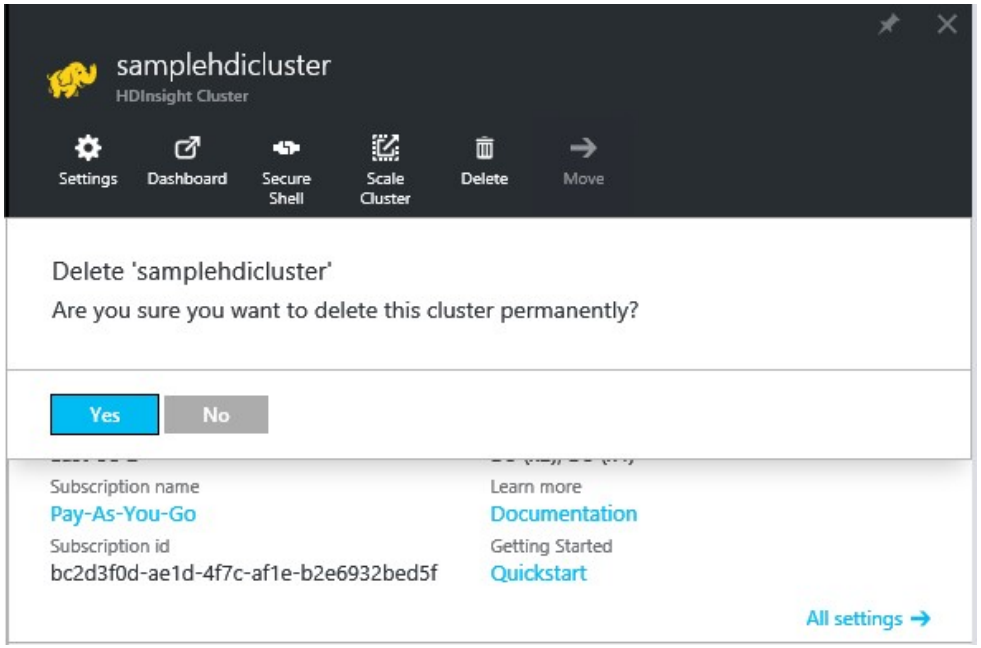
The Azure Portal lets you view and change all these settings after the cluster has been created

Deleting a HDI Cluster

A running cluster can be deleted permanently freeing up the used cores.

Freed cores can be used to create a new cluster or expand an existing one.

Storage (WASB or ADLS) must be deleted separately

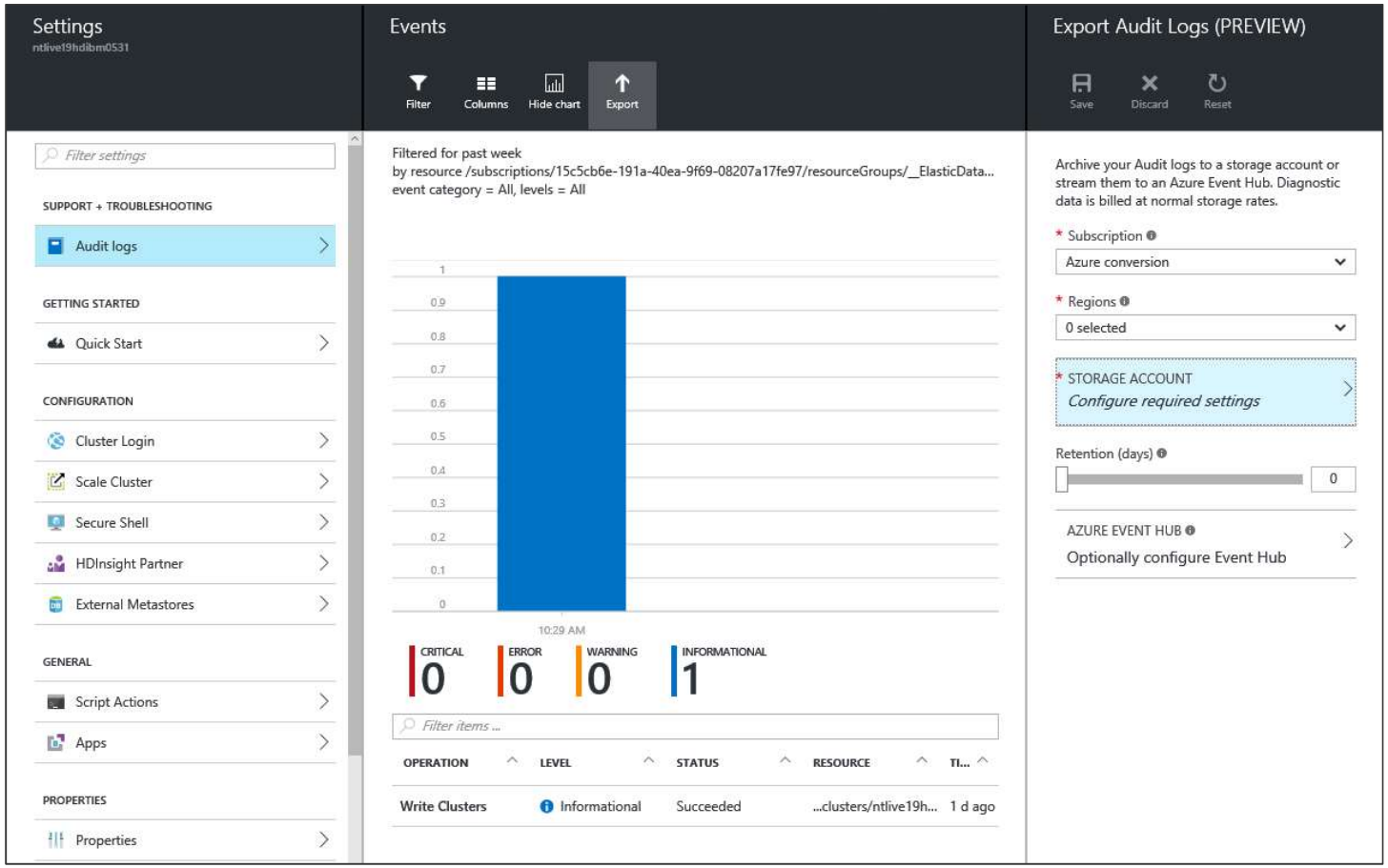


Audit Logs

Audit Logs

Audit Logs shows ***Critical, Error, Warning*** and ***Informational*** events

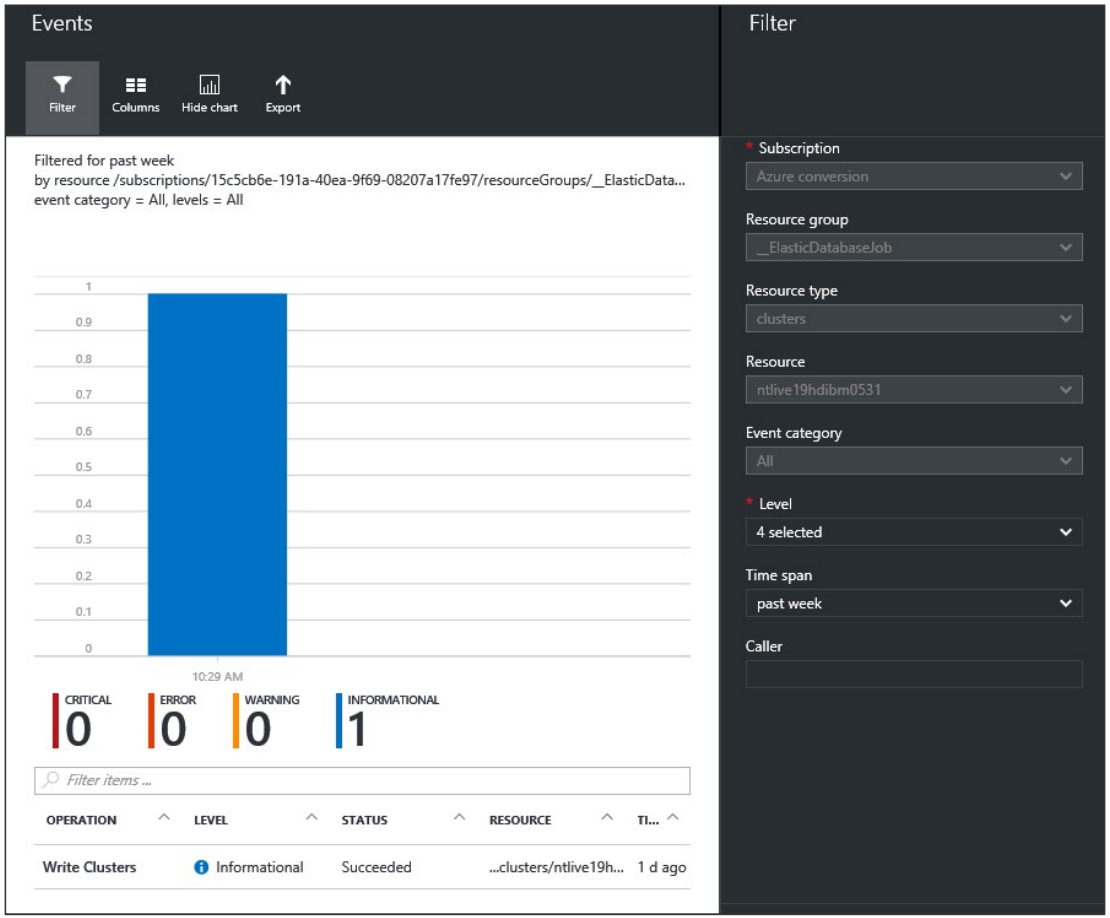
Audit logs can be archived into Azure storage or stream to Azure Event Hub

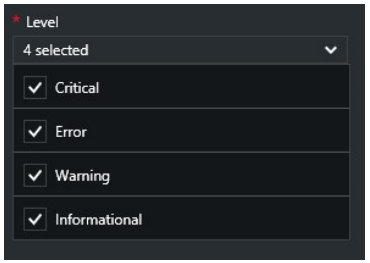


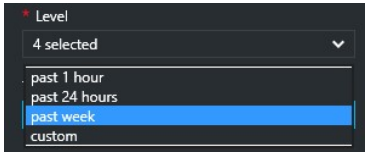
Filtering Audit Logs

Audit Logs entries can be filtered by:

- Time
- Type
- Level
- …







Microsoft

Configuring Hadoop

Core Hadoop Configuration files

Administrators configure settings for HDFS, Yarn and MapReduce (and other services) through these files

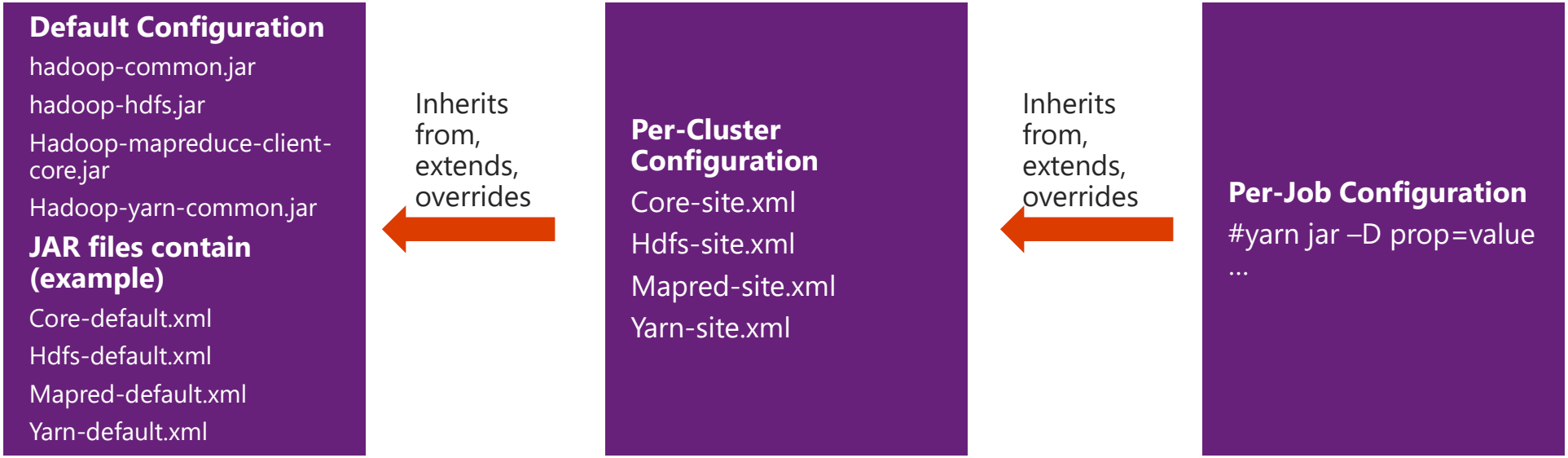
File Name	File Format	File Purpose
core-site.xml	Hadoop configuration XML	Hadoop core configuration settings that can be used by HDFS, YARN MapReduce and others
hdfs-site.xml	Hadoop configuration XML	HDFS configuration settings (NameNode and DataNode)
yarn-site.xml	Hadoop configuration XML	YARN configuration setting
Mapred-site.xml	Hadoop configuration XML	MapReduce configuration settings
Hadoop-evn.sh	Bash script	Environment variables used by various Hadoop scripts and programs
log4j.properties	Java properties	System log file configuration settings
Hadoop-metrics2.properties	Java properties	Metrics publishing configuration settings.

Note: These files also define what should be recorded to the log files and how to process those log files. *Many of these settings can be configured using the Ambari Web UI* (details in later slides)

MR1

Configuration Precedence

The actual configuration for any job running on a cluster is derived from a combination of sources including the default configuration, the per-cluster or per-node configuration, and the per-job configuration.



*Note: Cluster nodes with different hardware configurations commonly need different *-site.xml files*

Slide 40

MR1 Madhu Reddy, 4/11/2016

Configuration: Final Properties

To prevent user applications from overriding a configuration property value, an administrator can declare the property value as ***final***.

- User applications may specify their own configuration settings when they are submitted to a cluster. In some cases, a user could choose a configuration setting that unfairly consumes a resource and negatively effects the performance other user applications.
- To prevent this, an administrator can declare a configuration property value as final. This prevents any user application from overriding a property's value.

Either the Ambari Web UI or a command-line editor can be used to make property settings final.


```
<property>
      <name> dfs.datanode.data.dir </name>
      <value> /hadoop/hdfs/data </value>
      <final> true </final>
</property>
```

Configuration Management Options

Option	Description	Benefit
Ambari Web UI	Browser-based graphic user management interface	Ease of use, pre-built and ready-to-go
REST APIs: Ambari, WebHDFS, YARN etc	Use HTTP verbs (GET, PUT, POST, DELETE) management interface	Integration with other web-based management interfaces.
Manual Editing	Manually edit and distribute configuration files, manually restart services	No reliance on a GUI, no need to install Ambari. [Not compatible with Ambari management]
Command-line	Per-framework command-line management utilities	Scriptable, no reliance on a GUI

In an Ambari-managed cluster it is recommended to exclusively use Ambari—using other management method may cause conflicts.

Monitoring and Managing Hadoop with Ambari Web UI

Apache Ambari: What is it?

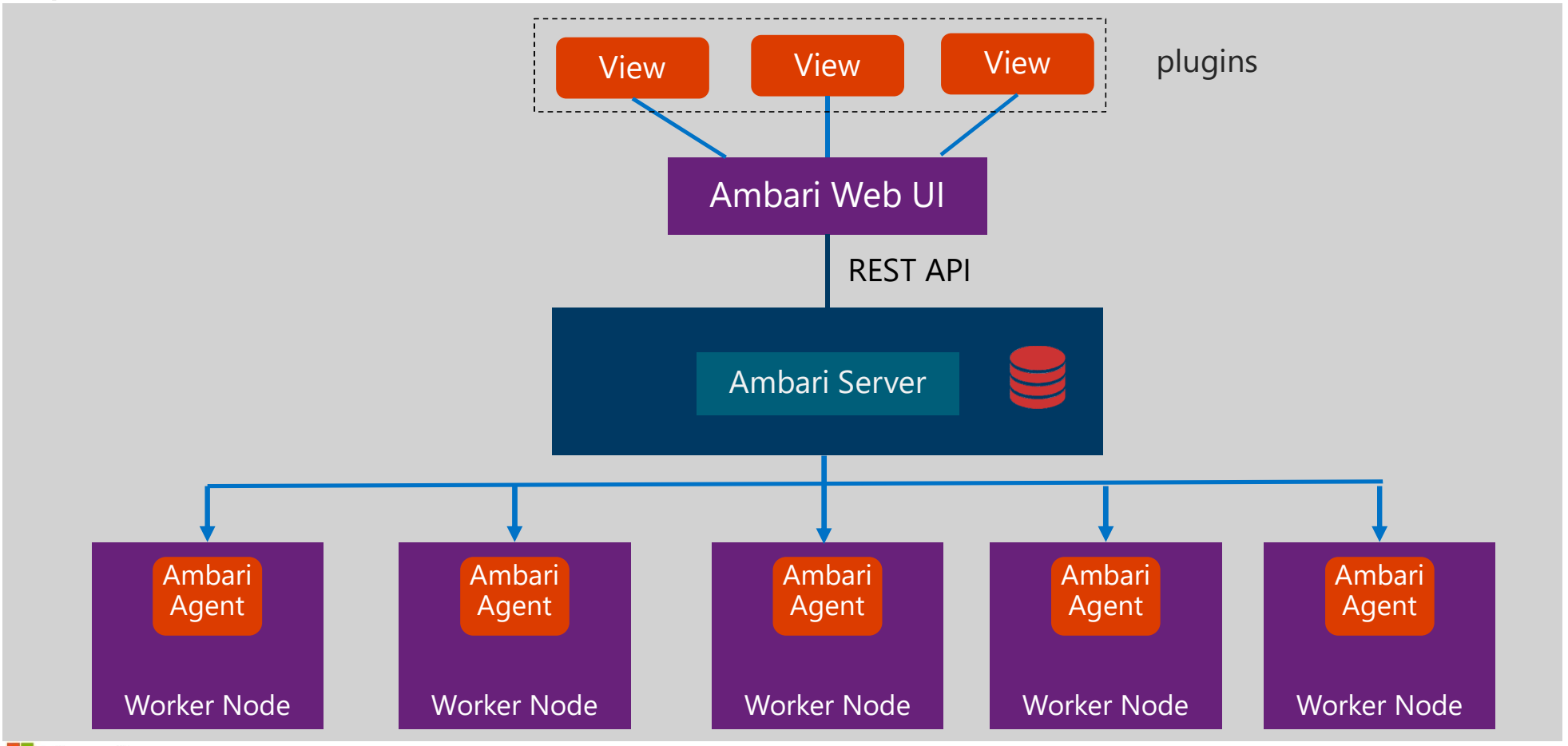
A 100% open source framework for provisioning, managing and monitoring Apache Hadoop clusters

Systems Administrators	**Provisioning**	Provides step-by-step wizard for installing Hadoop services across any number of hosts
		Handles configuration of Hadoop services for the cluster
	Managing	Provides central management for starting, stopping, and reconfiguring Hadoop services across the entire cluster
	Monitoring	Provides dashboard for monitoring health and status of the Hadoop cluster
		Leverages Ambari Metrics System for metrics collection
		Leverages Ambari Alert Framework for system alerting and will notify you when your attention is needed (e.g., a node goes down, remaining disk space is low, etc)
Application Developers and System Integrators	Can easily integrate Hadoop provisioning, management, and monitoring capabilities to their own applications with the Ambari REST APIs.	

Ambari: Management Features Overview

- Interactive Wizard Driven cluster Installation
- Non-interactive API-driven cluster installation
- Granular control of cluster services start up and shut down
- Cluster service configuration management
- Dashboard cluster monitoring with alerts
- REST API for integration with other vendors
- Ambari Views for custom plug-in

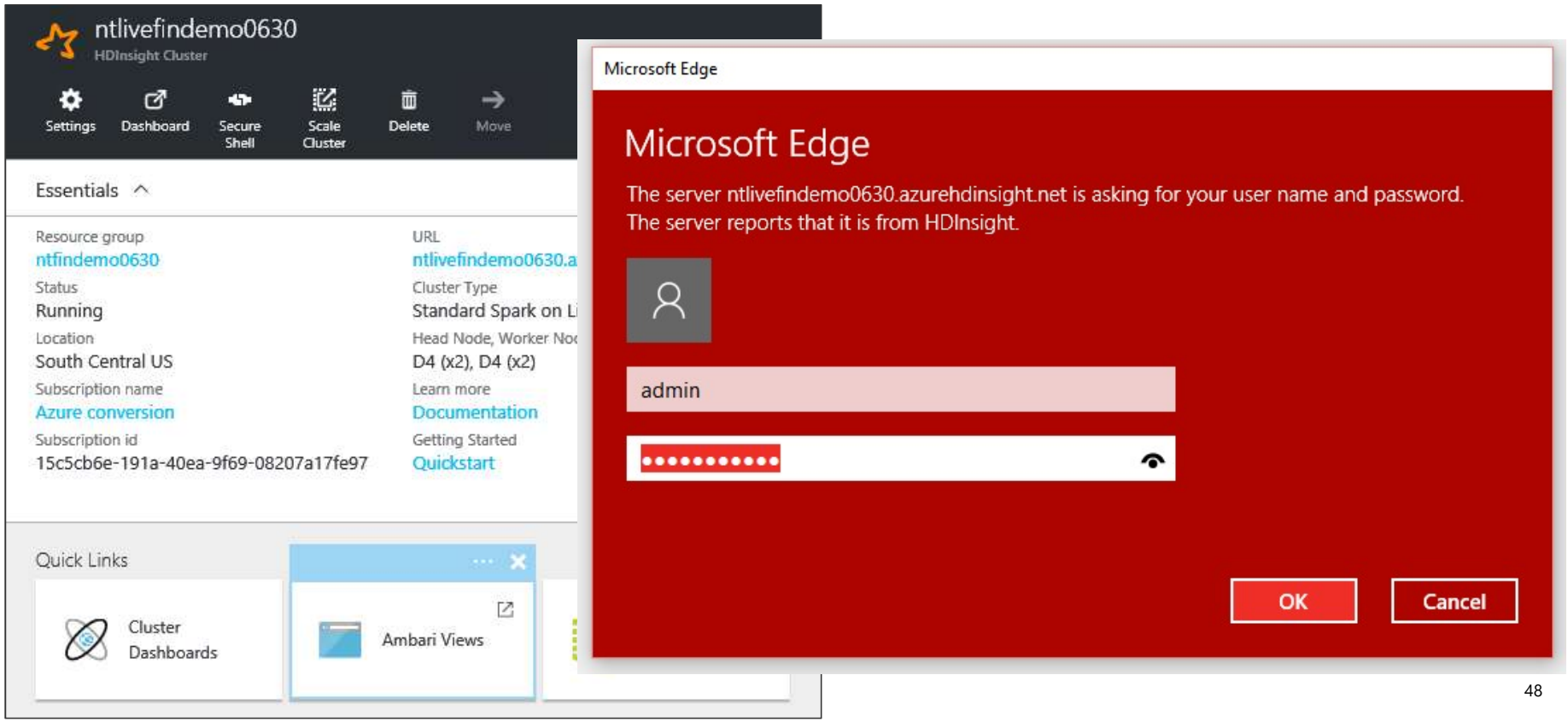
Apache Ambari: Architecture Overview



Managing HDInsight with Amabari

HDInsight and Ambari

The Ambari Web UI can be launched directly from the Azure HDInsight Portal



Ambari Web UI Dashboard

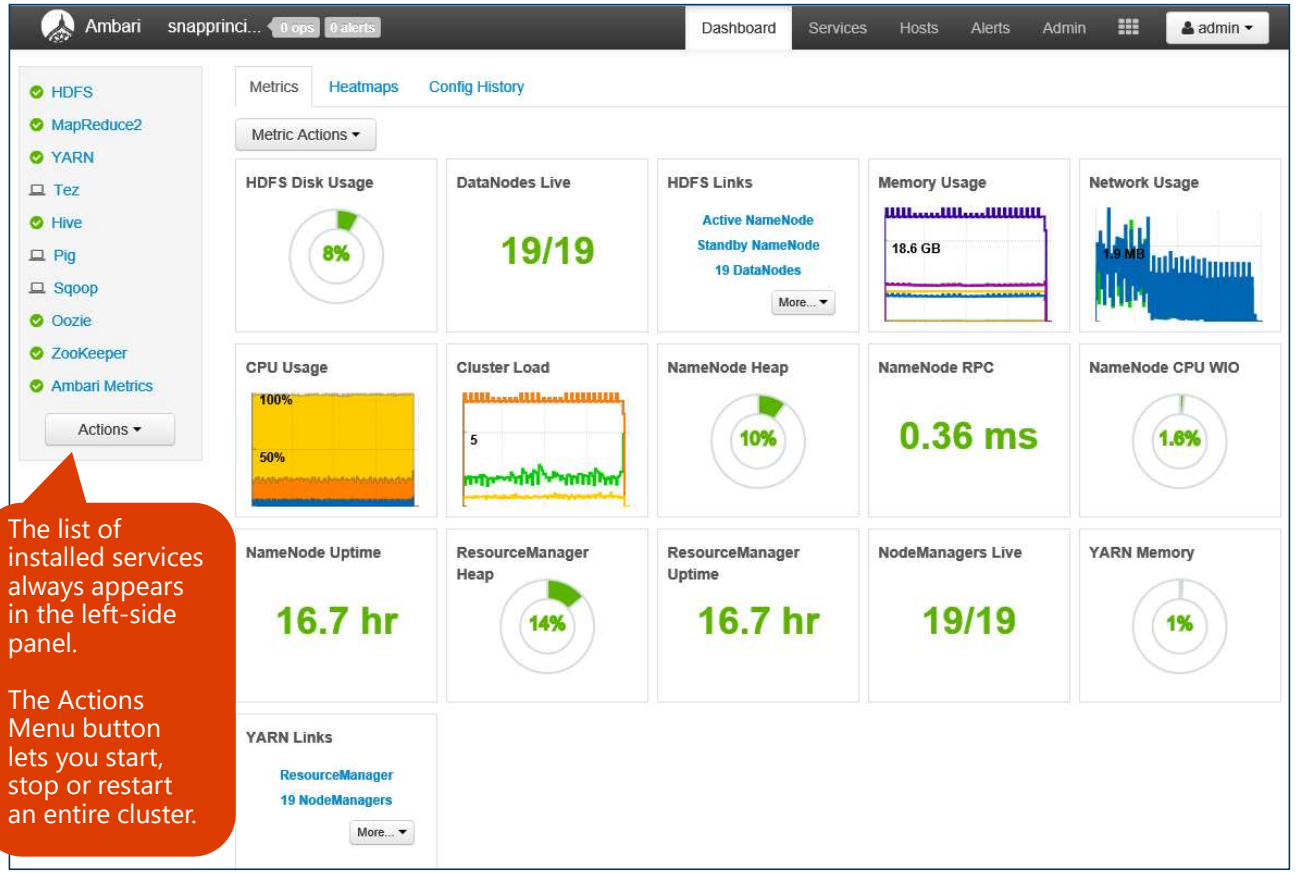
Ambari Web UI: Dashboard Metrics

The Metrics tab of the Dashboard page displays cluster-level system metrics including:

- CPU Usage
- HDFS Disk Usage
- Memory Usage
- Network Usage
- …

Dashboard enables you to understand the state of the cluster at-a-glance.

The dashboard look can be customized by adding and removing Widgets



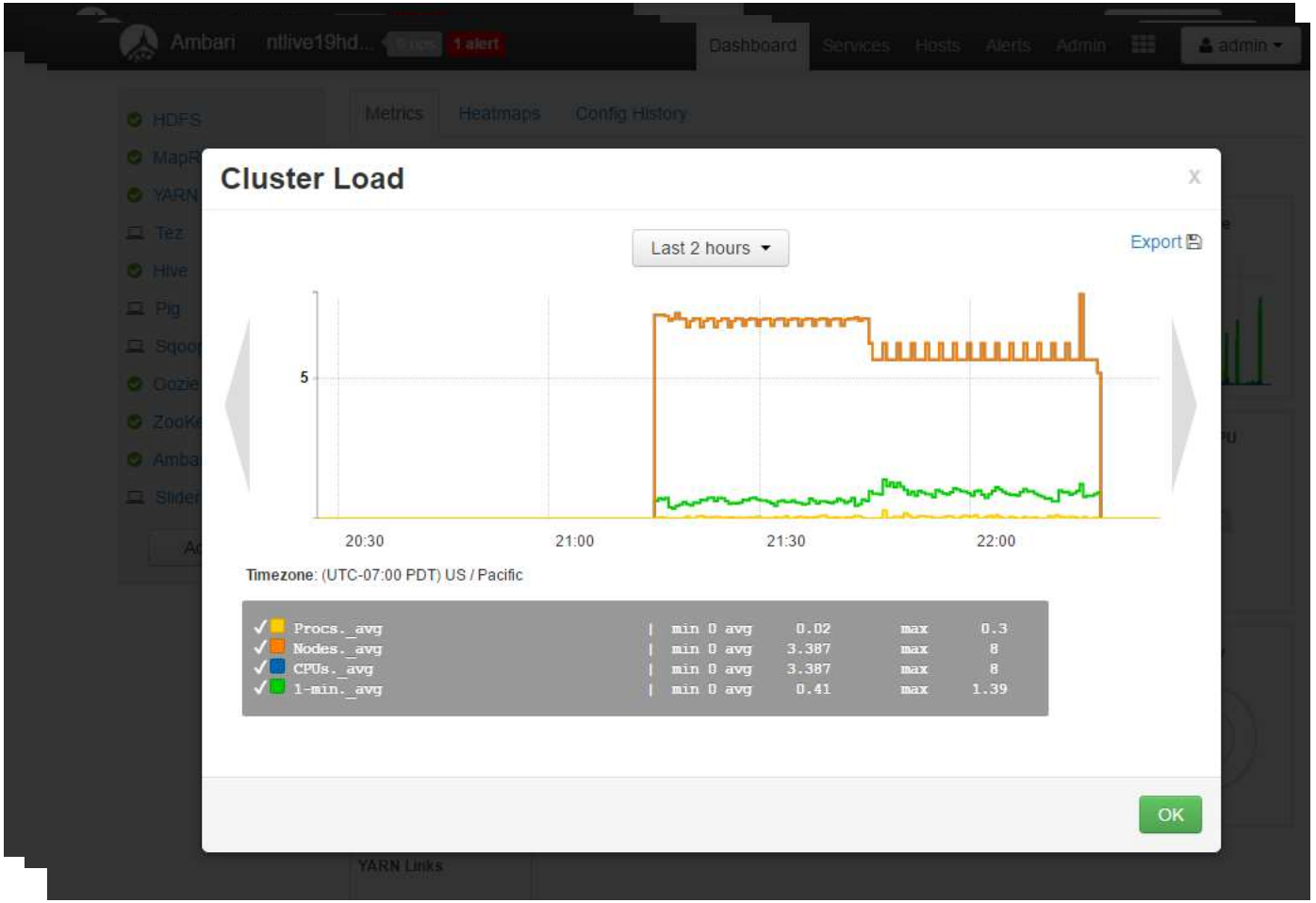
Ambari: Dashboard Metrics Drilldown

You can drilldown to get more details on

- CPU Usage
- Cluster Load
- Network Usage
- Memory Usage

The usage stats can be viewed over any custom period.

For other metrics you see additional info by hovering over the Widget.

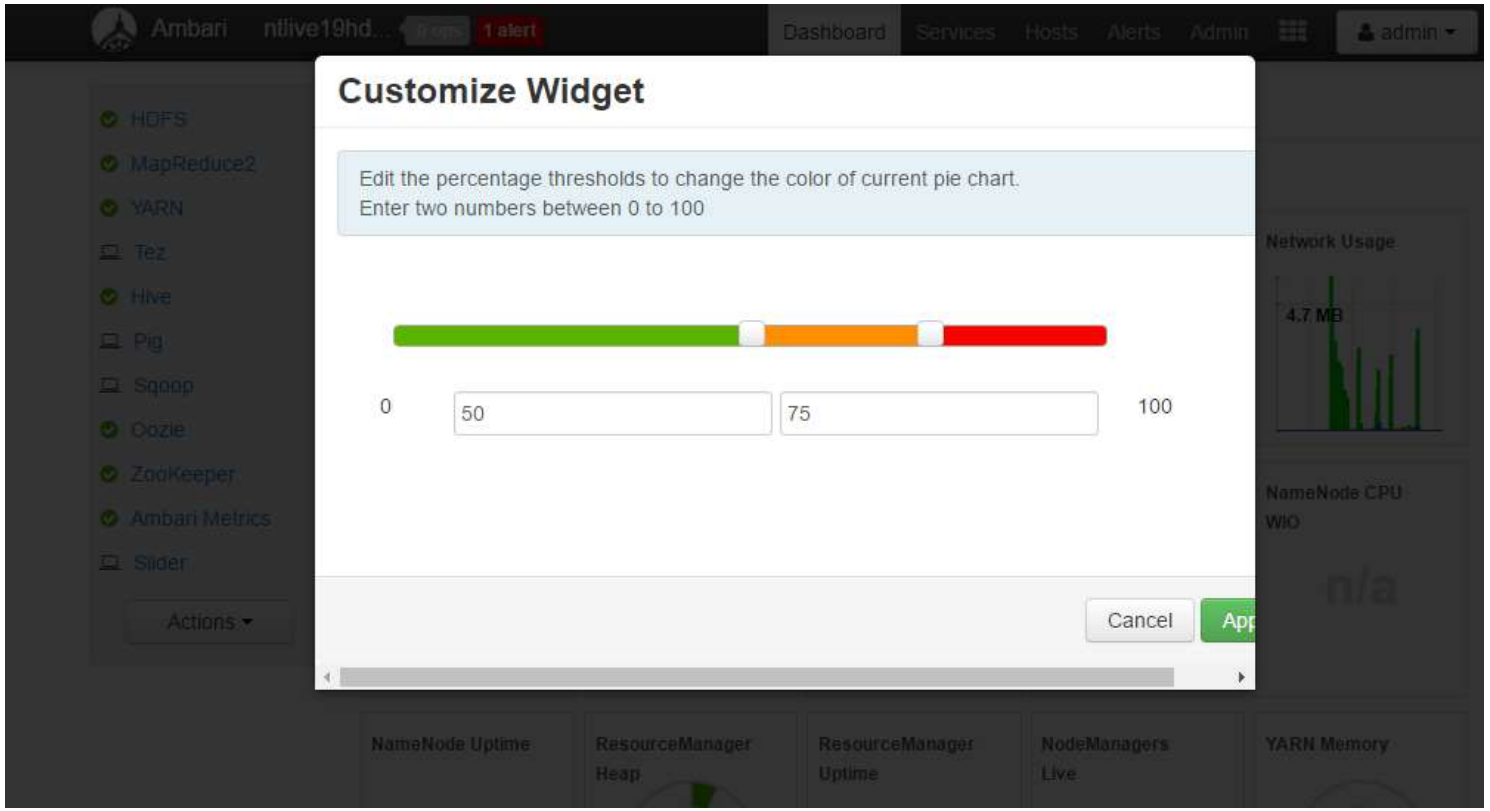


Ambari: Dashboard Widget Customization

For these Widgets:

- YARN Memory
- Node Managers
- Resource Managers
- NameNode CPU
- NameNode RPC
- NameNode Heap

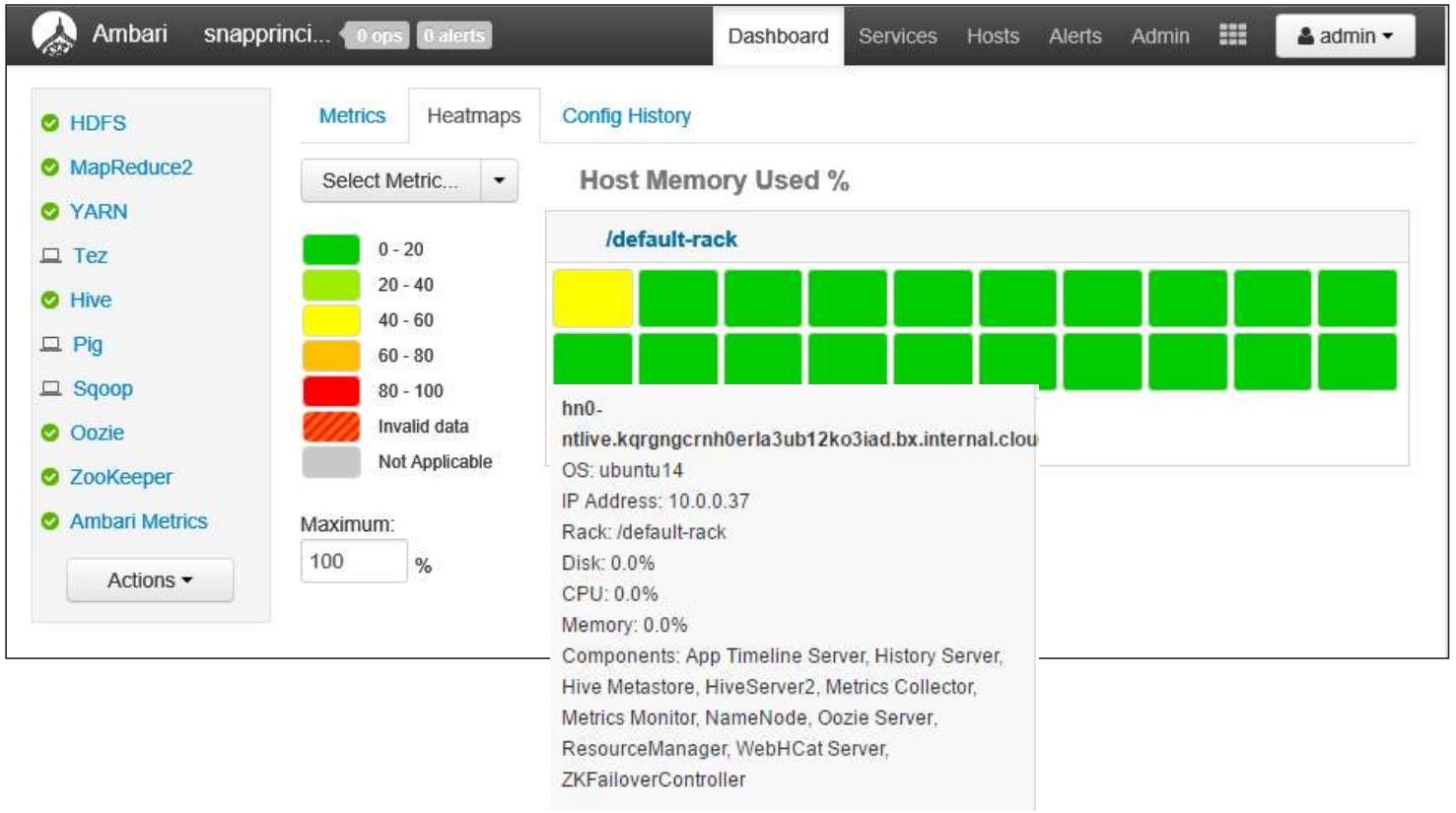
The color can be customized by configuring the % thresholds



Ambari Dashboard: HeatMap

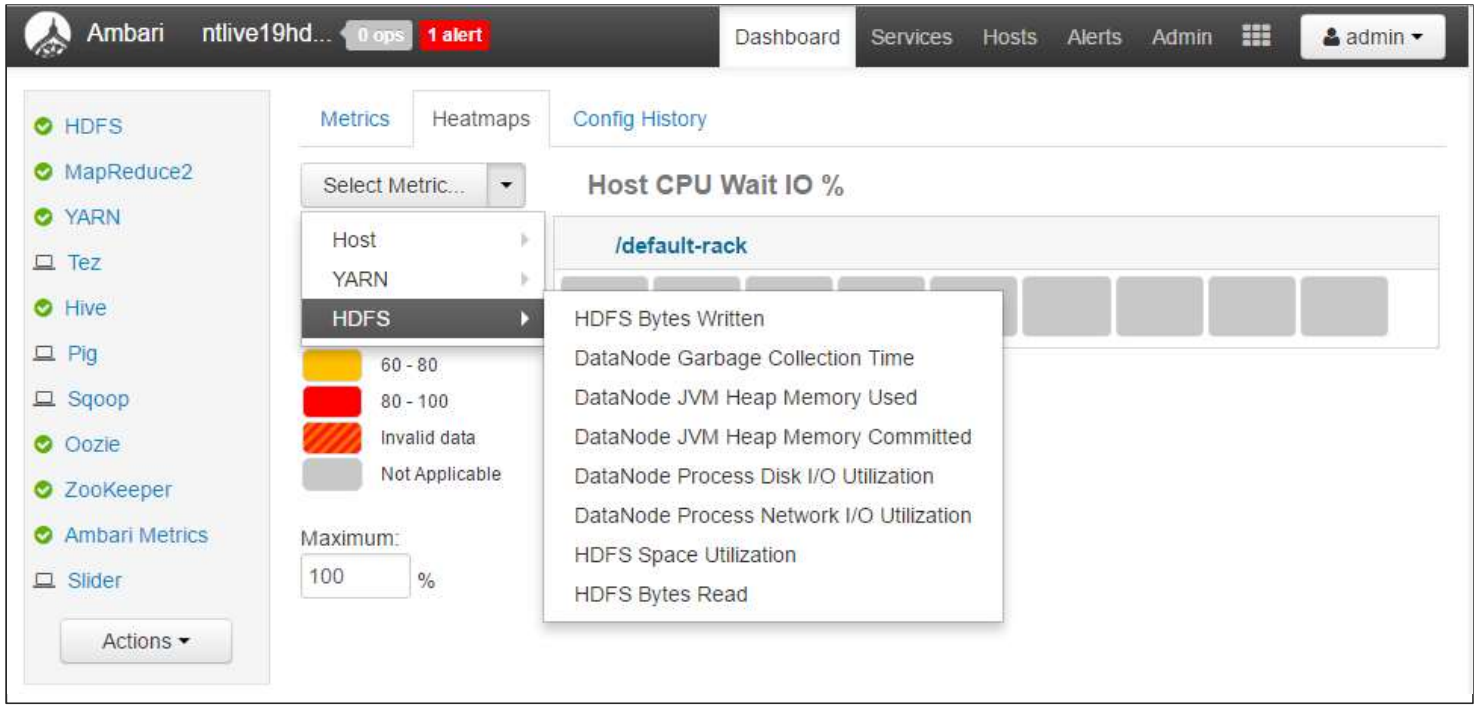
The Dashboard Heatmap view provides a color-coded view of each of the nodes in the cluster for selected metrics.

Hovering over each of the nodes pops ups additional information



Ambari Dashboard: HeatMap

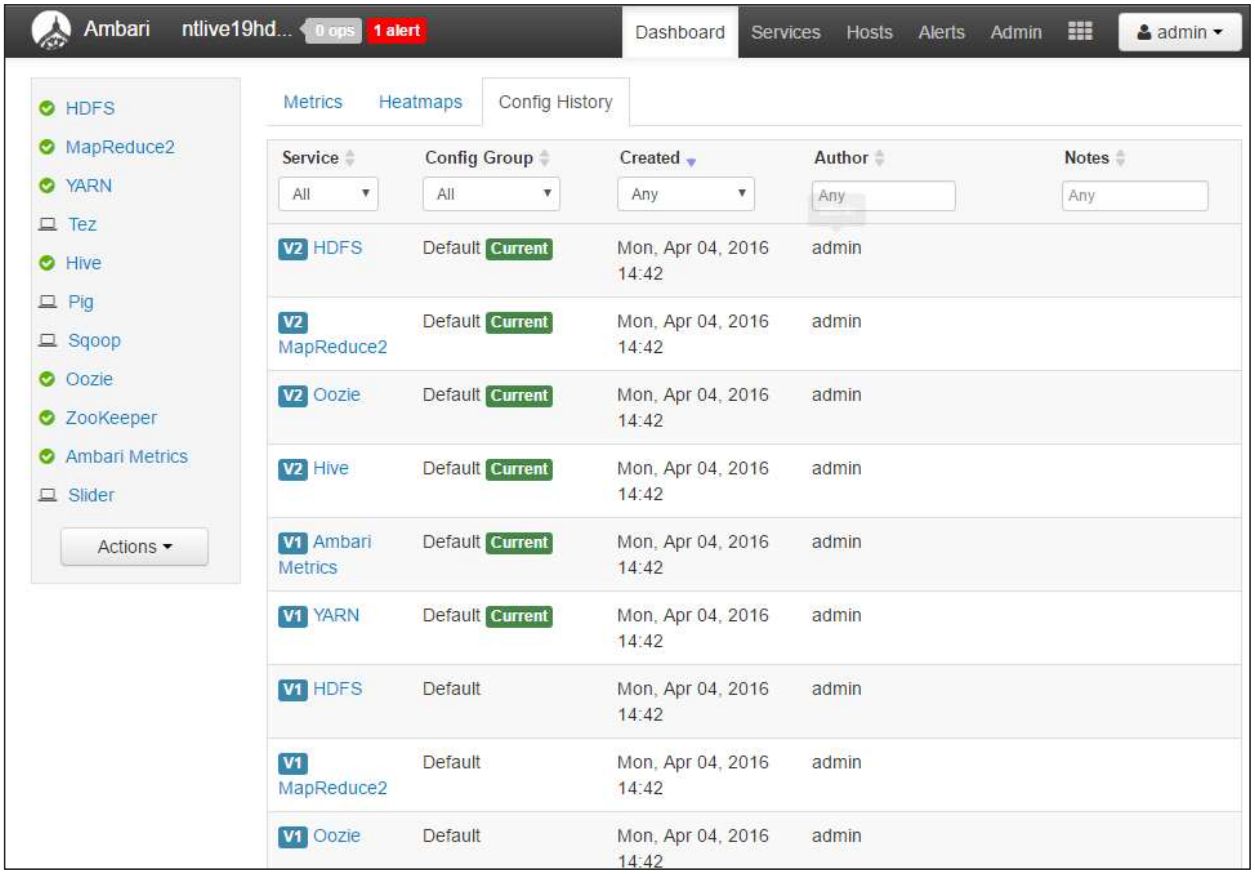
You can choose to show the Heatmap for ‘Host’, ‘Yarn’ and ‘HDFS’.
Each has an number of associated metrics for which the heatmap can be displayed.



Ambari Dashboard: 'Config History'

The Dashboard 'Config History' view displays the list of the configuration changes made, along with 'when' and 'who' details.

Additional config history details can be seen by drilling down into the specific services—this can also be seen from the 'Services' view or by clicking on the Services links on the left of the page



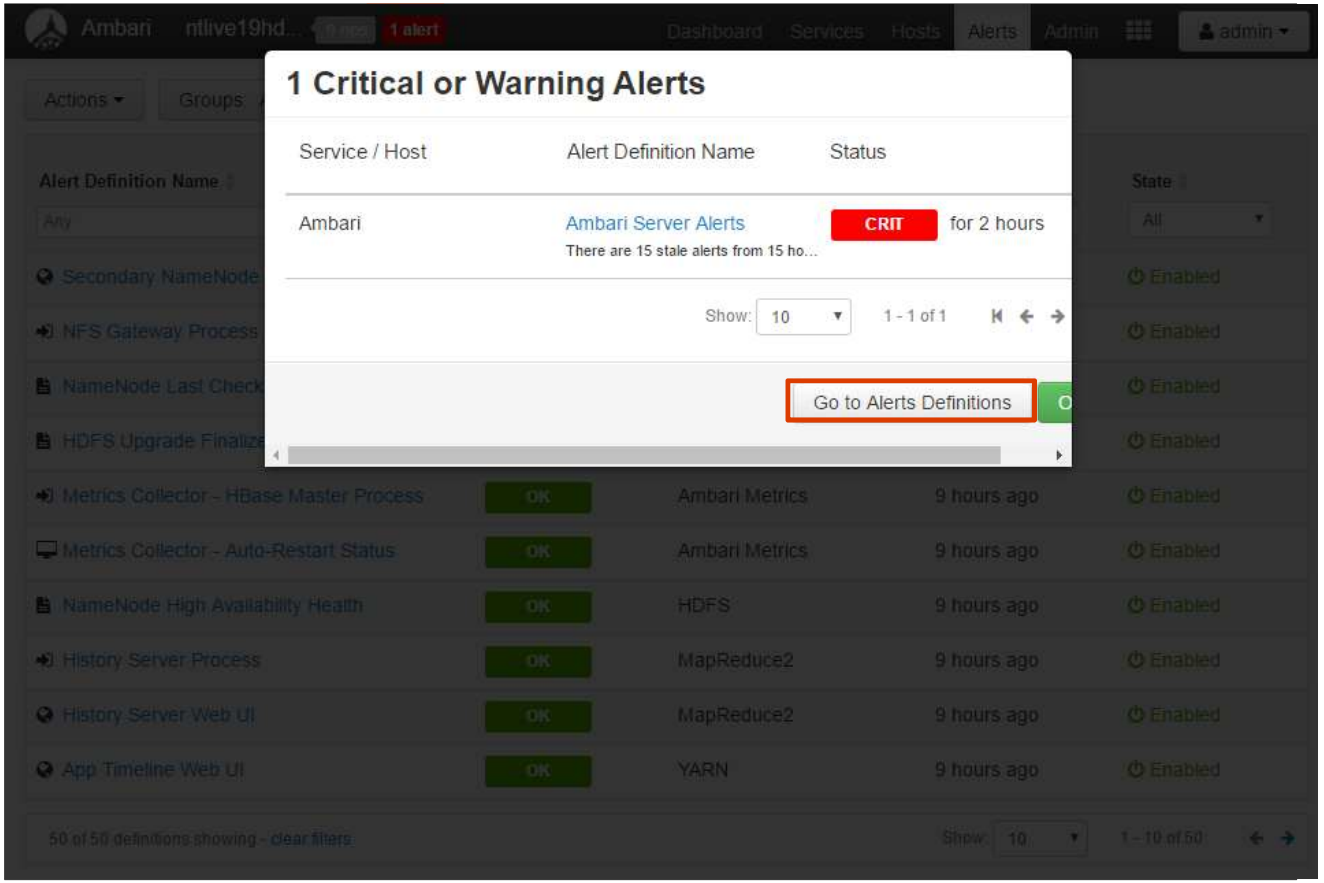
Ambari UI: Alerts

Ambari UI: Alerts

Ambari Web UI display any critical or Warning alerts at the top the page.

Clicking on the alert, pops up the list of alerts and current status

Clicking on 'Go to Alerts' definition display the complete list of alerts



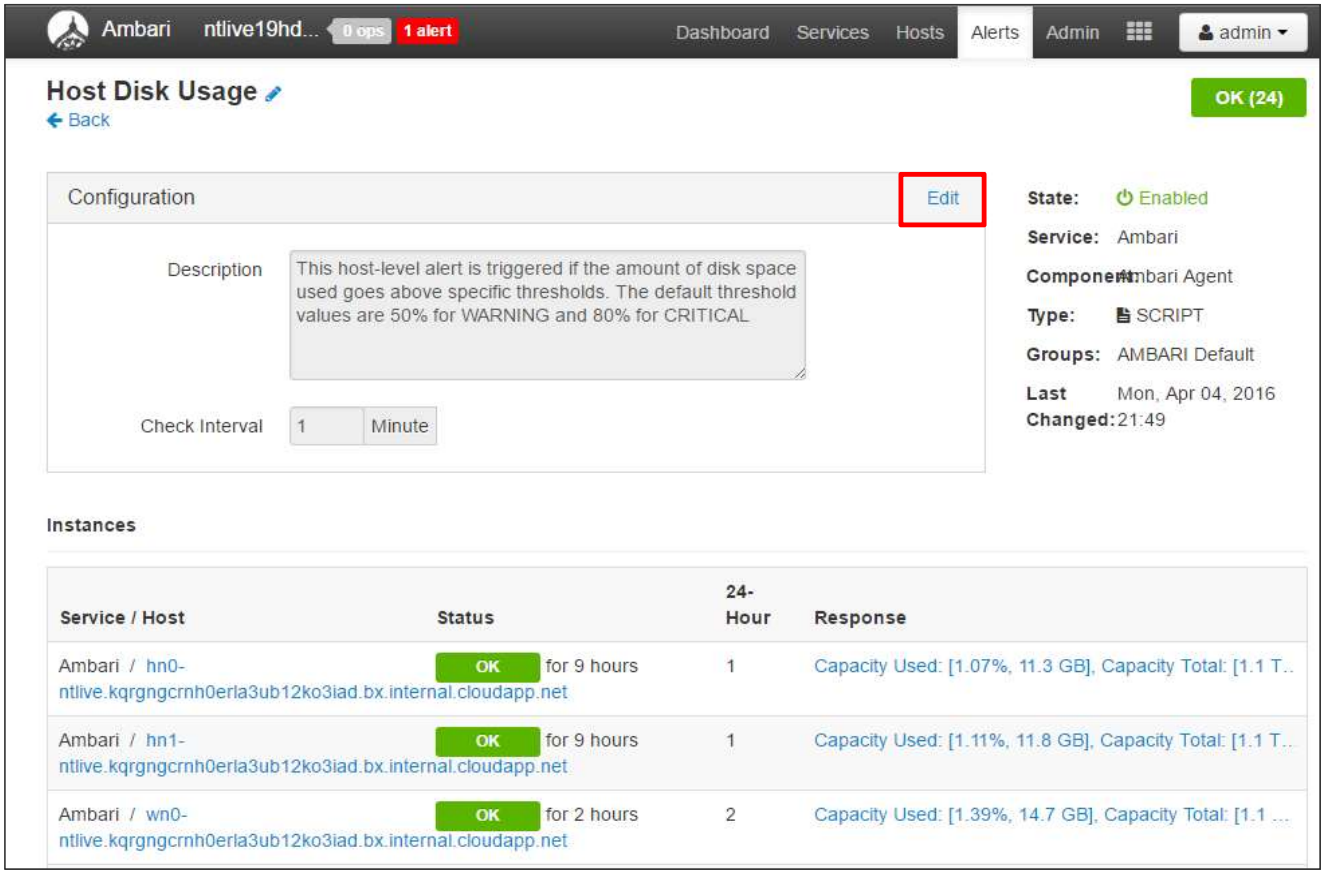
Ambari: List of Alerts

Ambari alerts are classified as:

- Critical
- OK
- Unknown
- None

You can drill down into specific alerts for details.

You can set the 'Check Interval' for the alerts by editing the alert.



Ambari UI: Services

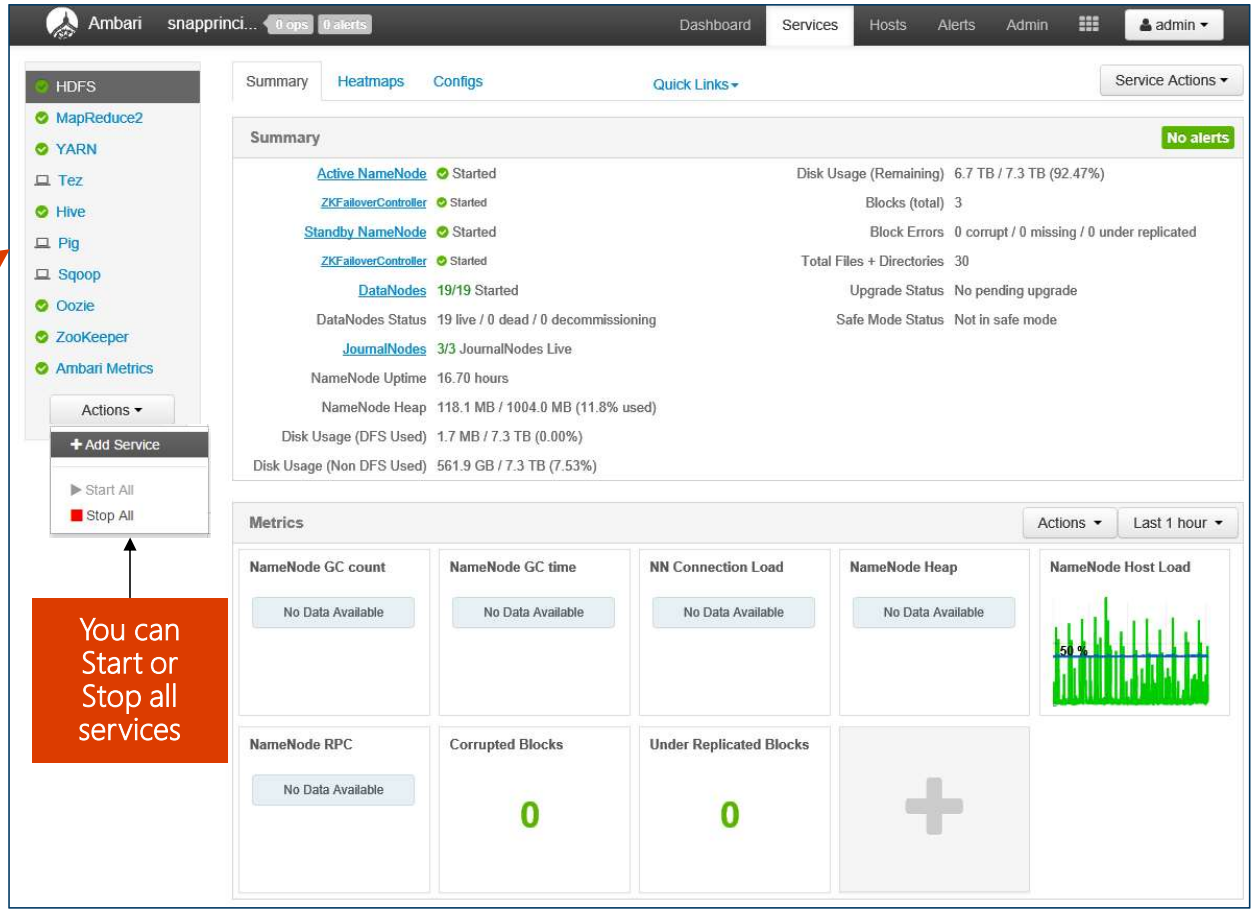
Ambari Web UI: Services

The **Services** page provides quick insights into the status of the services running on the cluster.

In this case the list of services running on the cluster include: **HDFS, MapReduce2, YARN, Hive, Oozie and Zookeeper**

Icons indicate status or actions that should be taken.

Shown here the details for HDFS for the last 1 hour.



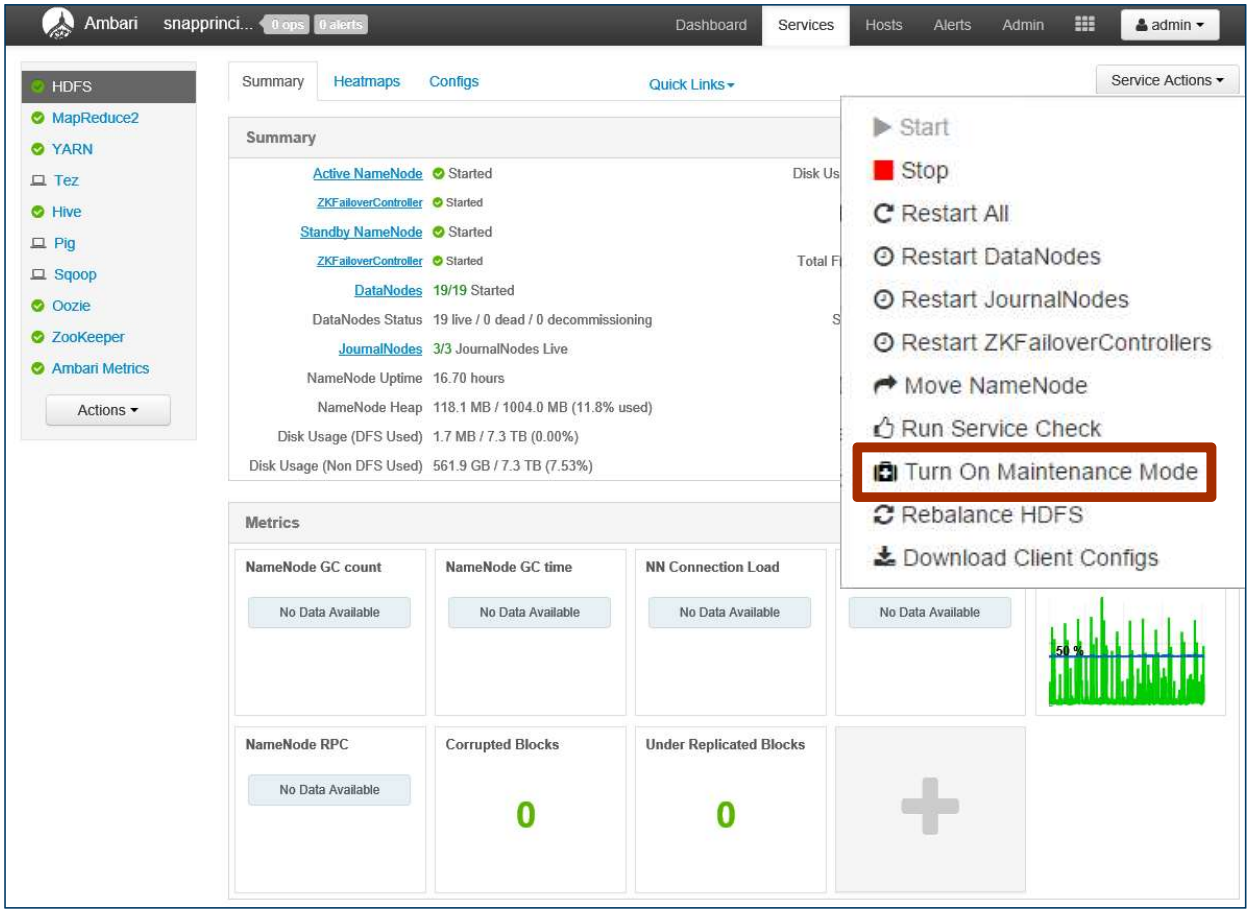
Ambari Web UI: Service Actions

For each service there are a list of associated "**Service Actions**" to manage, monitor and configure the service.

As the Service Actions menu button is context-sensitive, the menu choices are different for each service.

The Service Actions for HDFS are shown here.

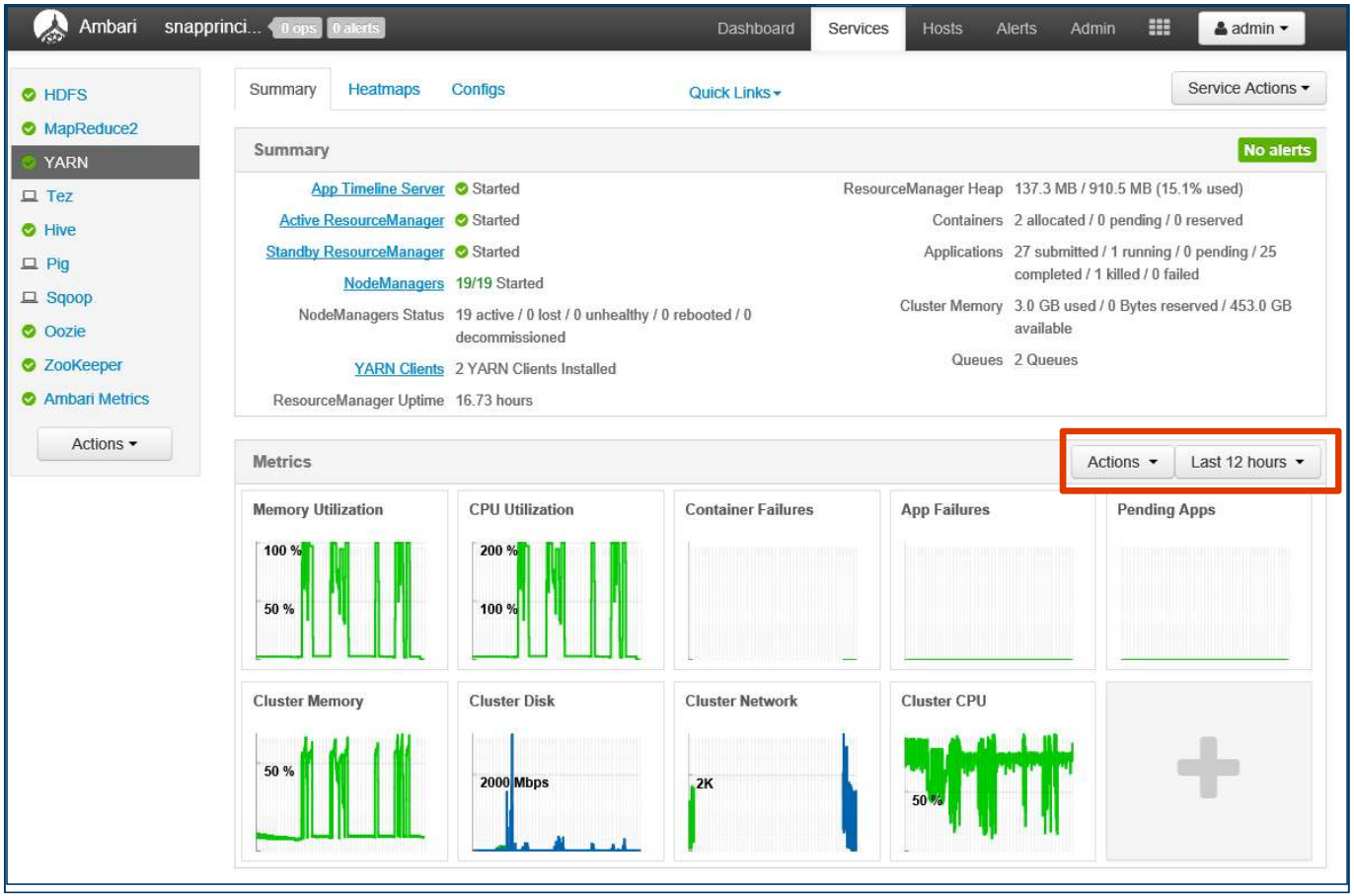
Maintenance Mode should be enabled when making cluster hardware or software changes. It suppresses Ambari alerts, warnings and status change indicators



Ambari Web UI: Services (YARN)

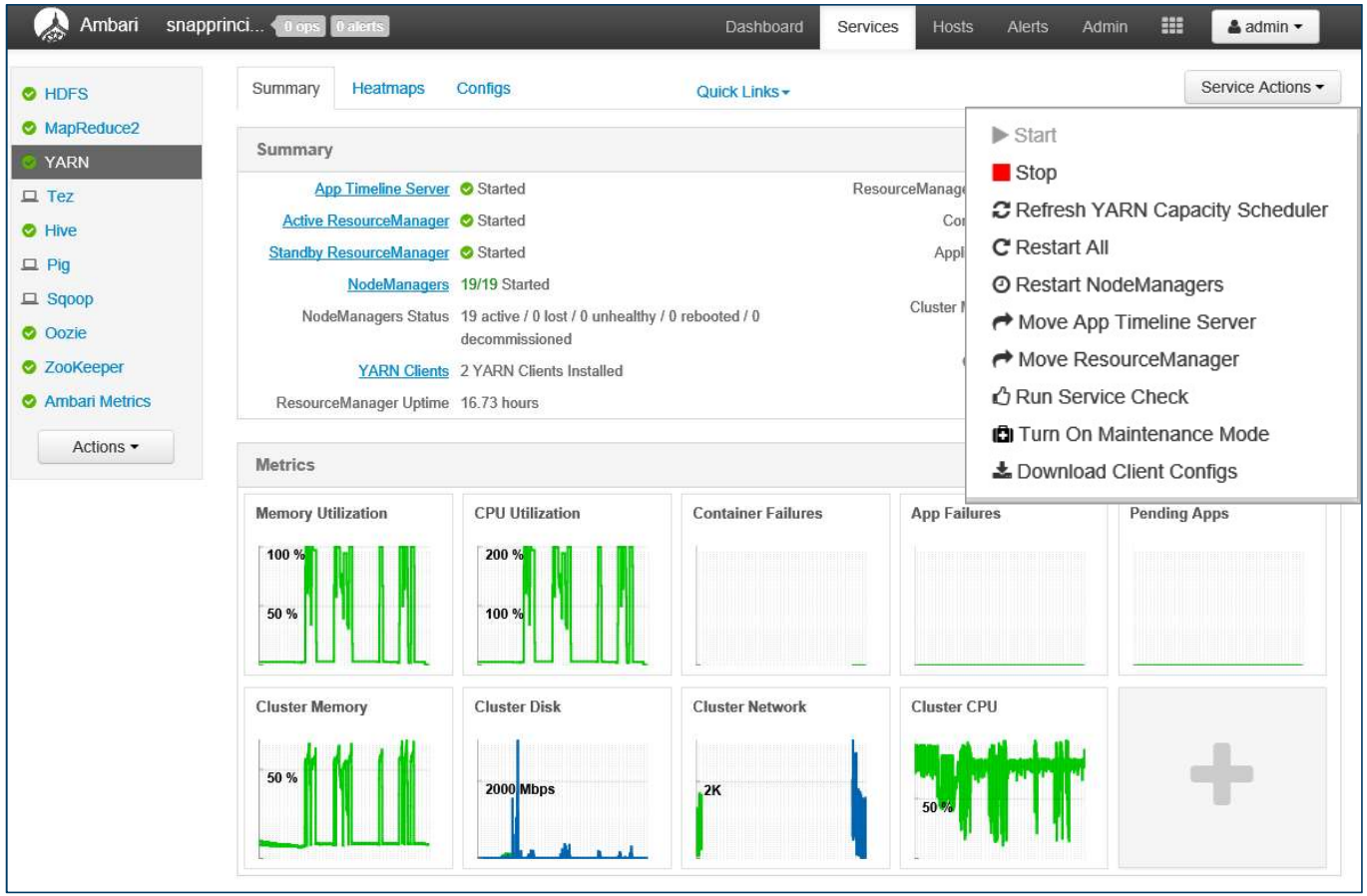
The **Services** page details for YARN for the last 1 hour

… and the last 12 hours



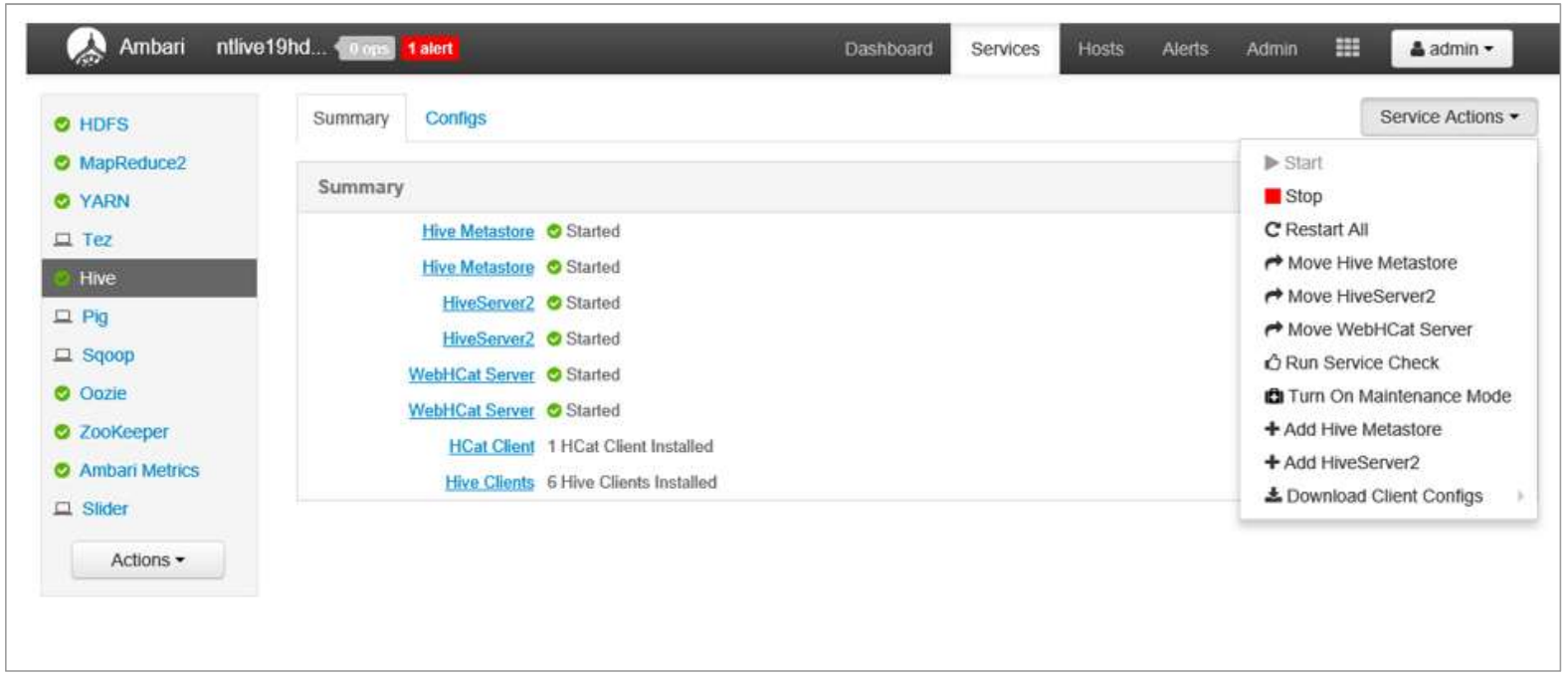
Ambari: YARN "Service Actions"

Here are the list of actions that can be taken with YARN.



Ambari Web UI: Hive Service Actions

This is the Hive Services page with the list of associated actions.

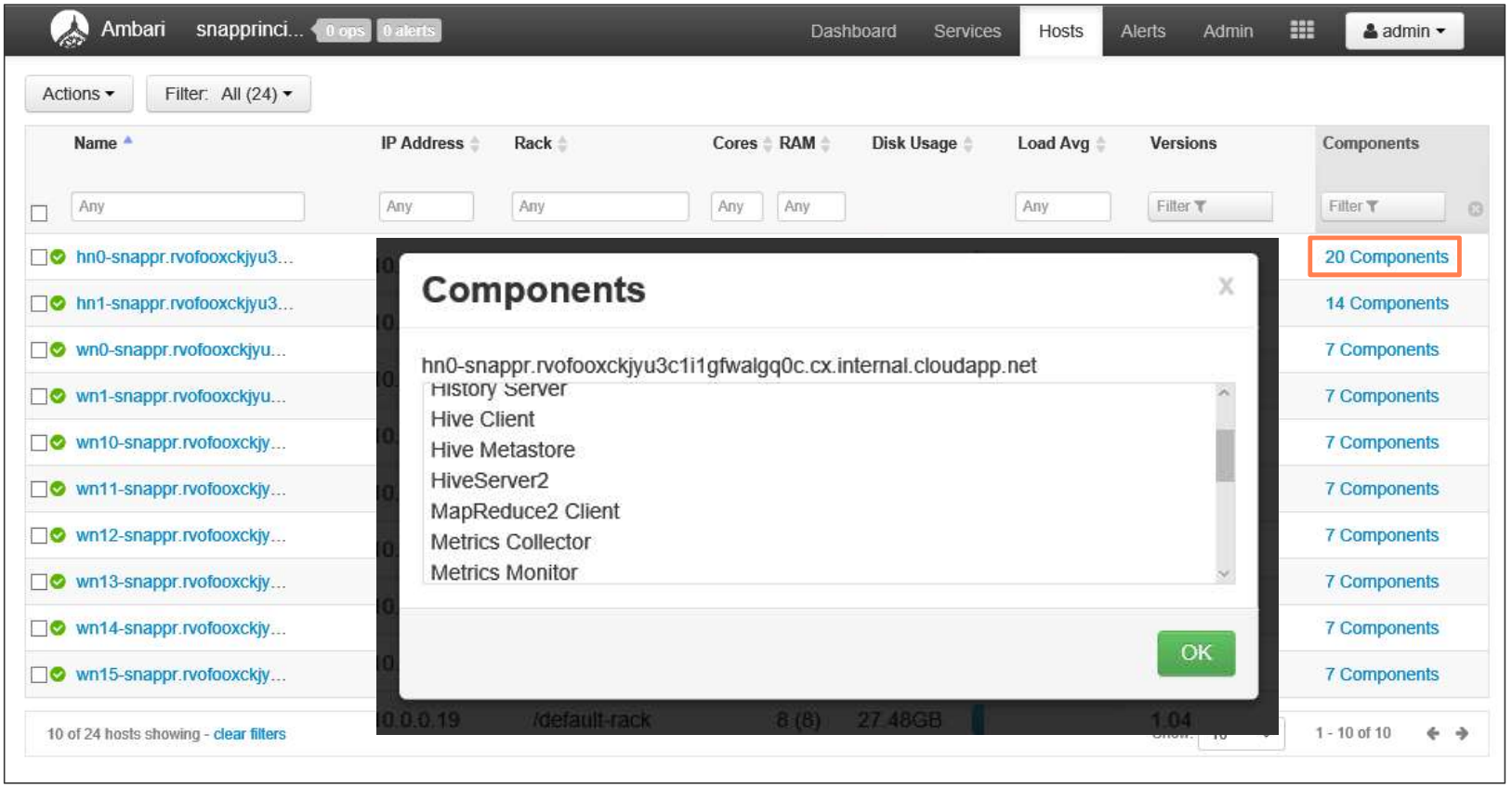


Ambari Web UI: Hosts

Ambari Web UI: Hosts

The Hosts page provides system-level metrics for each node in the cluster including.

Clicking on the components link, provides more details on the list of components running on the node.



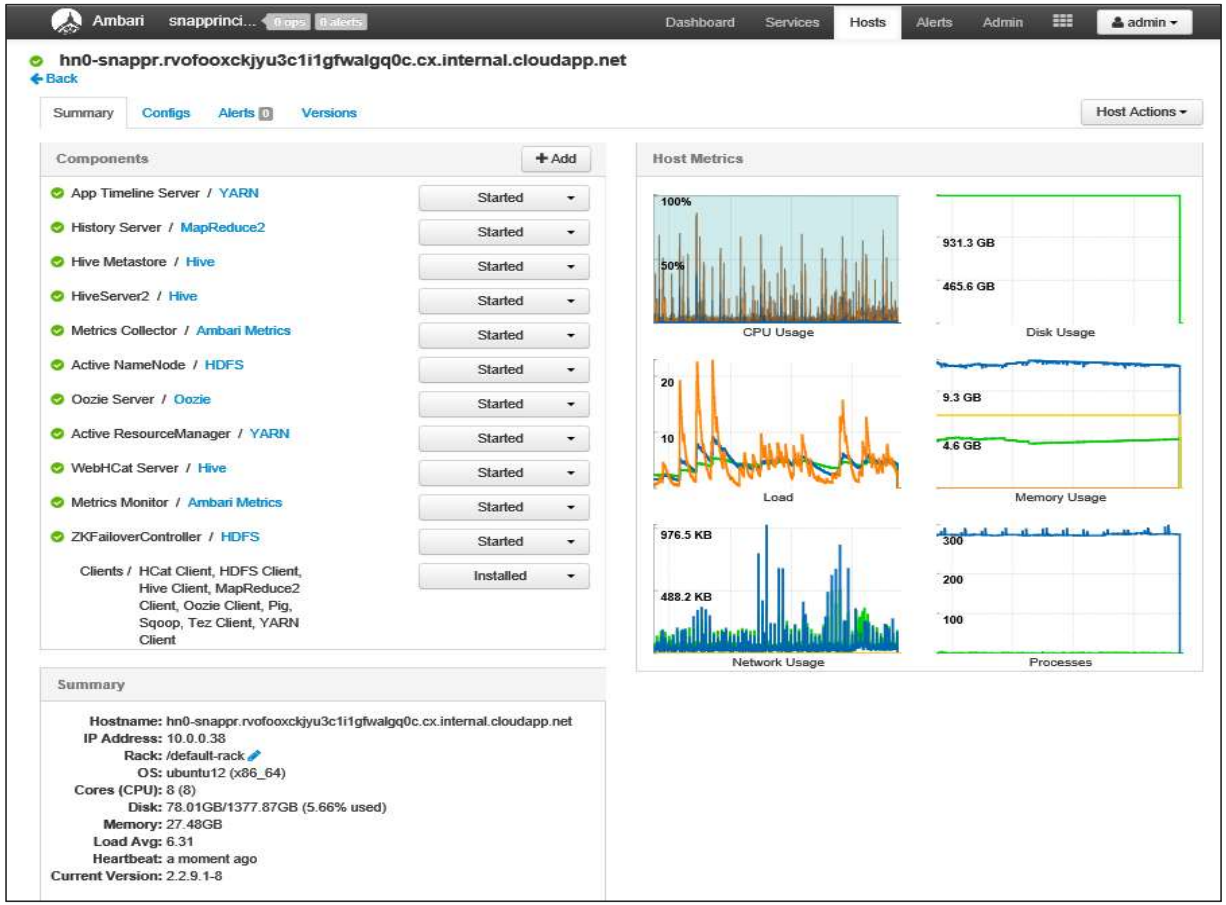
Amber Web UI: Hosts Drilldown

You can drilldown into the details of any of the nodes in the cluster.

At a glance you can see the charts for CPU, Memory and Network usage.

The summary system-configuration information is also displayed.

You can see—and change—the status of each of the components running on the node.



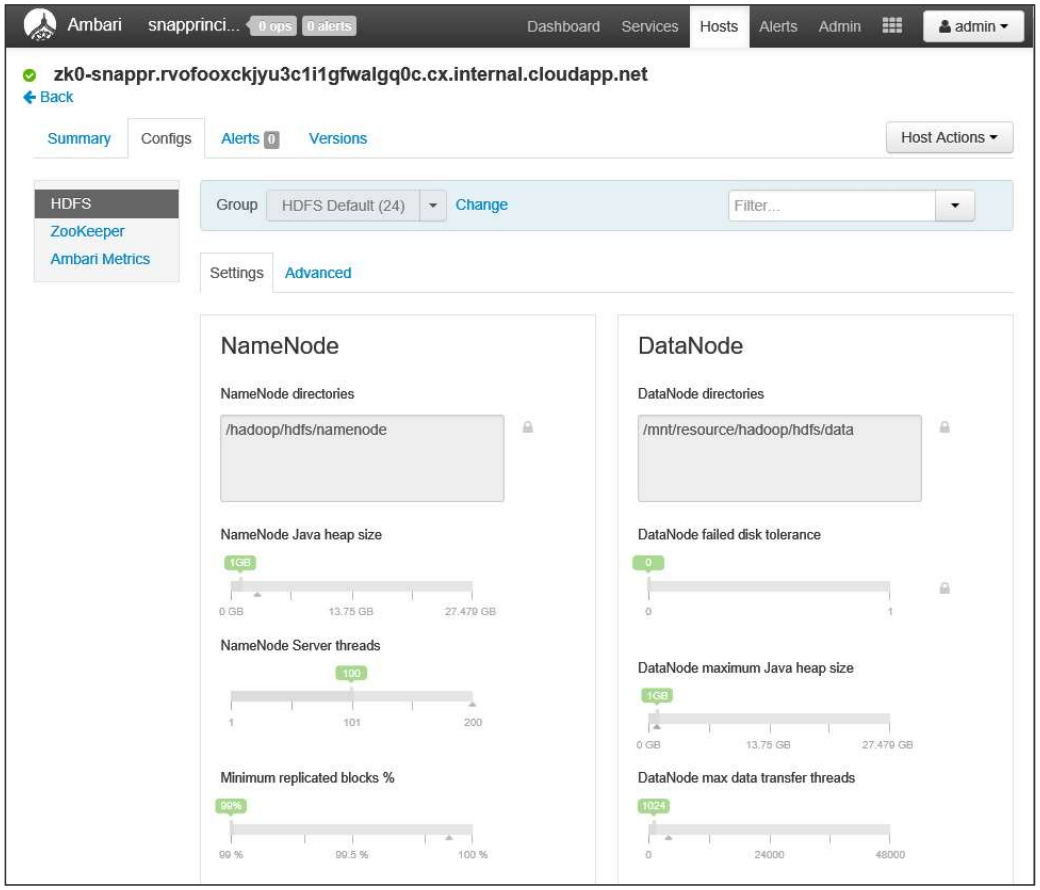
Ambari Web UI: Host Component Drilldown

On the hosts page you can drill down into the details about any of the components running on the node.

This shows key metrics about the ***HDFS component*** running on the this node.

You can ***configure HDFS parameters*** such as:

- NameNode Java heap size
- NameNode Server threads
- Minimum replicated blocks %
- DataNode failed disk tolerance
- DataNode max Java heap size
- DataNode max data transfer threads



Ambari Web UI: User Views

Ambari: Capacity Scheduler View

The YARN Capacity Scheduler allows Hadoop to be shared among multiple independent tenants while providing guaranteed capacity and predictable SLAs.

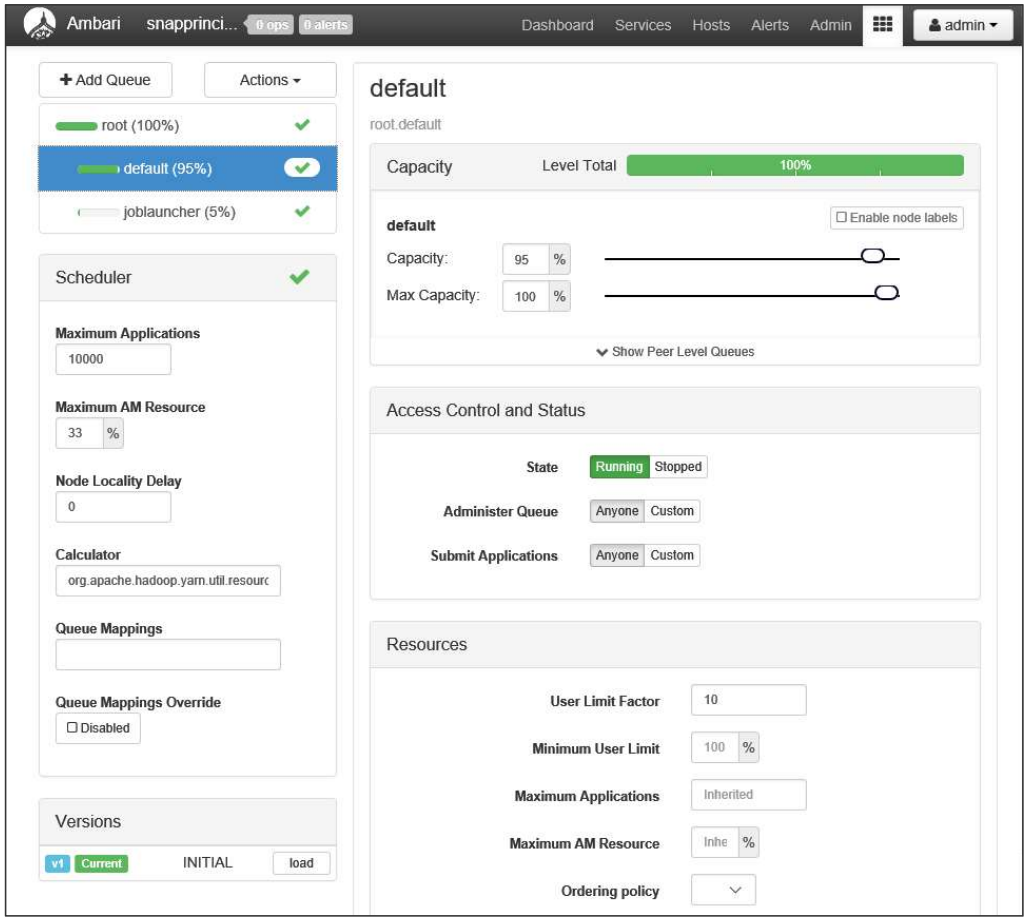
The Capacity Scheduler divides resources through use of **YARN queues**, which are sized based on the relative allocations given to various tenants.

The ***Capacity Scheduler View*** lets you create and modify ***YARN queues*** and see their distribution at-a-glance.

The UI enforces configuration rules, highlights invalid conditions.

With the Capacity Scheduler View you can:

- Partition Hadoop resources among tenants.
- Define, view and modify queue definitions.
- Establish fine-grained control on who can run jobs in queues.

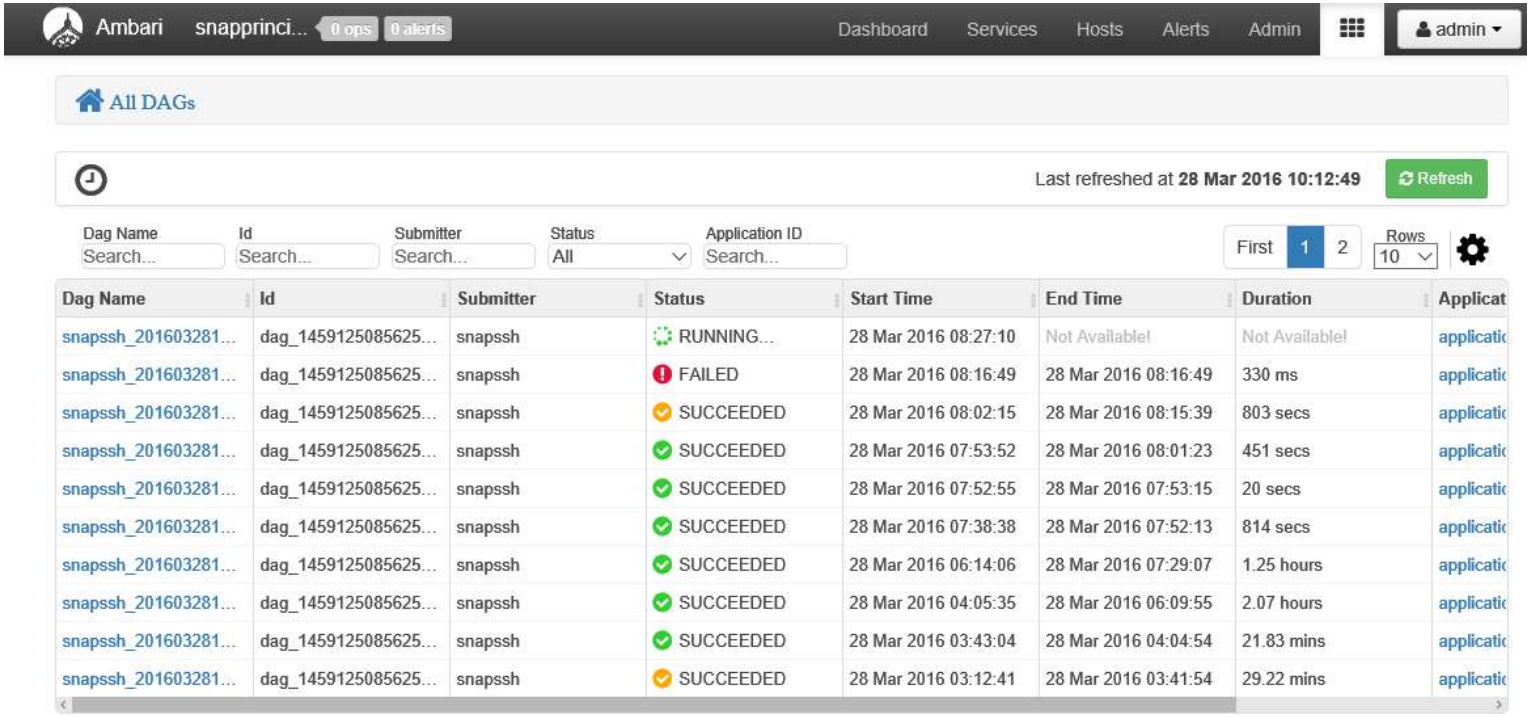


Microsoft

Ambari: Tez View

The Tez View lists all the DAGs (currently executing and historical) over a time period.

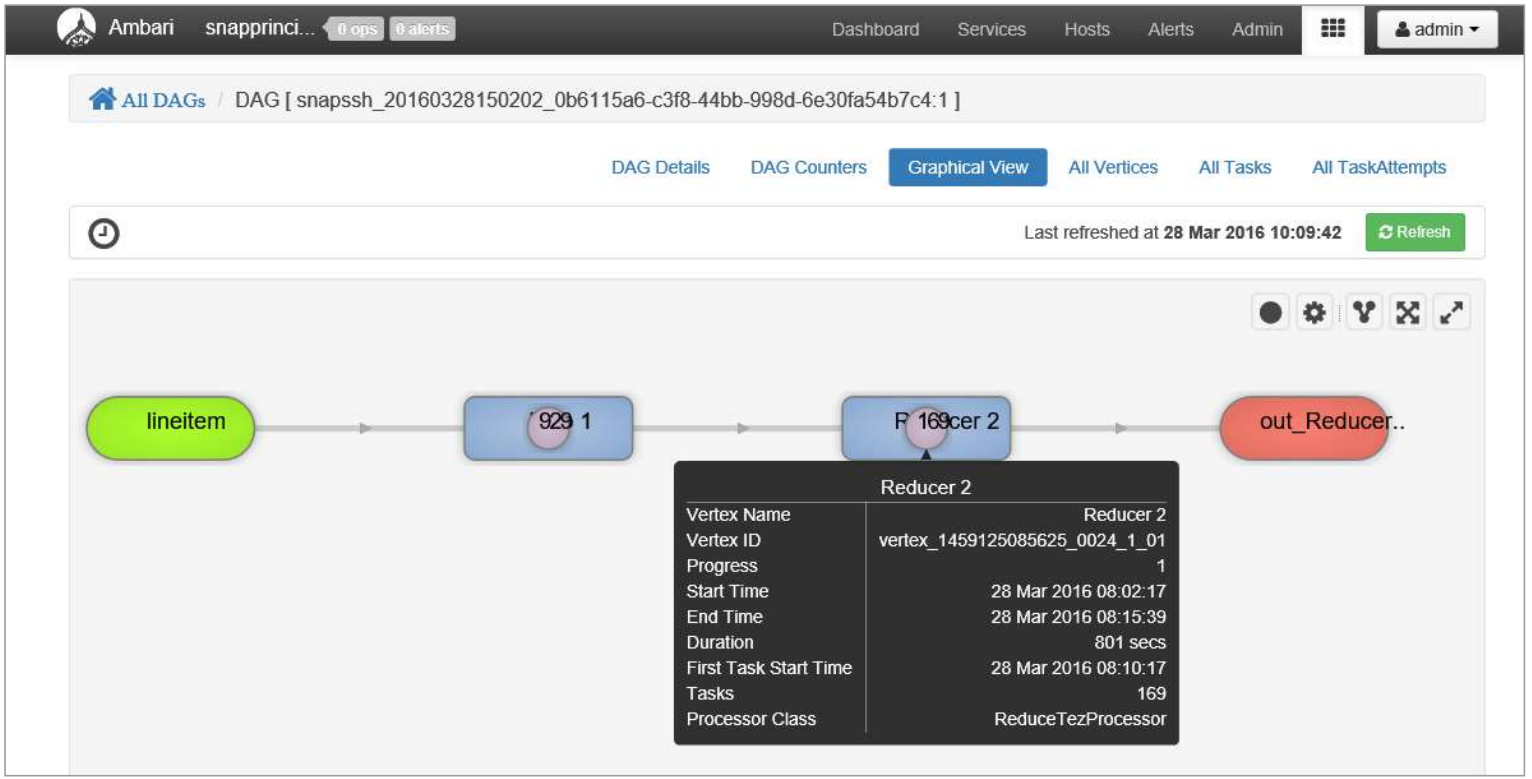
You can drill down into specific DAG to see more details



Dag Name	Id	Submitter	Status	Start Time	End Time	Duration	Applicat
snapssh_201603281...	dag_1459125085625...	snapssh	RUNNING...	28 Mar 2016 08:27:10	Not Available!	Not Available!	applicatic
snapssh_201603281...	dag_1459125085625...	snapssh	FAILED	28 Mar 2016 08:16:49	28 Mar 2016 08:16:49	330 ms	applicatic
snapssh_201603281...	dag_1459125085625...	snapssh	SUCCEEDED	28 Mar 2016 08:02:15	28 Mar 2016 08:15:39	803 secs	applicatic
snapssh_201603281...	dag_1459125085625...	snapssh	SUCCEEDED	28 Mar 2016 07:53:52	28 Mar 2016 08:01:23	451 secs	applicatic
snapssh_201603281...	dag_1459125085625...	snapssh	SUCCEEDED	28 Mar 2016 07:52:55	28 Mar 2016 07:53:15	20 secs	applicatic
snapssh_201603281...	dag_1459125085625...	snapssh	SUCCEEDED	28 Mar 2016 07:38:38	28 Mar 2016 07:52:13	814 secs	applicatic
snapssh_201603281...	dag_1459125085625...	snapssh	SUCCEEDED	28 Mar 2016 06:14:06	28 Mar 2016 07:29:07	1.25 hours	applicatic
snapssh_201603281...	dag_1459125085625...	snapssh	SUCCEEDED	28 Mar 2016 04:05:35	28 Mar 2016 06:09:55	2.07 hours	applicatic
snapssh_201603281...	dag_1459125085625...	snapssh	SUCCEEDED	28 Mar 2016 03:43:04	28 Mar 2016 04:04:54	21.83 mins	applicatic
snapssh_201603281...	dag_1459125085625...	snapssh	SUCCEEDED	28 Mar 2016 03:12:41	28 Mar 2016 03:41:54	29.22 mins	applicatic

Ambari: Tez View (DAG Details)

The Graphical View lets you visualize the DAG execution flow graphically. You can get move details about any vertex by clicking on it.



Microsoft

Get started today!

- For more information visit: http://azure.microsoft.com/en-us/services/hdinsight/