



HDInsight Administration and Security

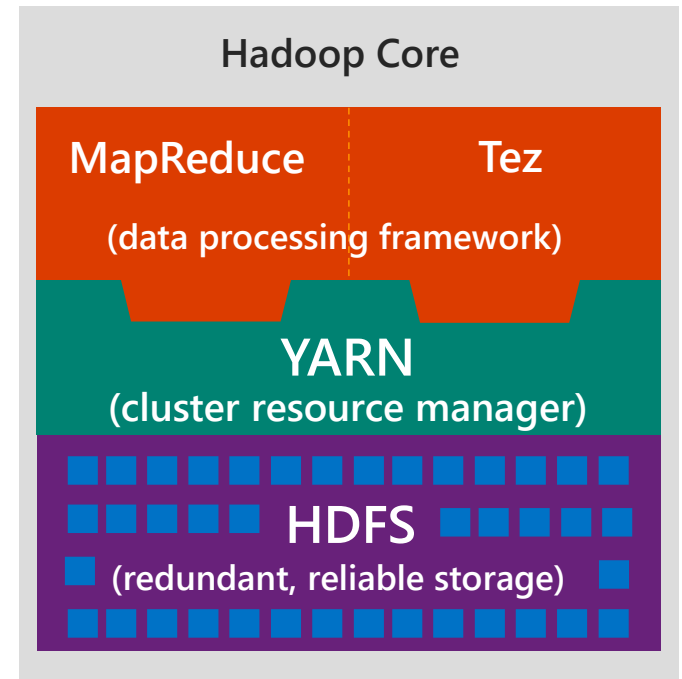
Technical Overview

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?

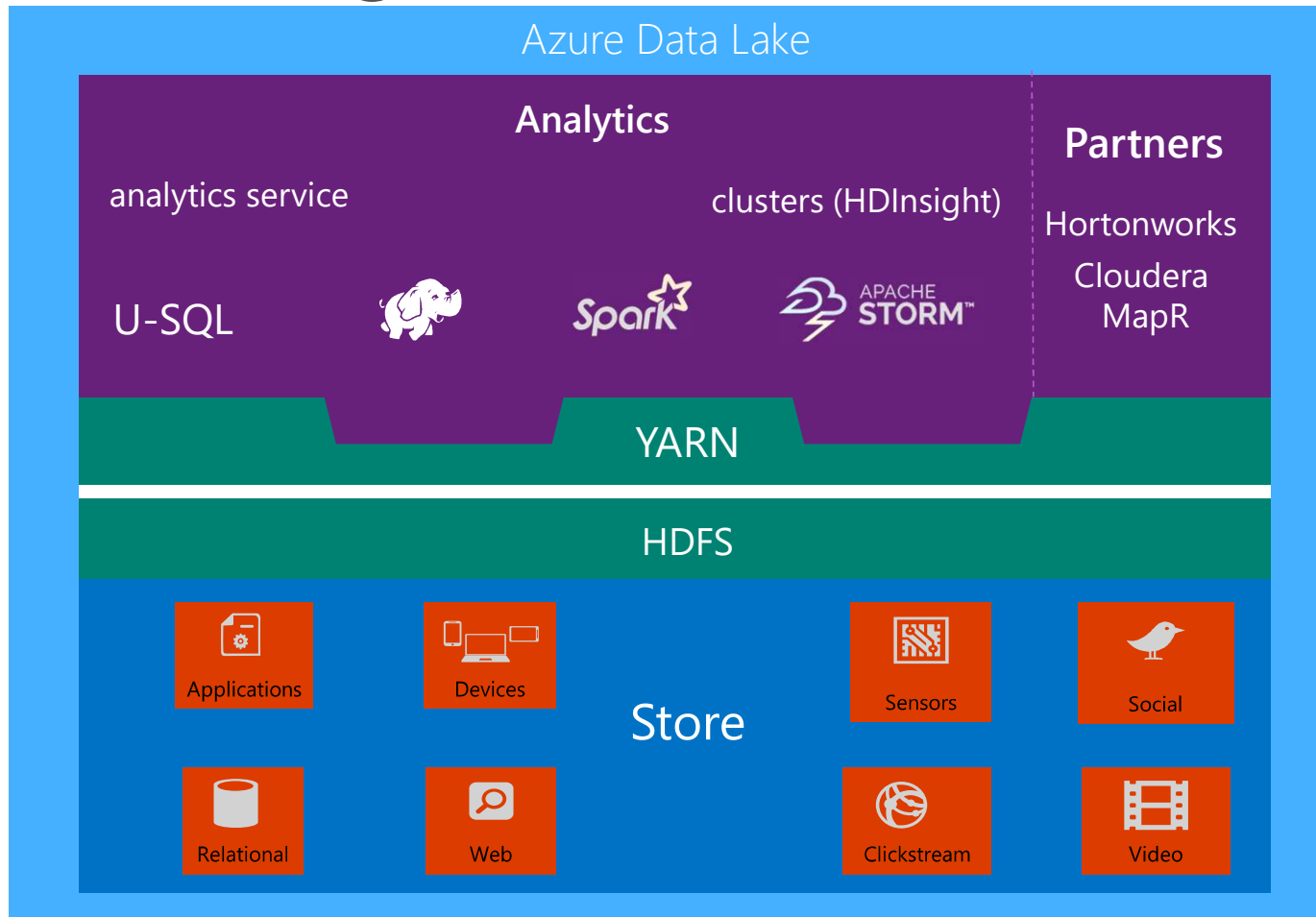
A standard Apache Hadoop distribution offered as a managed service on Microsoft Azure

- ❖ 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

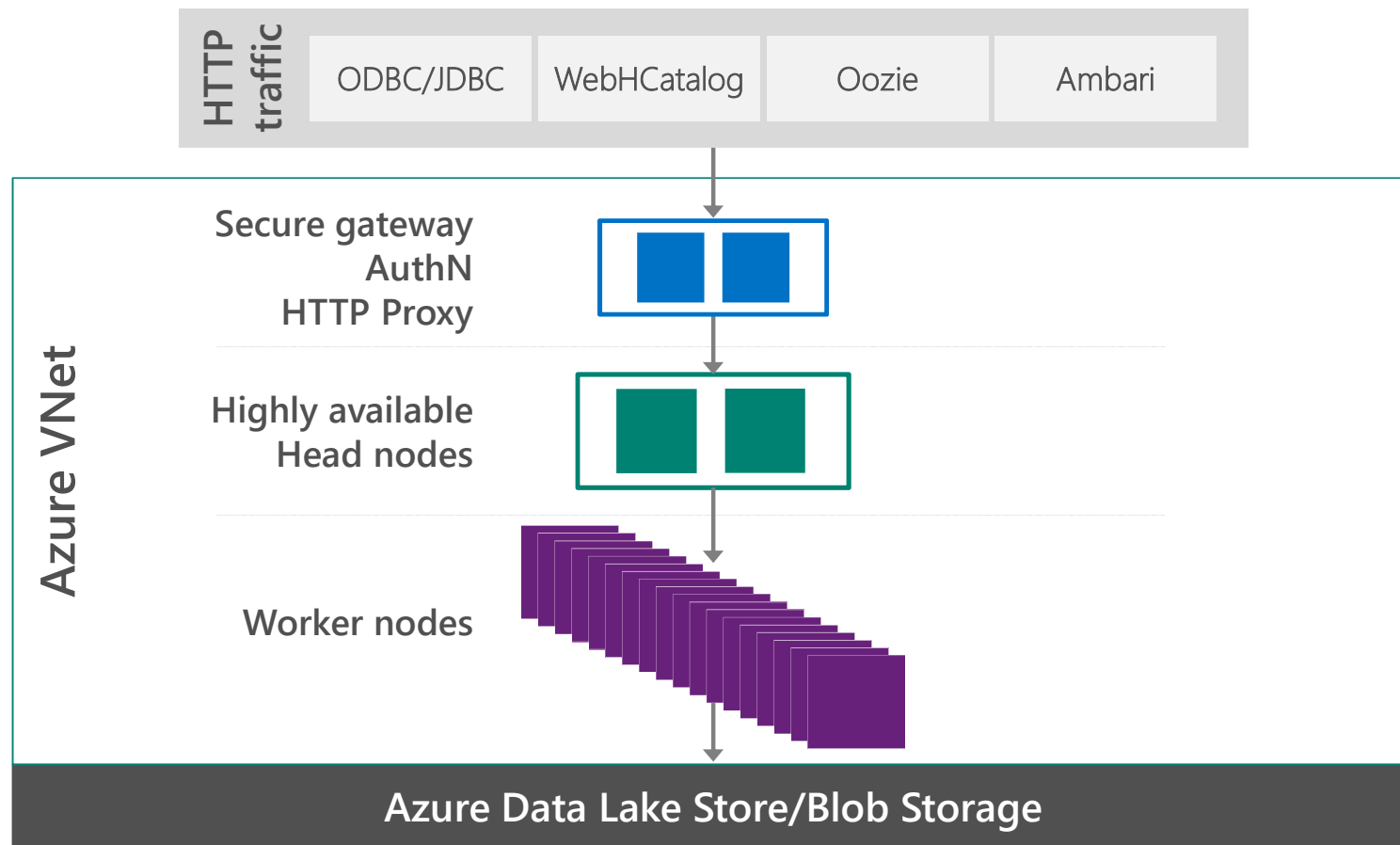


HDInsight: How/Where it fits?

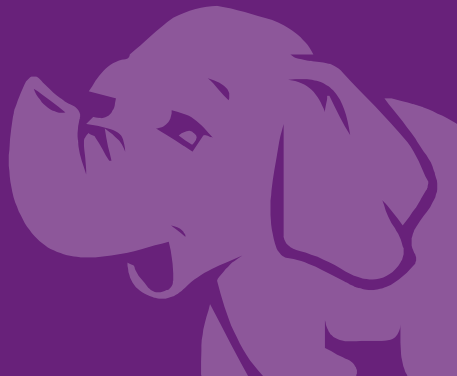


- ✓ Integrated analytics and storage
- ✓ Fully Managed
- ✓ Easy to use – “dial for scale”
- ✓ Proven at scale
- ✓ Analyze data of any size, shape or speed
- ✓ Open-standards based






HDInsight cluster architecture



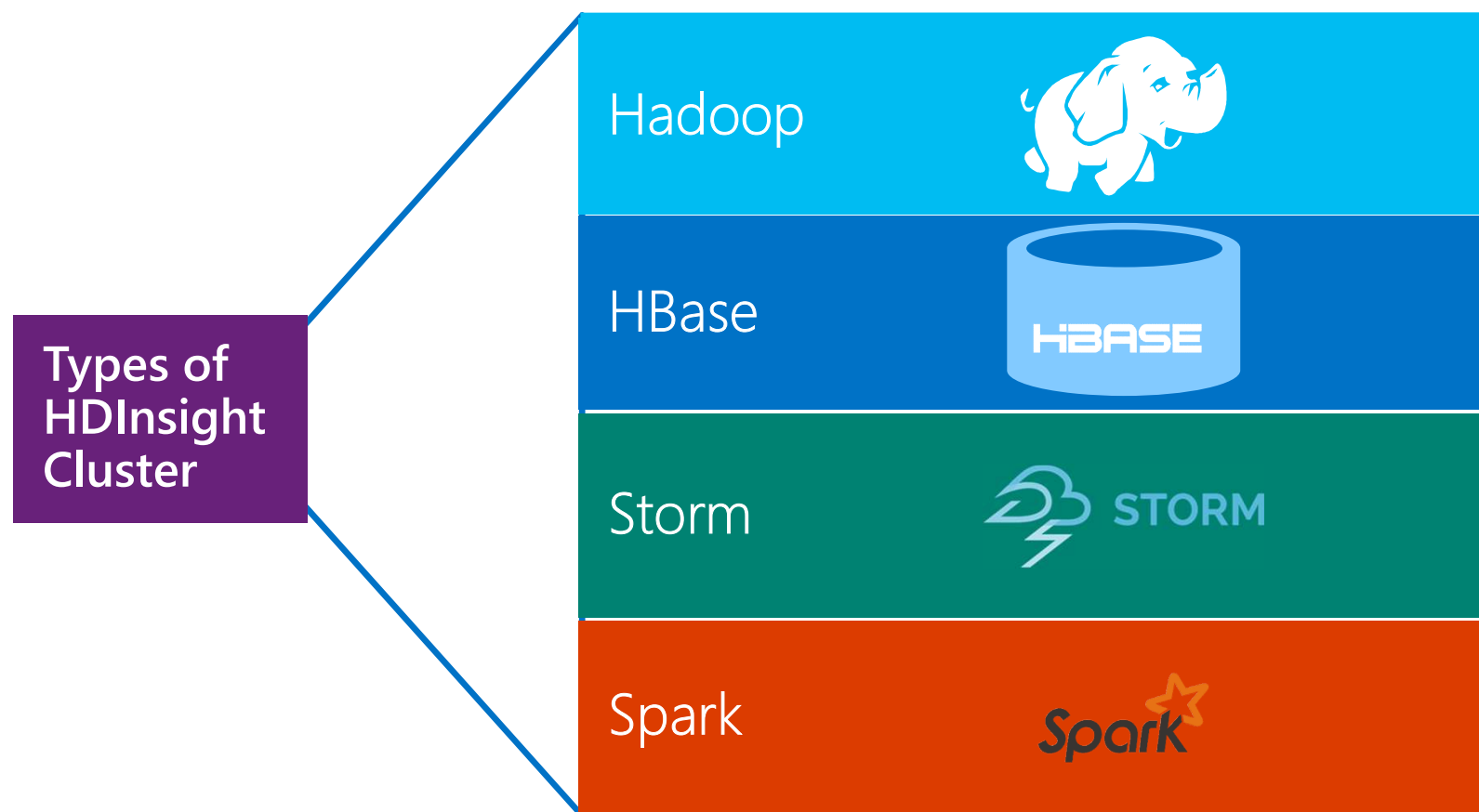
HDInsight on Linux: Administration Overview



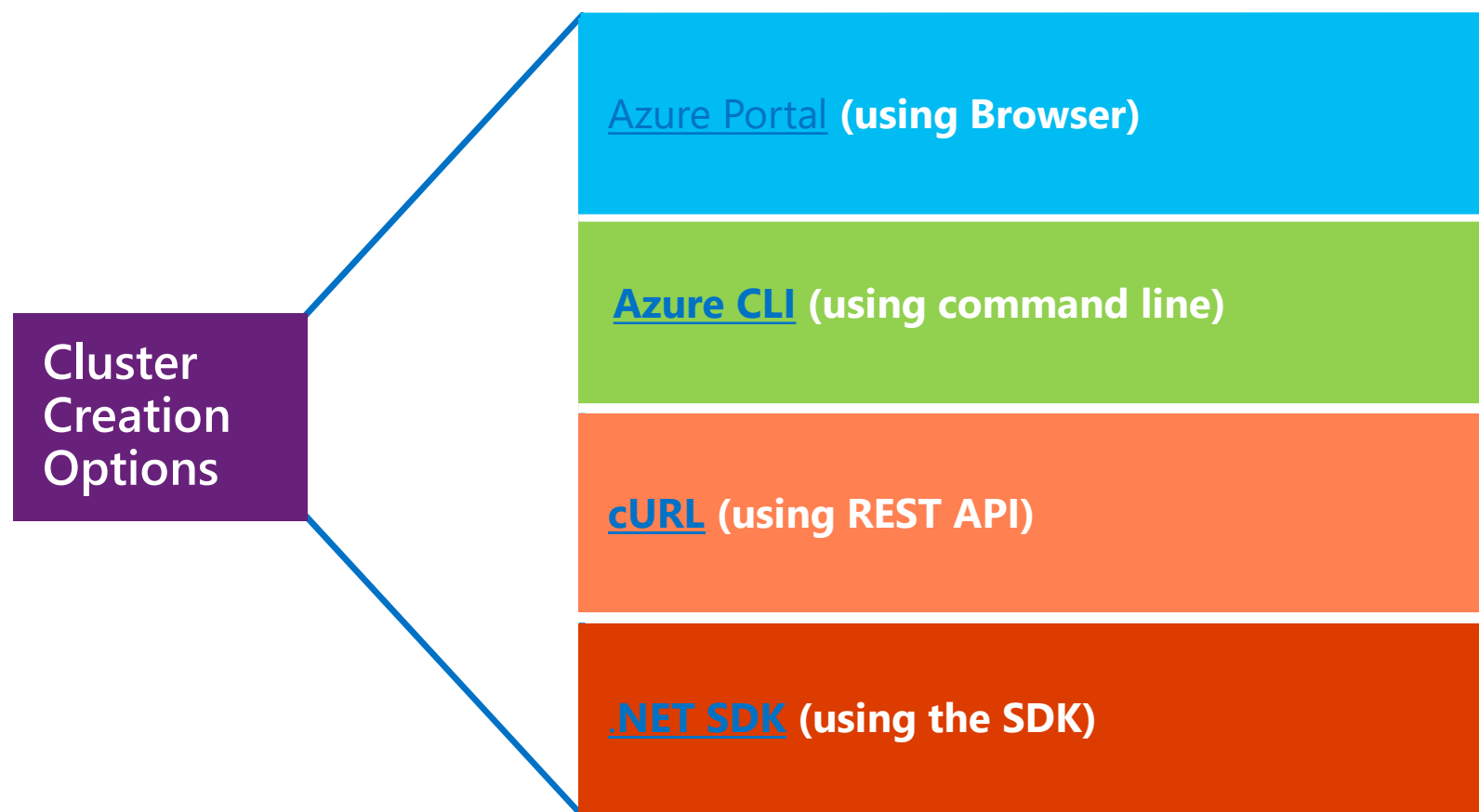
Agenda

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

Cluster Types Overview



Ways to create HDInsight (Linux) Clusters



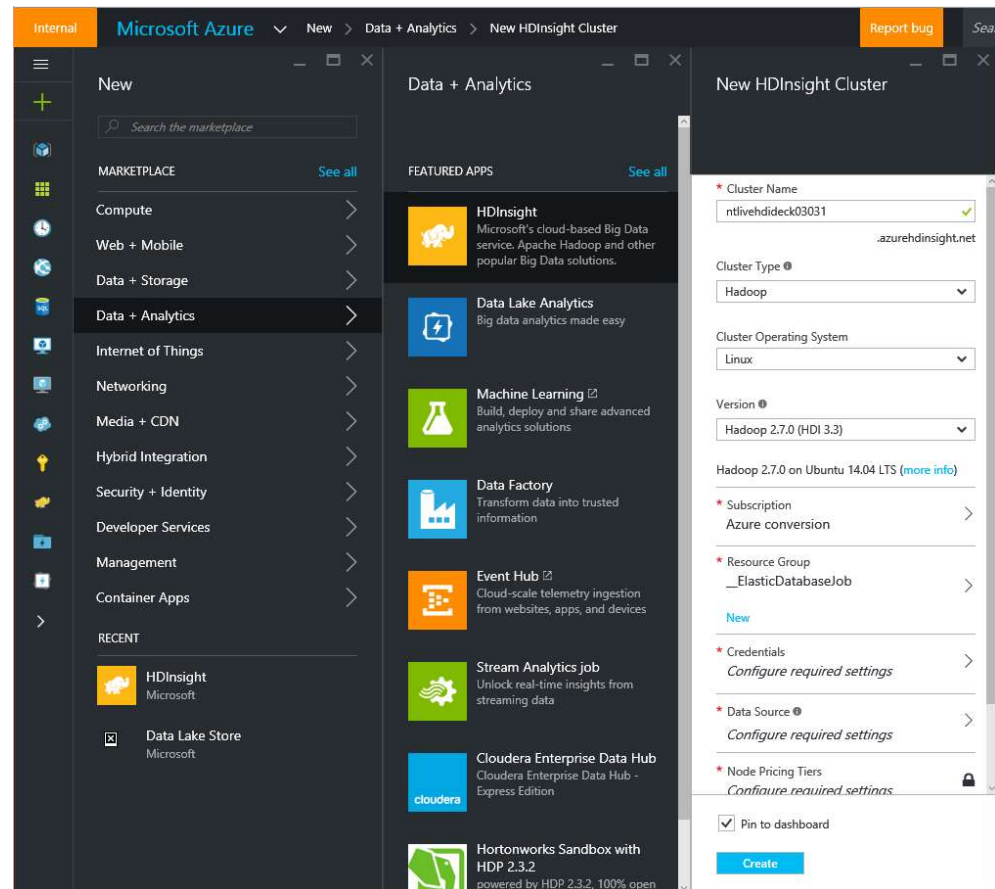
Creating a HDInsight Cluster via the Portal

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

The screenshot shows the 'New HDInsight Cluster' configuration page in the Microsoft Azure portal. The breadcrumb navigation at the top reads: 'Microsoft Azure > New > Data + Analytics > New HDInsight Cluster > Cluster Type configuration'. The left sidebar contains navigation links: 'New', 'Resource groups', 'All resources', 'Recent', 'App Services', 'Virtual machines (classic)', 'Virtual machines', 'SQL databases', 'Cloud services (classic)', and 'Subscriptions'. The main configuration area is divided into two panels. The left panel, titled 'New HDInsight Cluster', contains fields for 'Cluster Name' (samplehdi), 'Subscription' (Pay-As-You-Go), 'Credentials', 'Data Source', 'Node Pricing Tiers', and 'Resource Group' (ADLA_Benchmark). The right panel, titled 'Cluster Type configuration', shows 'Cluster Type' set to 'Hadoop' and 'Operating System' set to 'Linux'. A tooltip is displayed over the 'Hadoop' cluster type, listing its components and other available cluster types: HBase, Storm, and Spark. The tooltip text is: 'Hadoop: Terabyte-scale processing with Hadoop components like Hive (SQL in Hadoop), Pig, and Oozie. HBase: Fast and scalable NoSQL database. Storm: Reliably process infinite streams of data in real-time. Spark: Fast data analytics and cluster computing using in-memory processing.' The pricing section at the bottom shows a cost of '+ 0.00 USD/CORE/HOUR' for the selected configuration and '+ 0.02 USD/CORE/HOUR' for the HDInsight service.

Step 2 :Specify Version and Cluster Tier

Microsoft Azure

New > Data + Analytics > New HDInsight Cluster > Cluster Type configuration

New HDInsight Cluster

Cluster Type configuration

Learn about HDInsight and cluster versions. [Learn more](#)

* Cluster Name
samplehdi ✓
.azurehdinsight.net

* Subscription
Pay-As-You-Go >

Select Cluster Type ⓘ
Configure required settings ⓘ

* Credentials
Configure required settings

* Data Source ⓘ
Configure required settings

* Node Pricing Tiers
Please configure required settings

Optional Configuration

* Resource Group
ADLA_Benchmark >

New

Cluster Type ⓘ
Hadoop

Operating System
Linux Windows

Version
Hadoop 2.7.1 (HDI 3.4)

Cluster Tier (more info)

STANDARD	PREMIUM
Administration Manage, monitor, connect	Administration Manage, monitor, connect
Scalability On-demand node scaling	Scalability On-demand node scaling
99.9% Uptime SLA	99.9% Uptime SLA
Automatic patching	Automatic patching
Microsoft R Server for HDInsight	
+ 0.00 USD/CORE/HOUR	+ 0.02 USD/CORE/HOUR

Premium tier is available only with:

- Version 3.4
- For Hadoop and Spark

Other supported versions are 3.3 and 3.2

Step 3: Specify SSH and Admin Credentials

Microsoft Azure

New > Data + Analytics > New HDInsight Cluster > Cluster Credentials

New

Resource groups

All resources

Recent

App Services

Virtual machines (classic)

Virtual machines

SQL databases

Cloud services (classic)

Subscriptions

Browse >

New HDInsight Cluster

Cluster Name
samplehdi ✓
azurehdinsight.net

Subscription
Pay-As-You-Go >

Select Cluster Type ⓘ
Standard Hadoop on Linux (3.4) >

Credentials
Configure required settings >

Data Source ⓘ
Configure required settings >

Node Pricing Tiers
Please configure required settings 🔒

Optional Configuration 🔒

Cluster Credentials

Create login and remote access credentials for the cluster.

Cluster Login Username ⓘ
admin ✓

Cluster Login Password
..... ✓

Confirm Password
..... ✓

SSH Username ⓘ
sshuser ✓

SSH Authentication Type
PASSWORD PUBLIC KEY

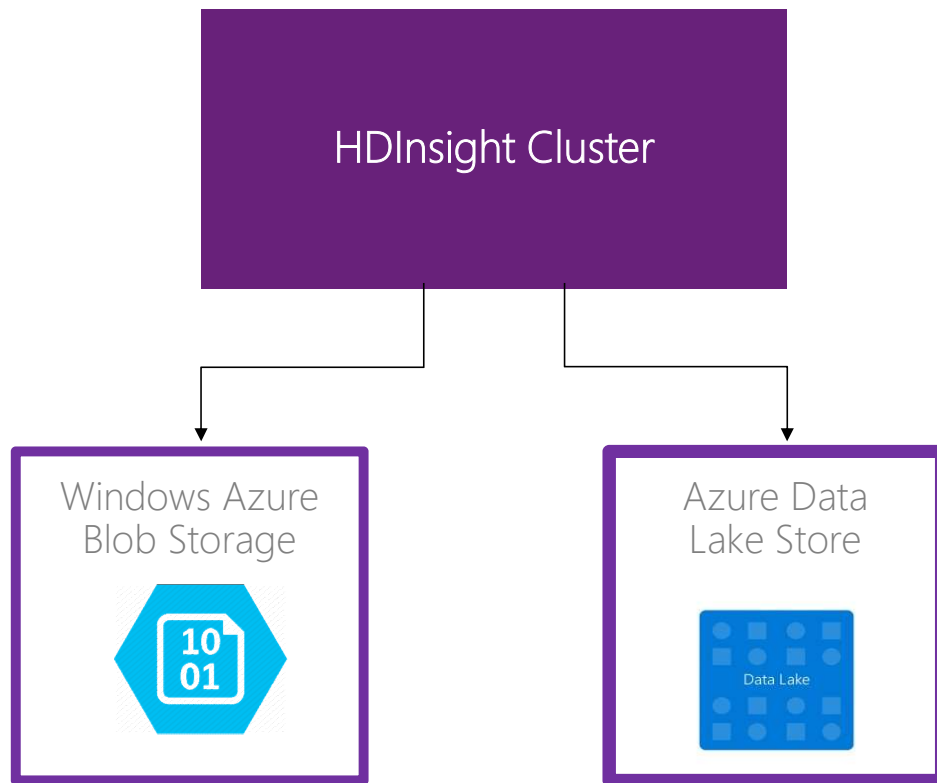
SSH Password
..... ✓

Confirm Password
..... ✓

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 as 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

Microsoft Azure

New > Data + Analytics > New HDInsight Cluster > Data Source

New

Resource groups

All resources

Recent

App Services

Virtual machines (classic)

Virtual machines

SQL databases

Cloud services (classic)

Subscriptions

Browse >

New HDInsight Cluster

Cluster Name
samplehdicluster ✓
.azurehdinsight.net

Subscription
Pay-As-You-Go >

Select Cluster Type ⓘ
Standard Hadoop on Linux (3.2) >

Credentials
Configured >

Data Source ⓘ
Configure required settings >

Node Pricing Tiers
Please configure required settings 🔒

Data Source

The cluster will use this data source as the primary location for most data access, such as job input and log output.

Selection Method ⓘ
From all subscriptions ▼

Create a new storage account ⓘ
wasbstorage ✓
[Select existing](#)

Choose Default Container ⓘ
samplehdiclustercontainer ✓

Location ⓘ
East US >

Cluster AAD Identity ⓘ
Not Configured >

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

Step 1: Create a Service Principal (Azure Active Director ([AAD] Identity) that can represent the cluster

Data Source	Cluster AAD Identity	Create Service Principal
<p>The cluster will use this data source as the primary location for most data access, such as job input and log output.</p>	<p>This Azure Active Directory identity will represent the cluster. The cluster will use this identity to access your Data Lake Store accounts.</p>	<p>We will create a certificate, AD Application, and Service Principal for you. The certificate will be available once the cluster is provisioned.</p>
<p>Selection Method ⓘ</p> <p>From all subscriptions ▼</p> <hr/> <p>* Create a new storage account</p> <p>wasbstorage ✓</p> <p>Select existing</p> <hr/> <p>* Choose Default Container ⓘ</p> <p>samplehdiclustercontainer ✓</p> <hr/> <p>* Location</p> <p>East US ></p> <p>Cluster AAD Identity ⓘ</p> <p>sampleprincipal1 ></p>	<p>Select AD Service Principal</p> <p>Use existing Create new</p> <hr/> <p>* Service Principal ⓘ</p> <p>Not Configured ></p> <p>Manage ADLS Access 🔒</p> <hr/> <p>Service Principal Info:</p> <p>Keep this info if you want to recreate your cluster.</p> <p>Download Certificate</p> <hr/> <p>Object ID:</p> <p>Not Configured 📄</p> <hr/> <p>Application ID:</p> <p>Not Configured 📄</p>	<p>* Service Principal name</p> <p>sampleprincipal1 ✓</p> <hr/> <p>Certificate start date</p> <p>2016-04-03 📅</p> <hr/> <p>Certificate expiration date</p> <p>2017-04-03 📅</p> <hr/> <p>* Certificate password</p> <p>..... ✓</p> <hr/> <p>* Confirm password</p> <p>..... ✓</p> <hr/> <p>i Once you hit Create below, we will create a Azure AD Application and Service Principal on your behalf.</p>
<p>Select</p>	<p>Select</p>	<p>Create</p>

Step 4(2): Specifying ADLS for Storage

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

Data Source

The cluster will use this data source as the primary location for most data access, such as job input and log output.

Selection Method ⓘ

From all subscriptions

* Create a new storage account

wasbstorage

Select existing

* Choose Default Container ⓘ

samplehdicustercontainer

* Location

East US

Cluster AAD Identity ⓘ

sampleprincipal1

Select

Cluster AAD Identity

This Azure Active Directory identity will represent the cluster. The cluster will use this identity to access your Data Lake Store accounts.

Select AD Service Principal

Use existing Create new

* Service Principal ⓘ

sampleprincipal1

Upload Existing Certificate

Select a file

Certificate

Uploaded successfully

* Certificate Password

.....

Manage ADLS Access

Service Principal Info:

Keep this info if you want to recreate your cluster.

Download Certificate

Select

Data Lake Store Root Folder Access

[Learn more](#)

Set the service principal's permissions on your Azure Data Lake Store (ADLS) accounts. You can change the root folder permissions of an ADLS account only if you are the owner of the root folder.

ADLS accounts whose root folders you own

DATA LAKE STORE	READ	WRITE	EXECUTE
adlsfordatagtesting	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
sampleadlsstorage1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
snapadlsforhdinsight	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
snapadlsforhdinsight2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Other ADLS accounts

DATA LAKE STORE	READ	WRITE	EXECUTE
tpchadls	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Save Permissions

Step 5: Specify Cluster Configuration

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

New HDInsight Cluster

Cluster Name

samplehdccluster

.azurehdinsight.net

Subscription

Pay-As-You-Go

Select Cluster Type

Standard Hadoop on Linux (3.2)

Credentials

Configured

Data Source

wasbstorage (East US)

Node Pricing Tiers

Please configure required settings

Optional Configuration

Resource Group

ADLA_Benchmark

☐ Pin to dashboard

Create

Node Pricing Tiers

To learn more, visit our pricing page. [Learn more](#)

Number of Worker nodes

4

Worker Nodes Pricing Tier

D3 (4 nodes, 16 cores)

Head Node Pricing Tier

D3 (2 nodes, 8 cores)

WORKER NODES

0.62 x 4 = 2.49

HEAD NODES

0.62 x 2 = 1.24

TOTAL COST

3.73

USD/HOUR (ESTIMATED)

24 of 60 cores would be used in East US.

This price estimate does not include storage costs, network egress costs, or subscription discounts.

Questions?

[Contact billing support.](#)

Note: Clusters with more than 32 Worker nodes require a Head node size with at least 8 cores and 14 GB RAM.

Select

Choose your pricing tier

Browse the available pricing tiers and their features. [Learn more](#)

Recommended

[View all](#)

D3 Optimized

4 Cores

14 GB RAM

8 Disks

200 GB Local SSD

0.62

USD/HOUR (ESTIMATED)

D4 Optimized

8 Cores

28 GB RAM

16 Disks

400 GB Local SSD

1.24

USD/HOUR (ESTIMATED)

D12 Optimized

4 Cores

28 GB RAM

8 Disks

200 GB Local SSD

0.76

USD/HOUR (ESTIMATED)

Select

Microsoft

18

Step 6: Optional Configurations

Optionally you can configure:

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

The screenshot displays the 'Optional Configuration' page in the Microsoft Azure portal for a new HDInsight cluster. The breadcrumb navigation at the top reads: 'Microsoft Azure > New > Data + Analytics > New HDInsight Cluster > Optional Configuration'. The left-hand navigation pane includes links for 'New', 'Resource groups', 'All resources', 'Recent', 'App Services', 'Virtual machines (classic)', 'Virtual machines', 'SQL databases', 'Cloud services (classic)', and 'Subscriptions'. The main content area is divided into two panels. The left panel, titled 'New HDInsight Cluster', contains the following configuration items: 'Cluster Name' (samplehdiclusterv), 'Subscription' (Pay-As-You-Go), 'Select Cluster Type' (Standard Hadoop on Linux (3.2)), 'Credentials' (Configured), 'Data Source' (wasbstorage (East US 2)), 'Node Pricing Tiers' (D3/D3), and 'Resource Group' (ADLA_Benchmark). An 'Optional Configuration' link is highlighted in blue. At the bottom of this panel are a 'Pin to dashboard' checkbox and a 'Create' button. The right panel, titled 'Optional Configuration', lists four optional settings, all marked as 'Not Configured': 'Virtual Network', 'External Metastores', 'Script Actions', and 'Linked Storage Accounts'. Each item has a right-pointing arrow. A 'Select' button is located at the bottom of this panel.

Cluster Creation

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

The screenshot shows the HDInsight cluster management interface for a cluster named 'samplehdicluster'. The top navigation bar includes links for Settings, Dashboard, Secure Shell, Scale Cluster, Delete, and Move. A blue banner with an information icon and the text 'In Progress...' is prominently displayed. Below this, the 'Essentials' section provides key cluster details in a two-column layout:

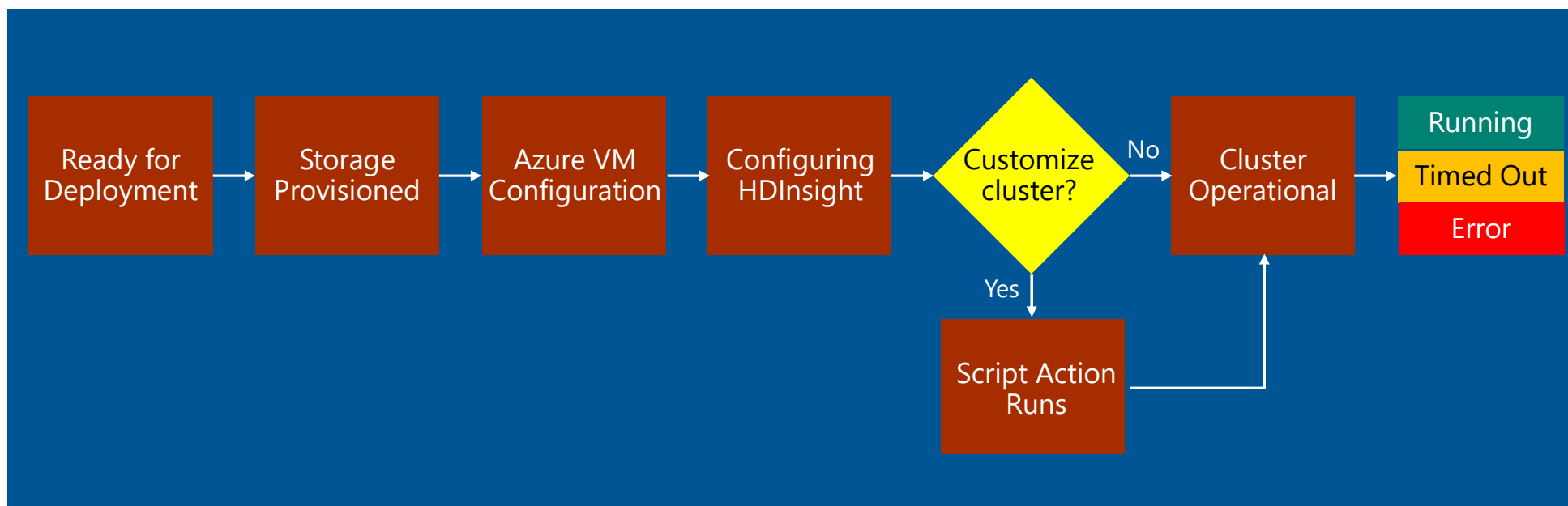
Property	Value
Resource group	ADLA_Benchmark
Status	State Azure VM Configuration
Location	East US 2
Subscription name	Pay-As-You-Go
Subscription id	bc2d3f0d-ae1d-4f7c-af1e-b2e6932bed5f
URL	samplehdicluster.azurehdinsight.net
Cluster Type	Standard Hadoop on Linux
Head Node, Worker Nodes	D3 (x2), D3 (x4)

Additional links for 'Learn more', 'Documentation', and 'Quickstart' are provided. An 'All settings' button is located at the bottom right of the Essentials section. The right-hand sidebar, titled 'Settings', contains several expandable sections: CONFIGURATION (Cluster Login, Scale Cluster, Secure Shell, HDInsight Partner, External Metastores), GENERAL (Script Actions, Apps), PROPERTIES (Properties, Azure Storage Keys, Cluster AAD Identity), and RESOURCE MANAGEMENT (Users, Tags).

Script Actions

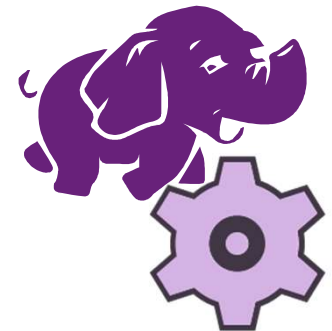
Customize with Script Actions

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



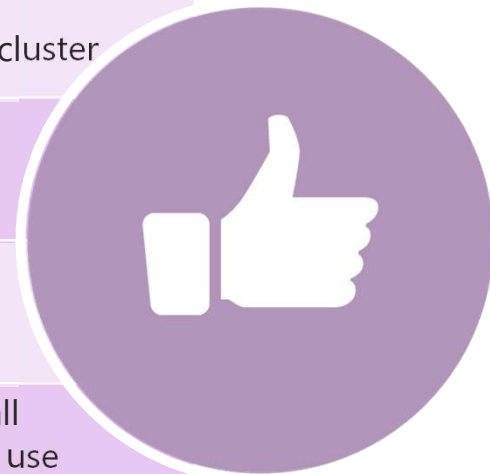
Script Actions: Key concepts

- 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

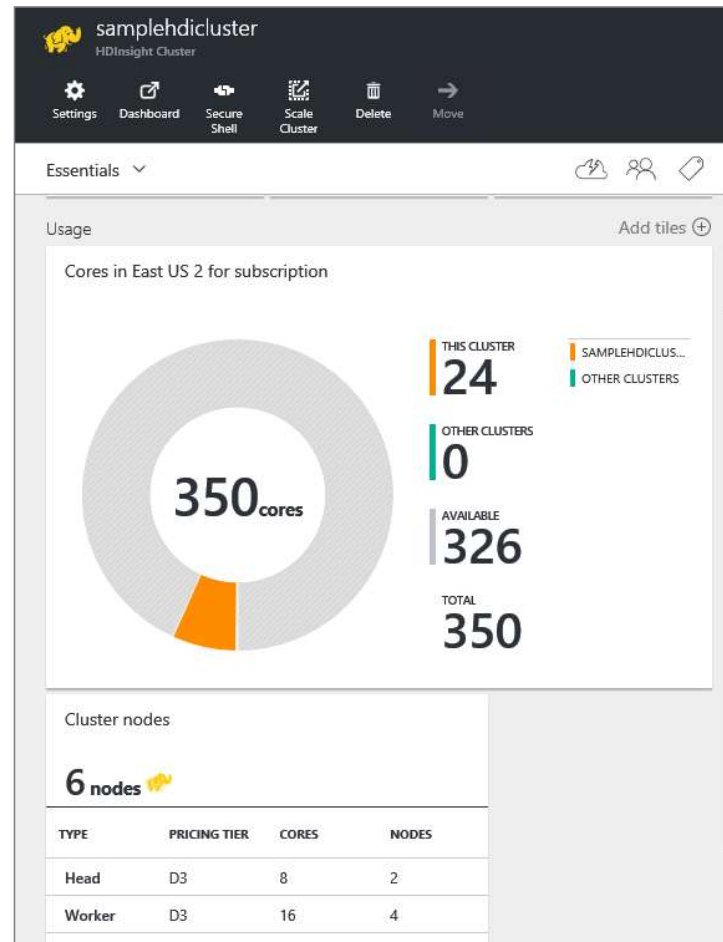
```
var tokenCreds = GetTokenCloudCredentials(); //See notes section for definition of this function
var subCloudCredentials = GetSubscriptionCloudCredentials(tokenCreds, "My Subscription ID");
var resourceManagementClient = new ResourceManagementClient(subCloudCredentials);
var rpResult = resourceManagementClient.Providers.Register("Microsoft.HDIInsight");
_hdiManagementClient = new HDInsightManagementClient(subCloudCredentials);
//specify the cluster configuration details
var parameters = new ClusterCreateParameters {
    ClusterSizeInNodes = 15,
    ClusterType = HDInsightClusterType.Hadoop,
    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
P5s30v6blljsLZYTnyReNUa5LtFw7eauGr
----- END SSH2 PUBLIC KEY -----";

};
//Now create the cluster
_hdiManagementClient.Clusters.Create("MyResourceGroup", "MySampleCluster", parameters);
```

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.

The screenshot displays the Azure HDInsight cluster management interface for a cluster named 'samplehdicluster'. The interface is divided into two main sections: 'Essentials' and 'Scale Cluster'.

Essentials Section:

- Resource group:** ADLA_Benchmark
- Status:** Running
- Location:** East US 2
- Subscription name:** Pay-As-You-Go
- Subscription id:** bc2d3f0d-ae1d-4f7c-af1e-b2e6932bed5f
- URL:** samplehdicluster.azurehdinsight.net
- Cluster Type:** Standard Hadoop on Linux
- Head Node, Worker Nodes:** D3 (x2), D3 (x4)

Scale Cluster Section:

- Number of Worker nodes:** 4 (highlighted with a red box and a green checkmark)
- Worker Nodes Pricing Tier:** D3 (4 nodes, 16 cores)
- Head Node Pricing Tier:** D3 (2 nodes, 8 cores)
- Worker Nodes Cost:** 0.62 x 4 = 2.49
- Head Nodes Cost:** 0.62 x 2 = 1.24
- TOTAL COST:** 3.73 USD/HOUR (ESTIMATED)
- Usage:** 24 of 350 cores would be used in East US 2.

Quick Links Section:

- Cluster Dashboard
- Ambari Views
- Scale Cluster** (highlighted with a blue box)


Usage Section:







- Add tiles (+)

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





Essentials ^

Resource group

[ADLA_Benchmark](#)

Status

Running

Location

East US 2

Subscription name

[Pay-As-You-Go](#)

Subscription id

bc2d3f0d-ae1d-4f7c-af1e-b2e6932bed5f

URL

[samplehdicluster.azurehdinsight.net](#)

Cluster Type

Standard Hadoop on Linux

Head Node, Worker Nodes

D3 (x2), D3 (x4)

Learn more

[Documentation](#)


Getting Started


[Quickstart](#)


All settings →

Quick Links

Add tiles +

 Cluster Dashboard

 Ambari Views

 Scale Cluster

Users

samplehdicluster

+ Add



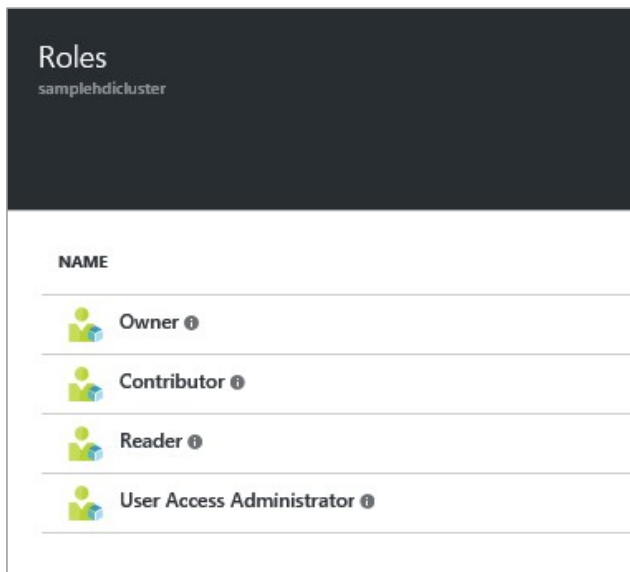
USER	ROLE	ACCESS	
 [redacted]@outlook.com	Owner	Inherited	...
 Subscription admins ⓘ	Owner	Inherited	...

Roles

samplehdicluster

NAME
 Owner ⓘ
 Contributor ⓘ
 Reader ⓘ
 User Access Administrator ⓘ

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 screenshot shows the 'samplehdccluster' HDInsight Cluster settings page in the Azure Portal. The top navigation bar includes links for Settings, Dashboard, Secure Shell, Scale Cluster, Delete, and Move. The main content area is divided into two columns. The left column, titled 'Essentials', displays cluster details: Resource group (ADLA_Benchmark), Status (Running), Location (East US 2), Subscription name (Pay-As-You-Go), and Subscription id (bc2d3f0d-ae1d-4f7c-af1e-b2e6932bed5f). It also shows the URL (samplehdccluster.azurehdinsight.net), Cluster Type (Standard Hadoop on Linux), and Head Node/Worker Nodes (D3 (x2), D3 (x4)). A 'Quick Links' section contains tiles for Cluster Dashboard, Ambari Views, and Scale Cluster. The right column, titled 'Settings', contains a 'Filter settings' search bar and a list of settings categories: SUPPORT + TROUBLESHOOTING (Audit logs), GETTING STARTED (Quick Start), CONFIGURATION (Cluster Login, Scale Cluster, Secure Shell, HDInsight Partner, External Metastores), GENERAL (Script Actions, Apps), PROPERTIES (Properties, Azure Storage Keys, Cluster AAD Identity), and RESOURCE MANAGEMENT (Users, Tags). A purple arrow points from the text box on the right to the 'Settings' column.

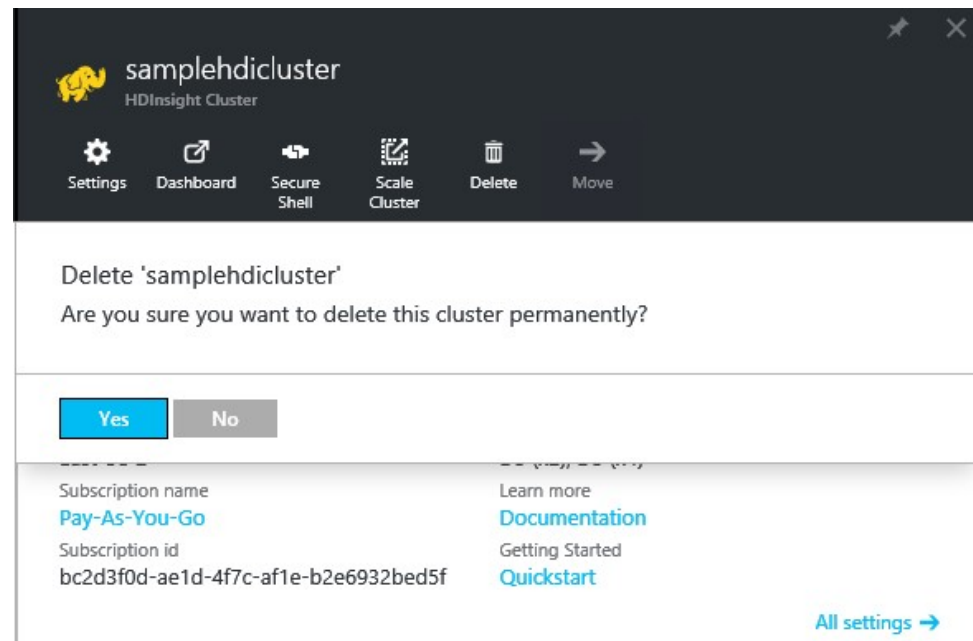
The Azure Portal lets you view and change all these settings after the cluster has been created

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

Settings
ntlive19hdbm0531

Events
Filter Columns Hide chart Export

Export Audit Logs (PREVIEW)
Save Discard Reset

Filter settings

SUPPORT + TROUBLESHOOTING

- Audit logs

GETTING STARTED

- Quick Start

CONFIGURATION

- Cluster Login
- Scale Cluster
- Secure Shell
- HDInsight Partner
- External Metastores

GENERAL

- Script Actions
- Apps

PROPERTIES

- Properties

Filtered for past week
by resource /subscriptions/15c5cb6e-191a-40ea-9f69-08207a17fe97/resourceGroups/_ElasticData...
event category = All, levels = All

10:29 AM

CRITICAL 0 ERROR 0 WARNING 0 INFORMATIONAL 1

Filter items ...

OPERATION	LEVEL	STATUS	RESOURCE	TI...
Write Clusters	Informational	Succeeded	...clusters/ntlive19h...	1 d ago

Archive your Audit logs to a storage account or stream them to an Azure Event Hub. Diagnostic data is billed at normal storage rates.

* Subscription
Azure conversion

* Regions
0 selected

* STORAGE ACCOUNT
Configure required settings

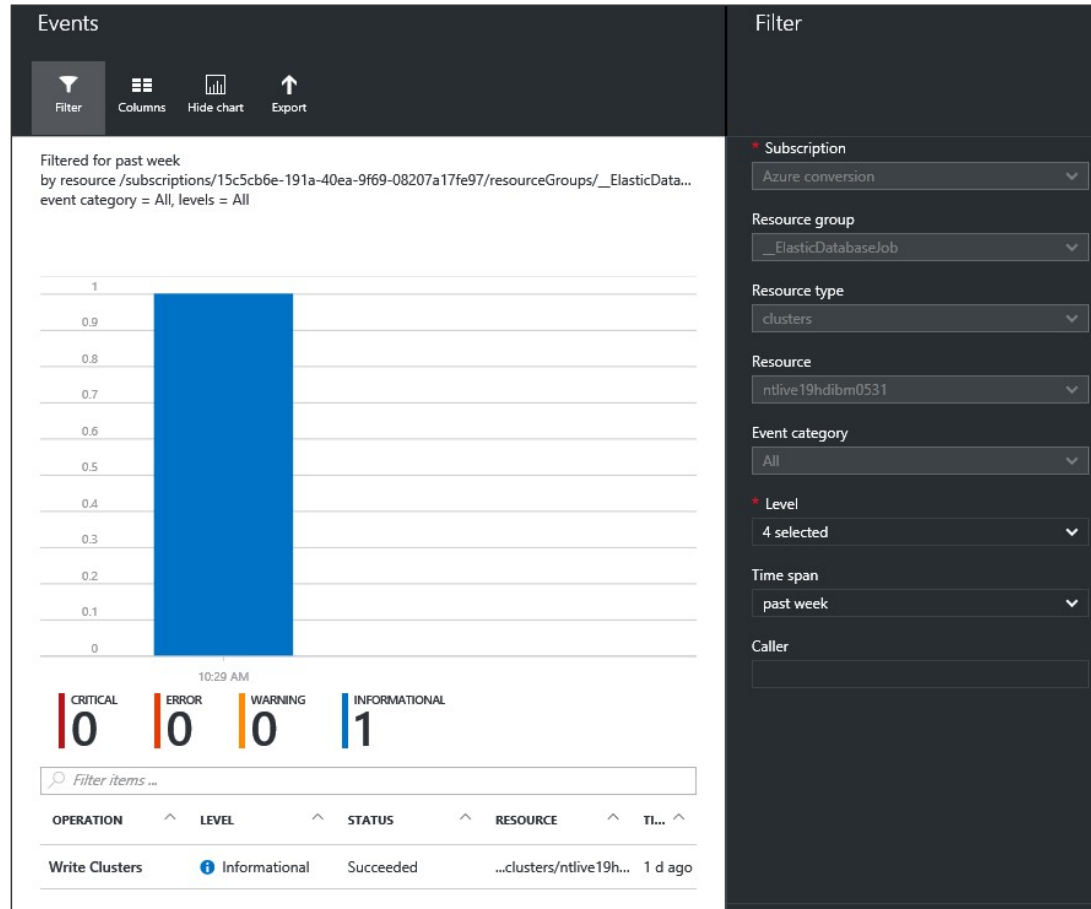
Retention (days)
0

AZURE EVENT HUB
Optionally configure Event Hub

Filtering Audit Logs

Audit Logs entries can be filtered by:

- Time
- Type
- Level
- ...



Level

4 selected

- ☒ Critical
- ☒ Error
- ☒ Warning
- ☒ Informational

Level

4 selected

- past 1 hour
- past 24 hours
- past week
- custom

Configuring Hadoop

Core Hadoop Configuration files

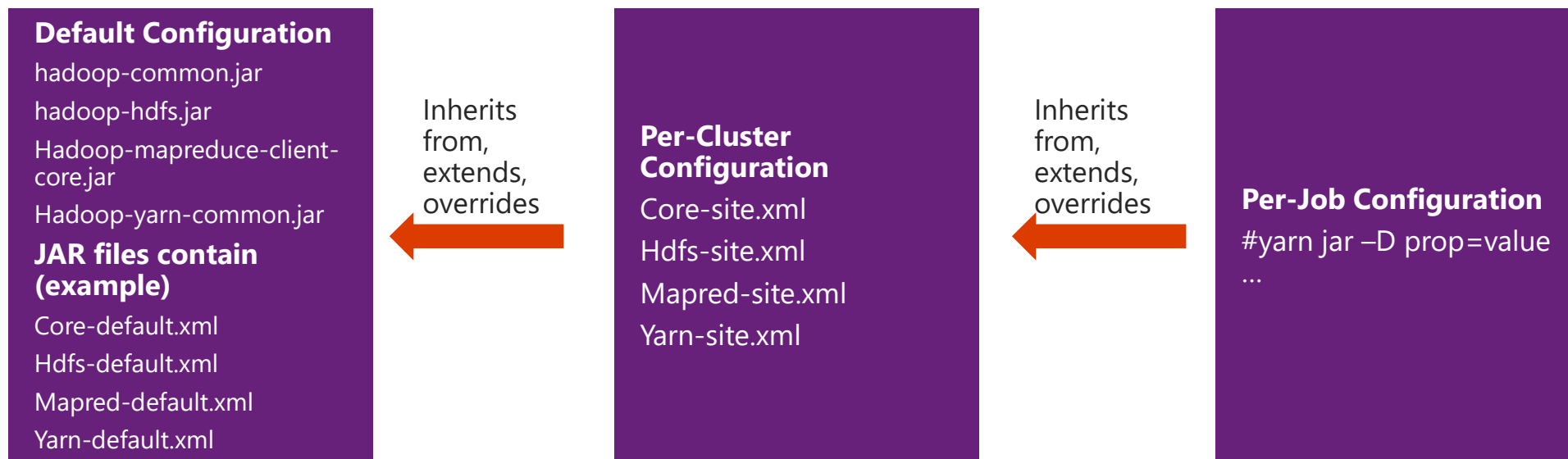
Administrators configure settings for HDFS, Yarn and MapReduce (and other services) through these files

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-env.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)

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

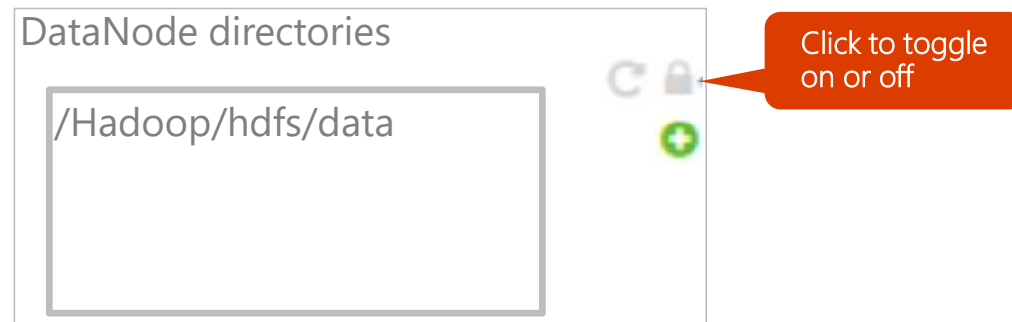
Madhu Reddy, 4/11/2016

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

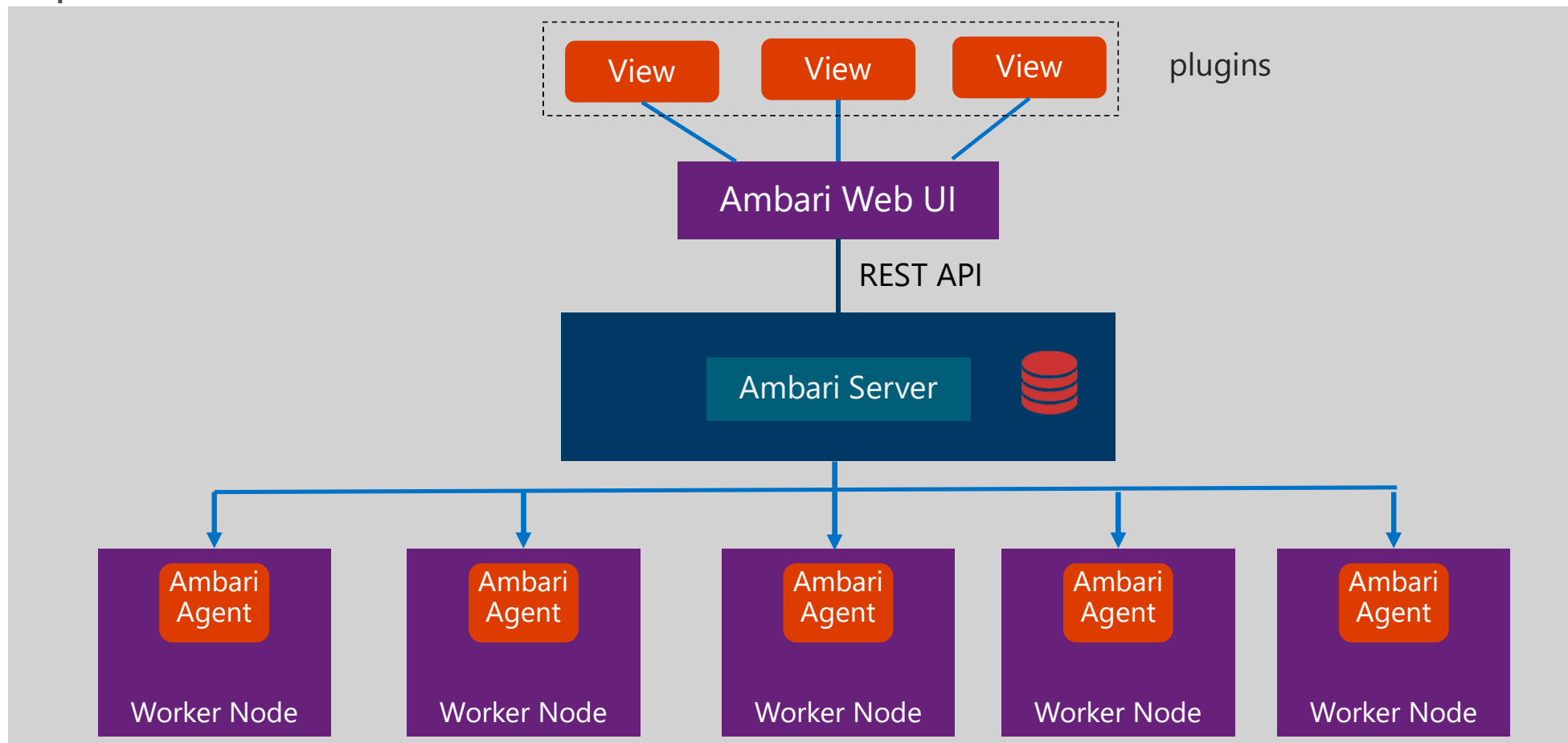
Systems Administrators	Provisioning	Provides step-by-step wizard for installing Hadoop services across any number of hosts
		Handles configuration of Hadoop services for the cluster
	Managing	Provides central management for starting, stopping, and reconfiguring Hadoop services across the entire cluster
	Monitoring	Provides dashboard for monitoring health and status of the Hadoop cluster
		Leverages Ambari Metrics System for metrics collection
Application Developers and System Integrators		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)
		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

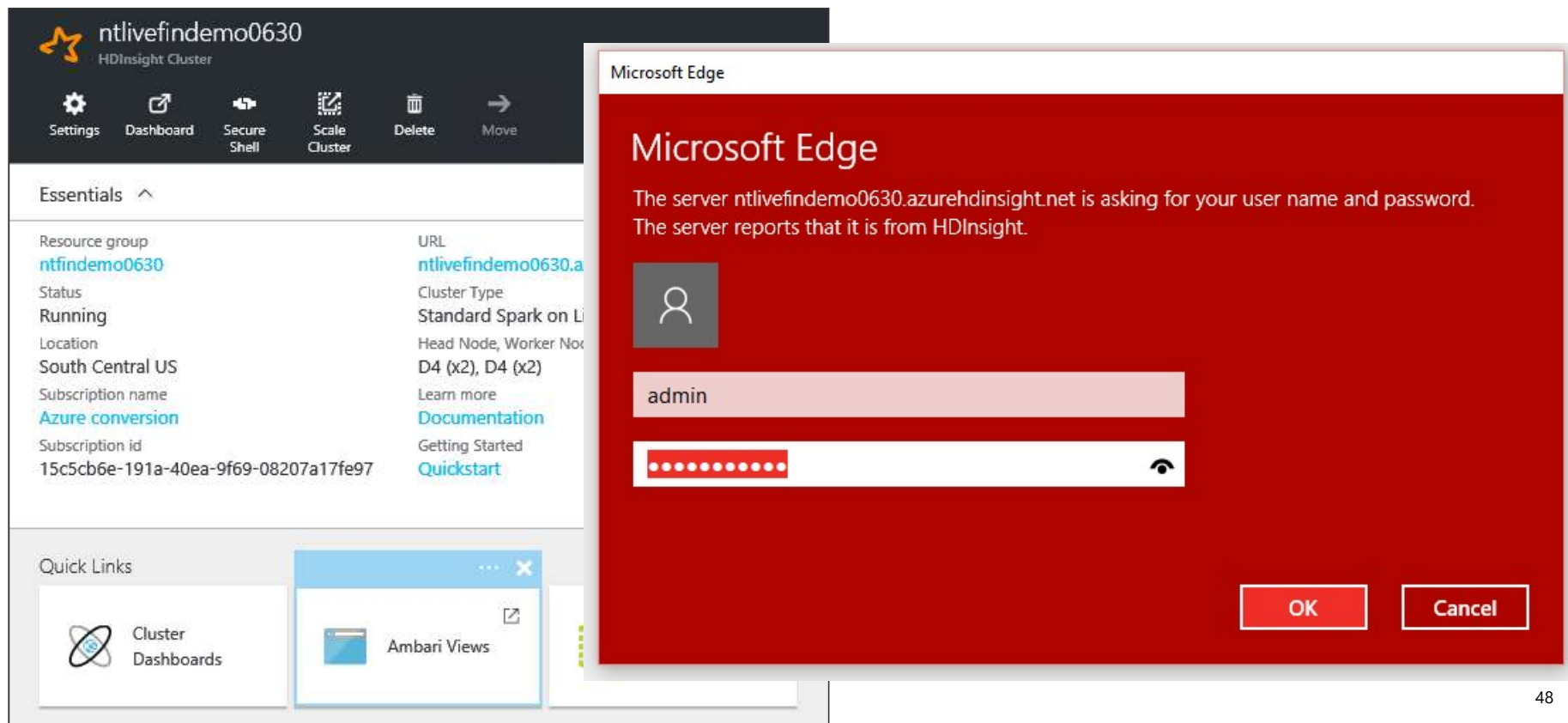
Apache Ambari: Architecture Overview



Managing HDInsight with Amabari

HDInsight and Ambari

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



Ambari Web UI Dashboard

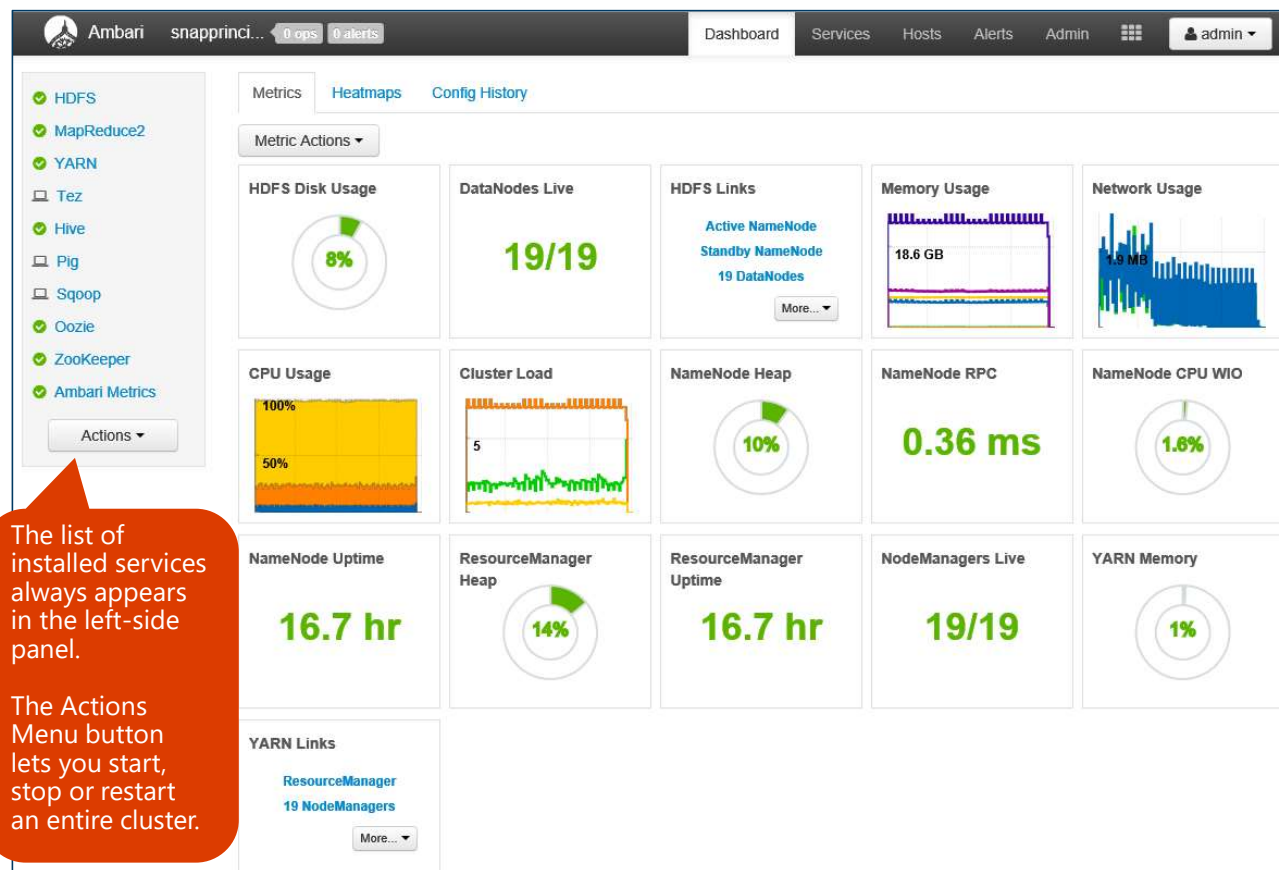
Ambari Web UI: Dashboard Metrics

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

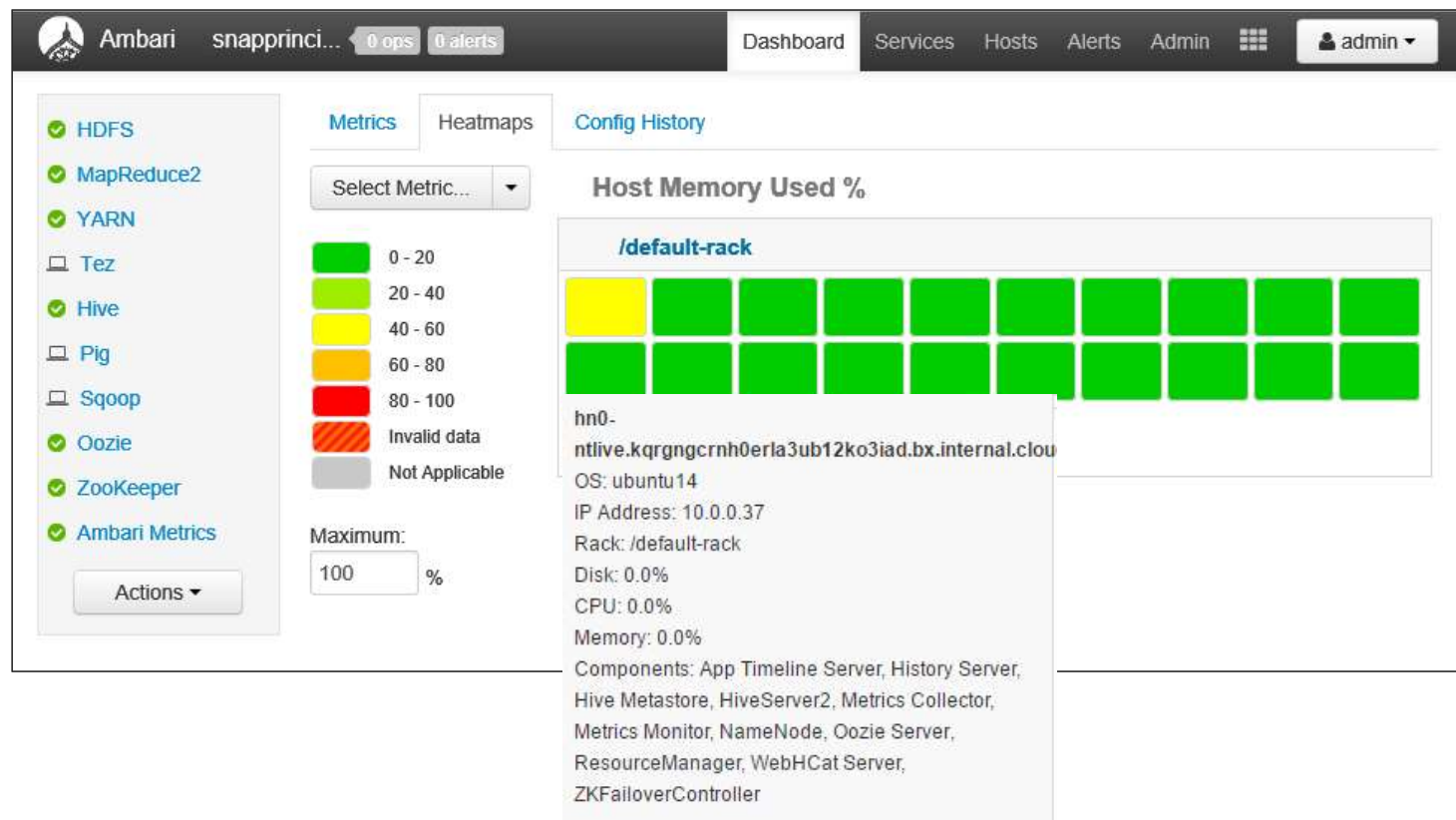
The color can be customized by configuring the % thresholds

The screenshot shows the Ambari dashboard interface. A 'Customize Widget' dialog box is open in the center. The dialog has a title bar 'Customize Widget' and a light blue instruction box that says: 'Edit the percentage thresholds to change the color of current pie chart. Enter two numbers between 0 to 100'. Below this is a horizontal color bar with three segments: green (0-50%), orange (50-75%), and red (75-100%). Below the bar are two input fields: the first contains '50' and the second contains '75'. The range is marked from 0 to 100. At the bottom right of the dialog are 'Cancel' and 'Apply' buttons. The background dashboard shows a sidebar with service status (HDFS, MapReduce2, YARN, Tez, Hive, Pig, Sqoop, Oozie, ZooKeeper, Ambari Metrics, Slider) and a main area with various widgets like 'NameNode Uptime', 'ResourceManager Heap', 'ResourceManager Uptime', 'NodeManagers Live', and 'YARN Memory'. A 'Network Usage' widget shows a bar chart with a value of 4.7 MB.

Ambari Dashboard: HeatMap

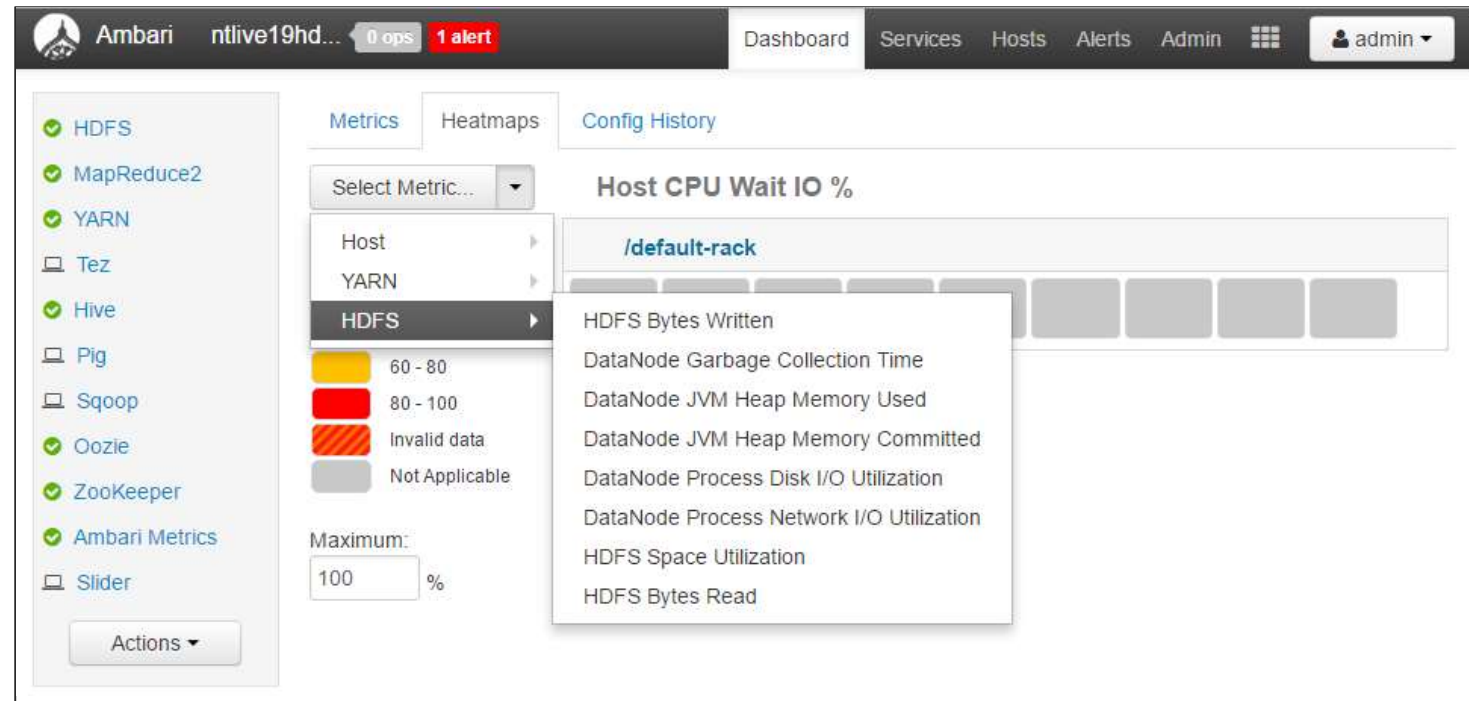
The Dashboard Heatmap view provides a color-coded view of each of the nodes in the cluster for selected metrics.

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

Service	Config Group	Created	Author	Notes
V2 HDFS	Default Current	Mon, Apr 04, 2016 14:42	admin	
V2 MapReduce2	Default Current	Mon, Apr 04, 2016 14:42	admin	
V2 Oozie	Default Current	Mon, Apr 04, 2016 14:42	admin	
V2 Hive	Default Current	Mon, Apr 04, 2016 14:42	admin	
V1 Ambari Metrics	Default Current	Mon, Apr 04, 2016 14:42	admin	
V1 YARN	Default Current	Mon, Apr 04, 2016 14:42	admin	
V1 HDFS	Default	Mon, Apr 04, 2016 14:42	admin	
V1 MapReduce2	Default	Mon, Apr 04, 2016 14:42	admin	
V1 Oozie	Default	Mon, Apr 04, 2016 14:42	admin	

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

The screenshot shows the Ambari UI Alerts page. A modal window titled "1 Critical or Warning Alerts" is displayed over the main alert list. The modal contains a table with the following data:

Service / Host	Alert Definition Name	Status
Ambari	Ambari Server Alerts There are 15 stale alerts from 15 ho...	CRIT for 2 hours

Below the table, there is a "Go to Alerts Definitions" button highlighted with a red box. The background shows a list of alert definitions with columns for Name, Status (OK), Alert Definition Name, Time (9 hours ago), and State (Enabled).

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.

The screenshot shows the Ambari web interface. The top navigation bar includes 'Dashboard', 'Services', 'Hosts', 'Alerts', and 'Admin'. The 'Alerts' tab is selected, showing a summary of '0 ops' and '1 alert'. The main content area displays the 'Host Disk Usage' alert configuration. A red box highlights the 'Edit' button in the top right corner of the configuration panel. The configuration includes a description, a check interval of 1 minute, and a state of 'Enabled'. To the right of the configuration panel, there is a sidebar with details: 'State: Enabled', 'Service: Ambari', 'Component: Ambari Agent', 'Type: SCRIPT', 'Groups: AMBARI Default', 'Last: Mon, Apr 04, 2016', and 'Changed: 21:49'. Below the configuration panel, there is a table titled 'Instances' showing the status of the alert across different hosts.

Service / Host	Status	24-Hour	Response
Ambari / hn0- ntlive.kqrgngcrnh0erla3ub12ko3iad.bx.internal.cloudapp.net	OK for 9 hours	1	Capacity Used: [1.07%, 11.3 GB], Capacity Total: [1.1 T...
Ambari / hn1- ntlive.kqrgngcrnh0erla3ub12ko3iad.bx.internal.cloudapp.net	OK for 9 hours	1	Capacity Used: [1.11%, 11.8 GB], Capacity Total: [1.1 T...
Ambari / wn0- ntlive.kqrgngcrnh0erla3ub12ko3iad.bx.internal.cloudapp.net	OK for 2 hours	2	Capacity Used: [1.39%, 14.7 GB], Capacity Total: [1.1 ...

Ambari UI: Services

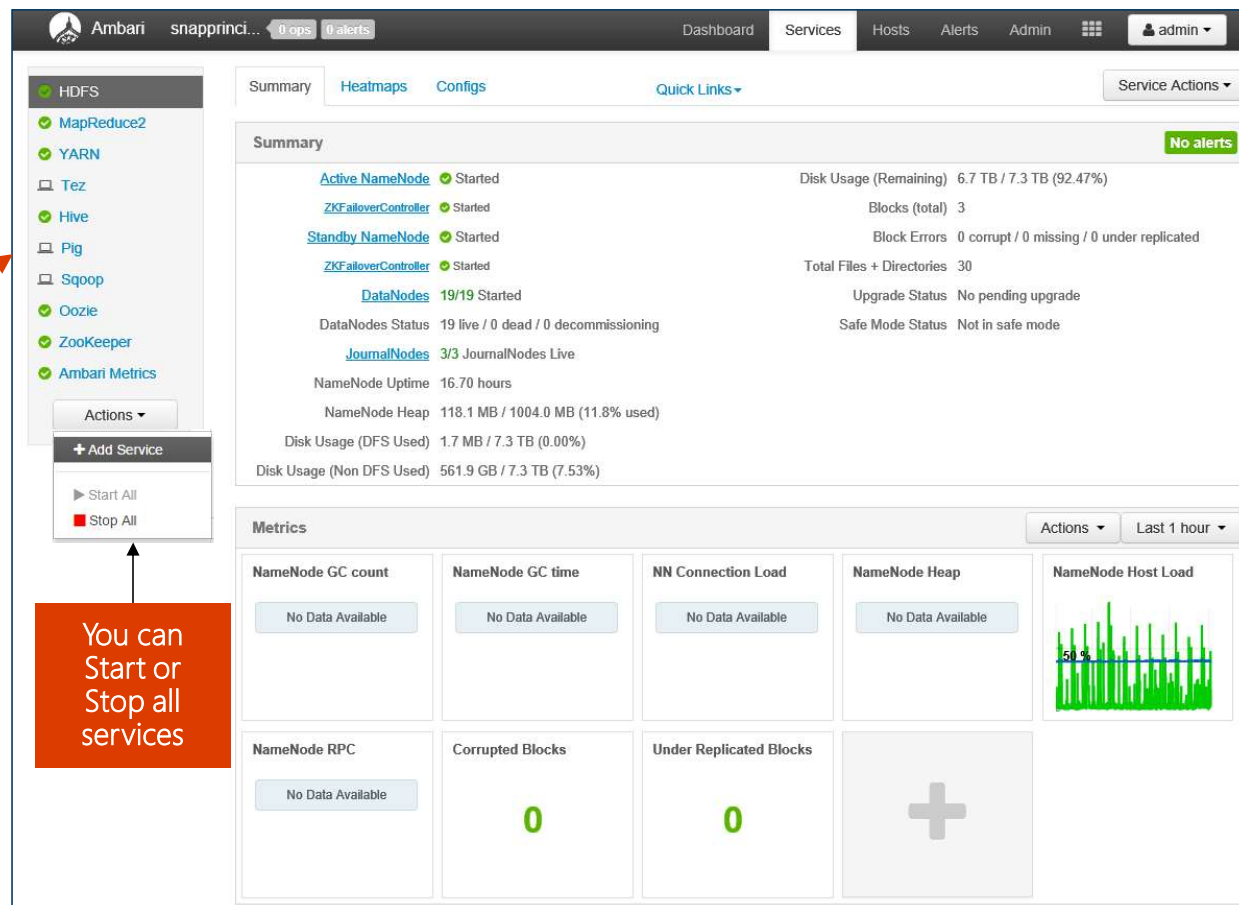
Ambari Web UI: Services

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.

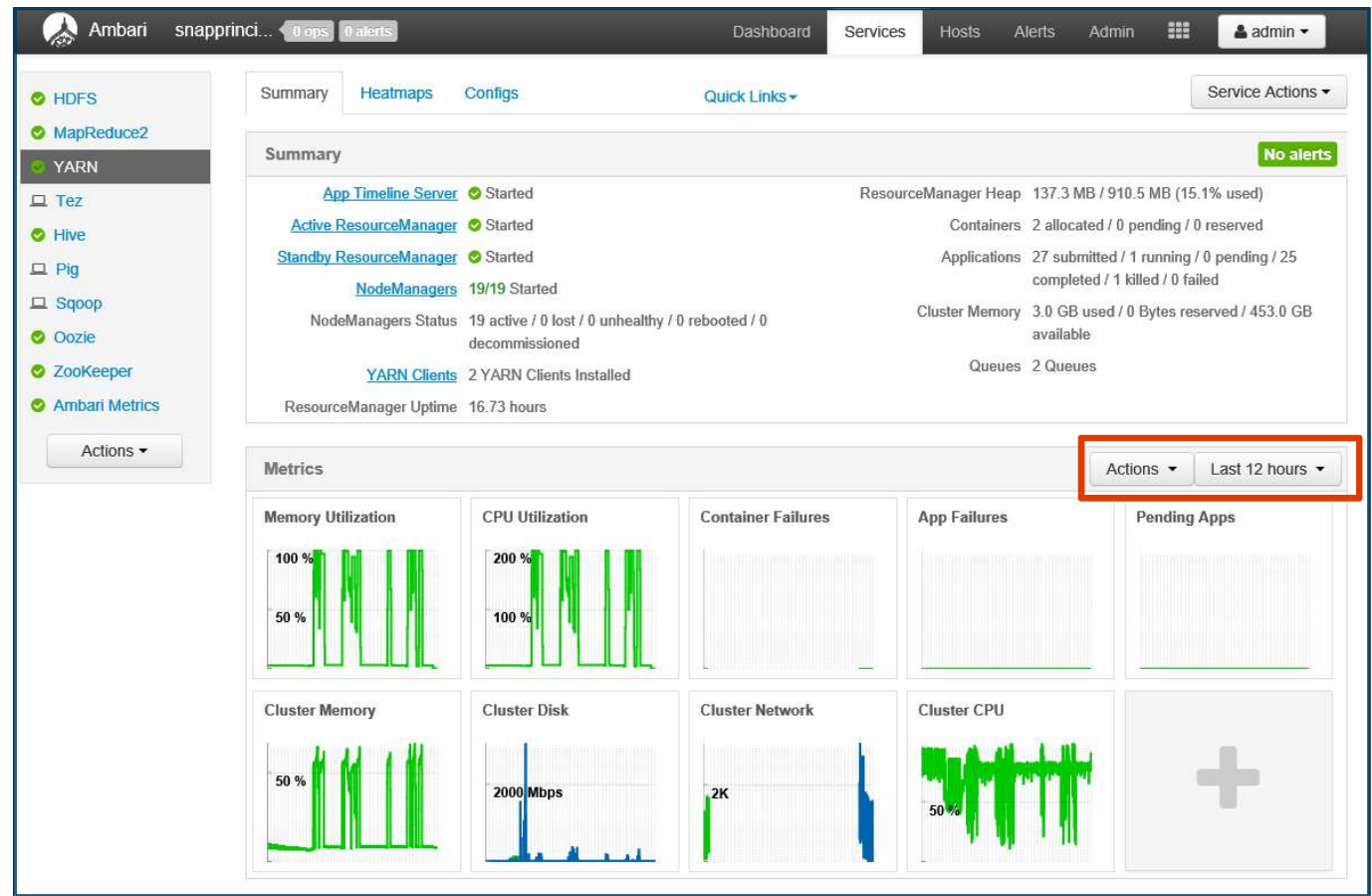
Maintenance Mode should be enabled when making cluster hardware or software changes. It suppresses Ambari alerts, warnings and status change indicators

The screenshot displays the Ambari Web UI interface. On the left, a sidebar lists services: HDFS, MapReduce2, YARN, Tez, Hive, Pig, Sqoop, Oozie, ZooKeeper, and Ambari Metrics. The 'Actions' button is visible below the list. The main content area shows the 'Summary' tab for the HDFS service. A 'Service Actions' dropdown menu is open on the right, listing various actions: Start, Stop, Restart All, Restart DataNodes, Restart JournalNodes, Restart ZKFailoverControllers, Move NameNode, Run Service Check, Turn On Maintenance Mode (highlighted with a red box), Rebalance HDFS, and Download Client Configs. The background shows the HDFS summary page with details like 'Active NameNode', 'Standby NameNode', 'DataNodes' (19/19 Started), 'JournalNodes' (3/3 Live), and 'Disk Usage'.

Ambari Web UI: Services (YARN)

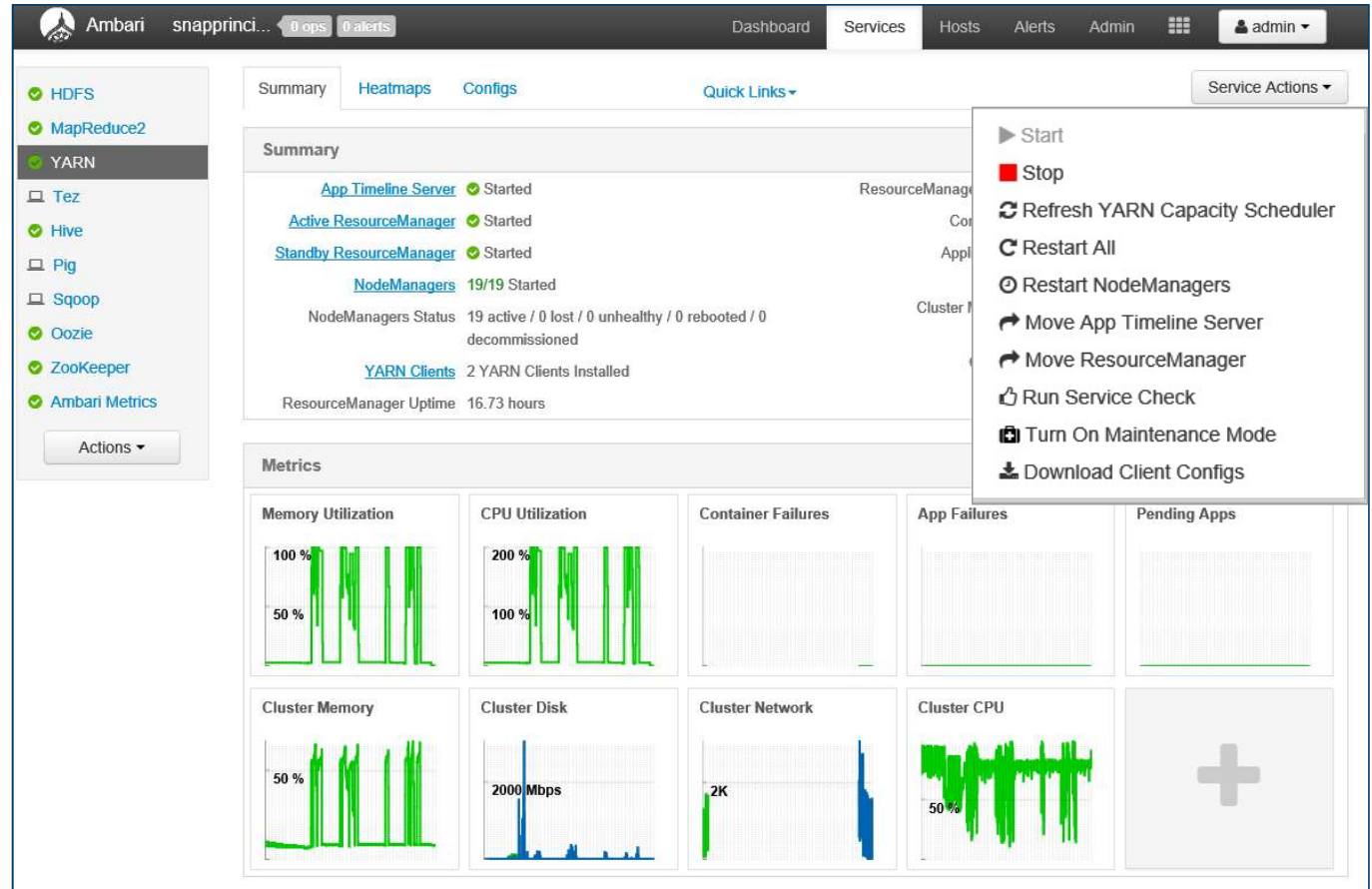
The **Services** page details
for YARN for the last 1 hour

... and the last 12 hours



Ambari: YARN "Service Actions"

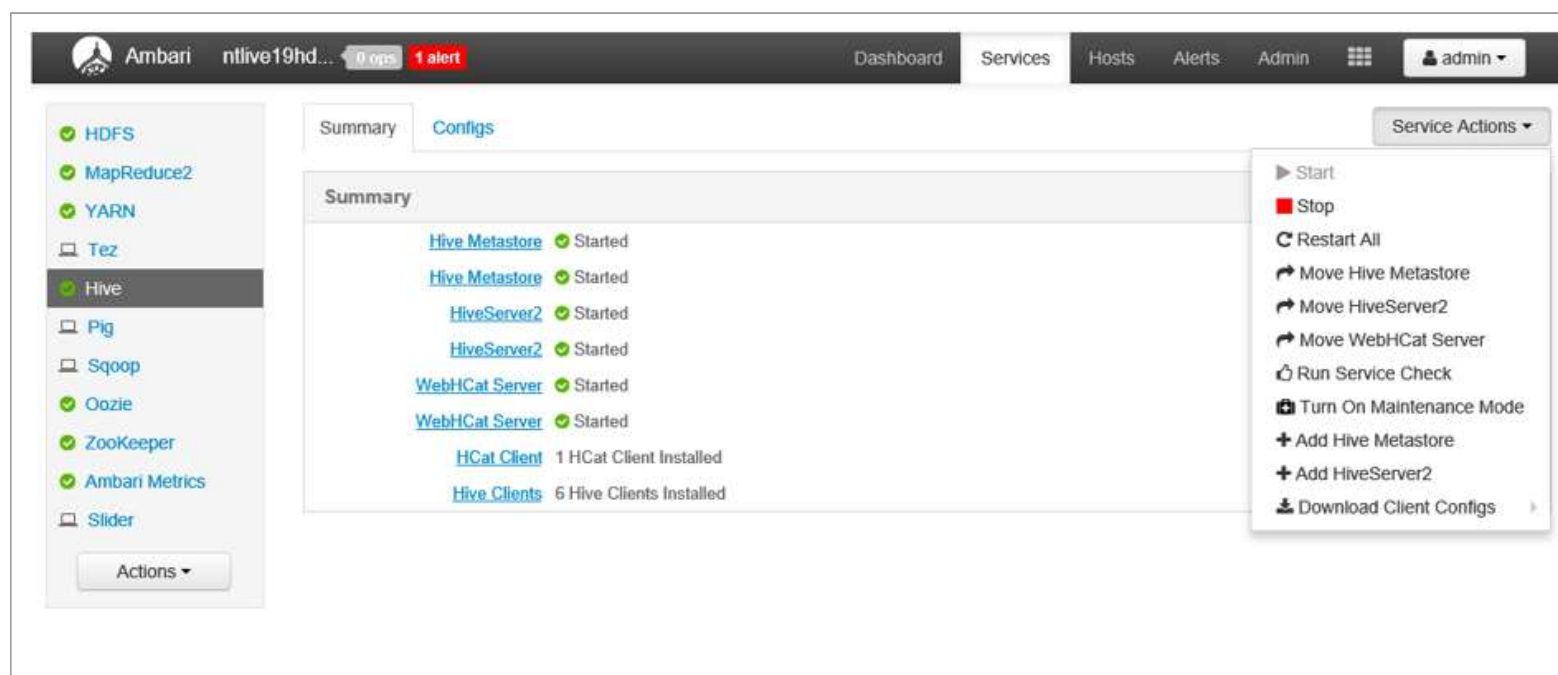
Here are the list of actions that can be taken with YARN.



The screenshot shows the Ambari web interface for the YARN service. The left sidebar lists various services: HDFS, MapReduce2, YARN (selected), Tez, Hive, Pig, Sqoop, Oozie, ZooKeeper, and Ambari Metrics. The main content area displays the YARN service summary, including status (Started), NodeManagers (19/19 Started), and YARN Clients (2 YARN Clients Installed). A 'Service Actions' dropdown menu is open, showing the following options: Start, Stop, Refresh YARN Capacity Scheduler, Restart All, Restart NodeManagers, Move App Timeline Server, Move ResourceManager, Run Service Check, Turn On Maintenance Mode, and Download Client Configs. Below the summary, there are several metrics charts: Memory Utilization, CPU Utilization, Container Failures, App Failures, Pending Apps, Cluster Memory, Cluster Disk, Cluster Network, and Cluster CPU.

Ambari Web UI: Hive Service Actions

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



The screenshot displays the Ambari Web UI interface for the Hive service. The top navigation bar includes the Ambari logo, the cluster name 'ntlive19hd...', and status indicators for '0 ops' and '1 alert'. The main navigation tabs are 'Dashboard', 'Services', 'Hosts', 'Alerts', and 'Admin'. The 'Services' tab is active, showing a list of services on the left: HDFS, MapReduce2, YARN, Tez, Hive (selected), Pig, Sqoop, Oozie, ZooKeeper, Ambari Metrics, and Slider. The 'Hive' service is highlighted, and its 'Summary' tab is selected. The summary shows the status of various Hive components: Hive Metastore (Started), HiveServer2 (Started), WebHCat Server (Started), HCat Client (1 HCat Client Installed), and Hive Clients (6 Hive Clients Installed). A 'Service Actions' dropdown menu is open, displaying a list of actions: Start, Stop, Restart All, Move Hive Metastore, Move HiveServer2, Move WebHCat Server, Run Service Check, Turn On Maintenance Mode, Add Hive Metastore, Add HiveServer2, and Download Client Configs.

Service	Status
Hive Metastore	Started
Hive Metastore	Started
HiveServer2	Started
HiveServer2	Started
WebHCat Server	Started
WebHCat Server	Started
HCat Client	1 HCat Client Installed
Hive Clients	6 Hive Clients Installed

- Start
- Stop
- Restart All
- Move Hive Metastore
- Move HiveServer2
- Move WebHCat Server
- Run Service Check
- Turn On Maintenance Mode
- Add Hive Metastore
- Add HiveServer2
- Download Client Configs

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.

The screenshot shows the Ambari Web UI interface. The top navigation bar includes links for Dashboard, Services, Hosts, Alerts, and Admin. The Hosts page is active, displaying a table of hosts with columns for Name, IP Address, Rack, Cores, RAM, Disk Usage, Load Avg, Versions, and Components. A modal window titled "Components" is open, showing the components for the host "hn0-snappr.rvofooxckjyu3c1i1gfwaigq0c.cx.internal.cloudapp.net". The components listed are: History Server, Hive Client, Hive Metastore, HiveServer2, MapReduce2 Client, Metrics Collector, and Metrics Monitor. The modal window has an "OK" button at the bottom right.

Name	IP Address	Rack	Cores	RAM	Disk Usage	Load Avg	Versions	Components
Any	Any	Any	Any	Any	Any	Any	Filter	Filter
hn0-snappr.rvofooxckjyu3...								20 Components
hn1-snappr.rvofooxckjyu3...								14 Components
wn0-snappr.rvofooxckjyu...								7 Components
wn1-snappr.rvofooxckjyu...								7 Components
wn10-snappr.rvofooxckjy...								7 Components
wn11-snappr.rvofooxckjy...								7 Components
wn12-snappr.rvofooxckjy...								7 Components
wn13-snappr.rvofooxckjy...								7 Components
wn14-snappr.rvofooxckjy...								7 Components
wn15-snappr.rvofooxckjy...								7 Components

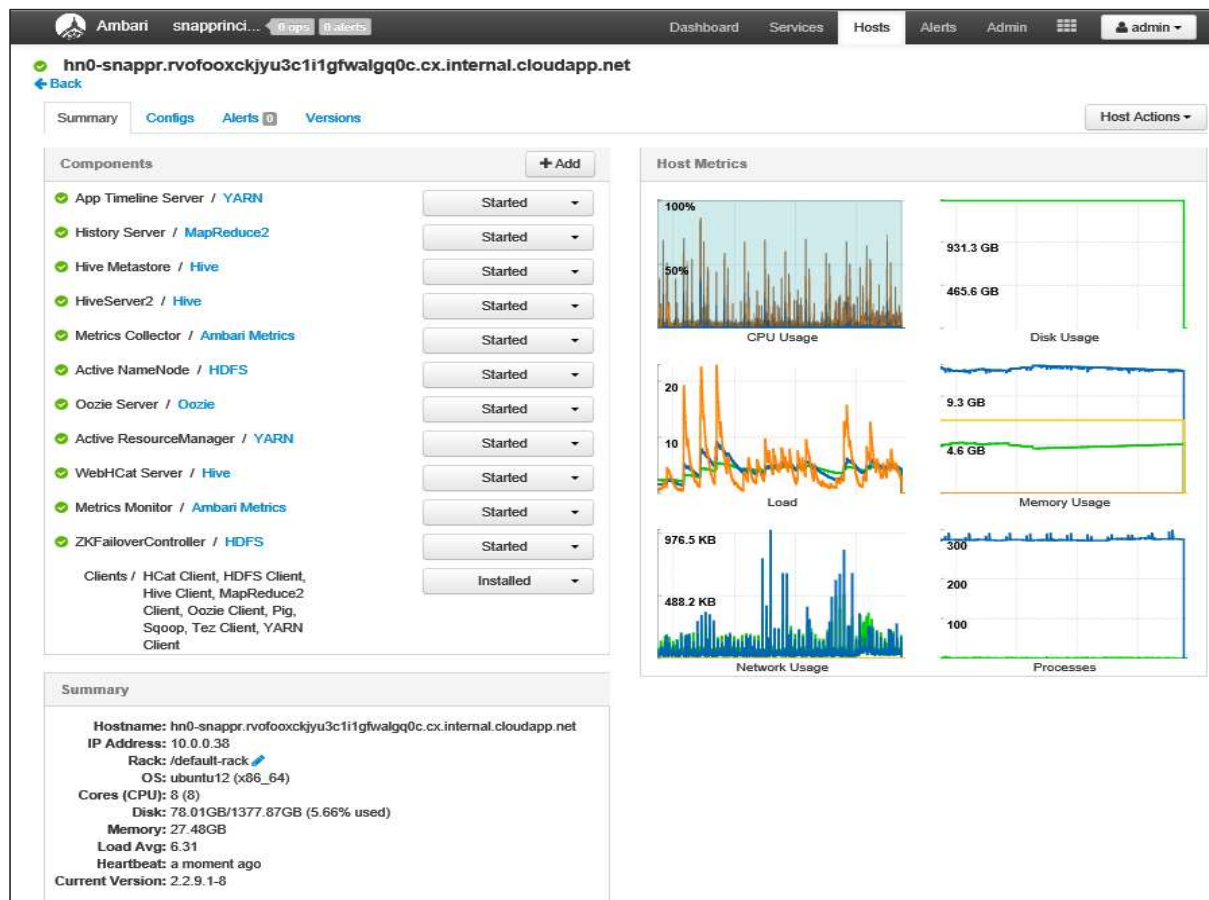
Amber Web UI: Hosts Drilldown

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.



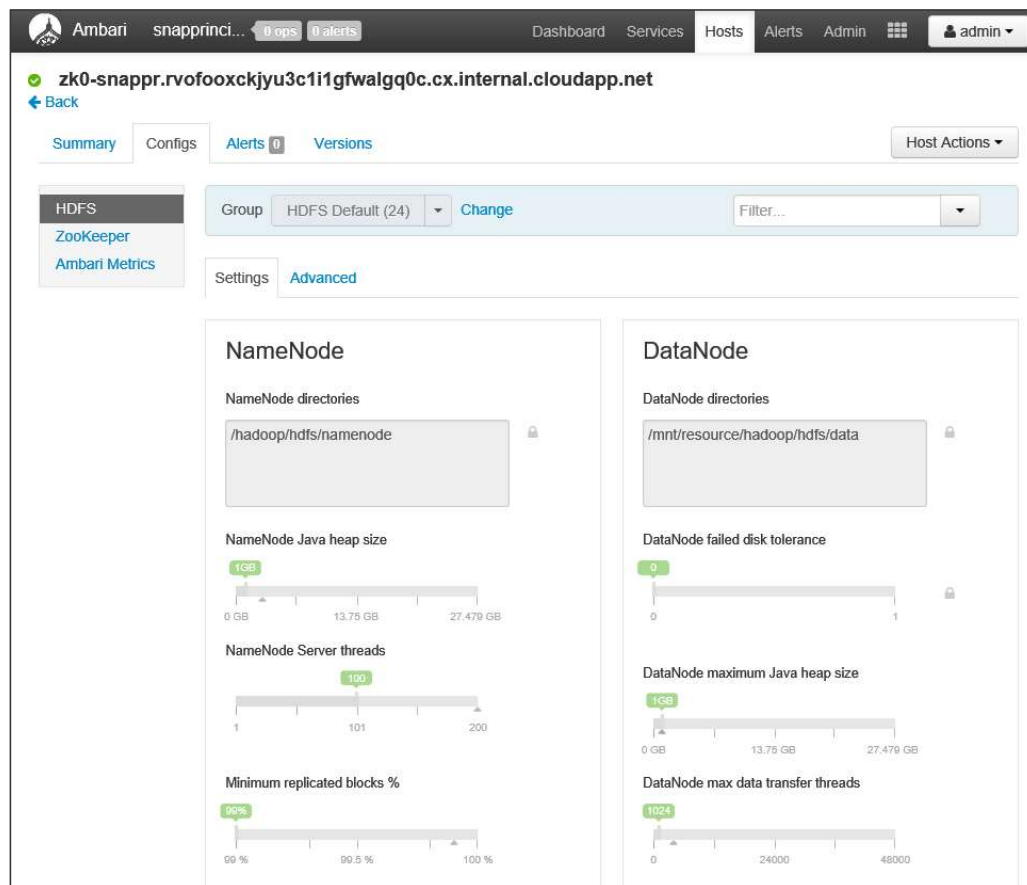
Ambari Web UI: Host Component Drilldown

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.

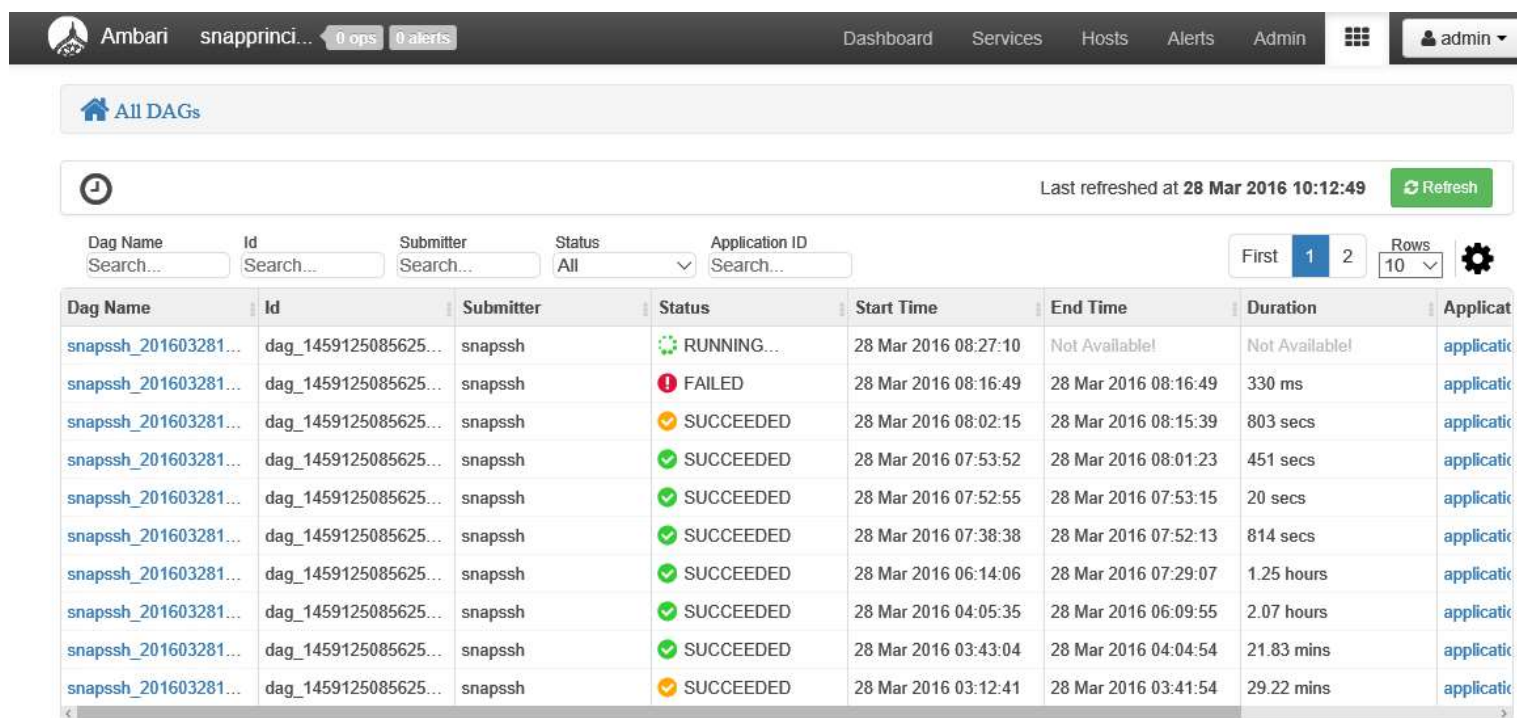
The screenshot shows the Ambari web interface for the Capacity Scheduler. The top navigation bar includes 'Ambari', 'snapprinci...', '0 ops', '0 alerts', and a user menu for 'admin'. The main content area is divided into several sections:

- Queue List:** A table showing the distribution of resources across queues: 'root (100%)', 'default (95%)', and 'joblauncher (5%)'. The 'default' queue is selected and highlighted in blue.
- Scheduler:** A section with a green checkmark indicating the scheduler is running. It includes fields for 'Maximum Applications' (10000), 'Maximum AM Resource' (33 %), and 'Node Locality Delay' (0).
- Calculator:** A field showing the resource calculator used: 'org.apache.hadoop.yarn.util.resourc'.
- Queue Mappings:** A section for defining queue mappings, currently empty.
- Queue Mappings Override:** A checkbox labeled 'Disabled'.
- Versions:** A section showing the current version of the scheduler, with buttons for 'v1', 'Current', 'INITIAL', and 'load'.
- Capacity:** A section for the 'default' queue, showing a 'Level Total' of 100%. It includes sliders for 'Capacity' (95 %) and 'Max Capacity' (100 %).
- Access Control and Status:** A section showing the 'State' as 'Running' and 'Stopped'. It also includes controls for 'Administer Queue' and 'Submit Applications', both set to 'Anyone'.
- Resources:** A section for resource configuration, including 'User Limit Factor' (10), 'Minimum User Limit' (100 %), 'Maximum Applications' (Inherited), 'Maximum AM Resource' (Inhe %), and 'Ordering policy'.

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



The screenshot shows the Ambari Tez View interface. At the top, there's a navigation bar with 'Ambari', 'snapprinci...', '0 ops', and '0 alerts'. Below this is a 'All DAGs' section with a search bar and a 'Last refreshed at 28 Mar 2016 10:12:49' timestamp. The main table lists DAGs with columns: Dag Name, Id, Submitter, Status, Start Time, End Time, Duration, and Application ID. The table shows several DAGs with statuses like RUNNING, FAILED, and SUCCEEDED.

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!	applicati...
snapssh_201603281...	dag_1459125085625...	snapssh	FAILED	28 Mar 2016 08:16:49	28 Mar 2016 08:16:49	330 ms	applicati...
snapssh_201603281...	dag_1459125085625...	snapssh	SUCCEEDED	28 Mar 2016 08:02:15	28 Mar 2016 08:15:39	803 secs	applicati...
snapssh_201603281...	dag_1459125085625...	snapssh	SUCCEEDED	28 Mar 2016 07:53:52	28 Mar 2016 08:01:23	451 secs	applicati...
snapssh_201603281...	dag_1459125085625...	snapssh	SUCCEEDED	28 Mar 2016 07:52:55	28 Mar 2016 07:53:15	20 secs	applicati...
snapssh_201603281...	dag_1459125085625...	snapssh	SUCCEEDED	28 Mar 2016 07:38:38	28 Mar 2016 07:52:13	814 secs	applicati...
snapssh_201603281...	dag_1459125085625...	snapssh	SUCCEEDED	28 Mar 2016 06:14:06	28 Mar 2016 07:29:07	1.25 hours	applicati...
snapssh_201603281...	dag_1459125085625...	snapssh	SUCCEEDED	28 Mar 2016 04:05:35	28 Mar 2016 06:09:55	2.07 hours	applicati...
snapssh_201603281...	dag_1459125085625...	snapssh	SUCCEEDED	28 Mar 2016 03:43:04	28 Mar 2016 04:04:54	21.83 mins	applicati...
snapssh_201603281...	dag_1459125085625...	snapssh	SUCCEEDED	28 Mar 2016 03:12:41	28 Mar 2016 03:41:54	29.22 mins	applicati...

Ambari: Tez View (DAG Details)

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

The screenshot shows the Ambari Tez View interface. The top navigation bar includes 'Ambari', 'snapprinci...', '0 ops', and '0 alerts'. The main header shows 'Dashboard', 'Services', 'Hosts', 'Alerts', and 'Admin'. The breadcrumb trail is 'All DAGs / DAG [snapssh_20160328150202_0b6115a6-c3f8-44bb-998d-6e30fa54b7c4:1]'. The 'Graphical View' tab is selected, with other tabs being 'DAG Details', 'DAG Counters', 'All Vertices', 'All Tasks', and 'All TaskAttempts'. A clock icon and 'Last refreshed at 28 Mar 2016 10:09:42' are shown, along with a 'Refresh' button. The DAG flow consists of four vertices: 'lineitem' (green), '929 1' (blue), 'P 169 cer 2' (blue), and 'out_Reducer..' (red). A tooltip for 'Reducer 2' is displayed over the 'P 169 cer 2' vertex.

Reducer 2	
Vertex Name	Reducer 2
Vertex ID	vertex_1459125085625_0024_1_01
Progress	1
Start Time	28 Mar 2016 08:02:17
End Time	28 Mar 2016 08:15:39
Duration	801 secs
First Task Start Time	28 Mar 2016 08:10:17
Tasks	169
Processor Class	ReduceTezProcessor



Get started today!

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