VMware vSphere Big Data Extensions Command-Line Interface Guide

vSphere Big Data Extensions 2.0

This document supports the version of each product listed and supports all subsequent versions until the document is replaced by a new edition. To check for more recent editions of this document, see http://www.vmware.com/support/pubs.

EN-001513-00

vmware®

You can find the most up-to-date technical documentation on the VMware Web site at:

http://www.vmware.com/support/

The VMware Web site also provides the latest product updates.

If you have comments about this documentation, submit your feedback to:

docfeedback@vmware.com

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

About This Book

VMware vSphere Big Data Extensions Command-Line Interface Guide describes how to use the Serengeti Command-Line Interface (CLI) to manage the vSphere resources that you use to create Hadoop and HBase clusters, and how to create, manage, and monitor Hadoop and HBase clusters with the Serengeti CLI.

VMware vSphere Big Data Extensions Command-Line Interface Guide also describes how to perform Hadoop and HBase operations with the Serengeti CLI, and provides cluster specification and Serengeti CLI command references.

Intended Audience

This guide is for system administrators and developers who want to use Serengeti to deploy and manage Hadoop clusters. To successfully work with Serengeti, you should be familiar with Hadoop and VMware® vSphere®.

VMware Technical Publications Glossary

VMware Technical Publications provides a glossary of terms that might be unfamiliar to you. For definitions of terms as they are used in VMware technical documentation, go to http://www.vmware.com/support/pubs.

Using the Serengeti Remote Command-Line Interface Client 1

The Serengeti Remote Command-Line Interface Client lets you access the Serengeti Management Server to deploy, manage, and use Hadoop.

Access the Serengeti CLI By Using the Remote Command-Line Interface Client

You can access the Serengeti Command-Line Interface (CLI) to perform Serengeti administrative tasks with the Serengeti Remote CLI Client.

IMPORTANT You can only run Hadoop commands from the Serengeti CLI on a cluster running the Apache Hadoop 1.2.1 distribution. To use the command-line to run Hadoop administrative commands for clusters running other Hadoop distributions, such as `cfg`, `fs`, `mr`, `pig`, and `hive`, use a Hadoop client node to run these commands.

Prerequisites

- Use the vSphere Web Client to log in to the vCenter Server on which you deployed the Serengeti vApp.
- Verify that the Serengeti vApp deployment was successful and that the Management Server is running.
- Verify that you have the correct password to log in to Serengeti CLI. See the *VMware vSphere Big Data Extensions Administrator's and User's Guide*.

  The Serengeti CLI uses its vCenter Server credentials.
- Verify that the Java Runtime Environment (JRE) is installed in your environment and that its location is in your `PATH` environment variable.

Procedure

1 Open a Web browser to connect to the Serengeti Management Server `cli` directory.

  `http://ip_address/cli`

2 Download the ZIP file for your version and build.

  The filename is in the format `VMware-Serengeti-cli-version_number-build_number.ZIP`.

3 Unzip the download.

  The download includes the following components.

  - The `serengeti-cli-version_number` JAR file, which includes the Serengeti Remote CLI Client.
  - The `samples` directory, which includes sample cluster configurations.
  - Libraries in the `lib` directory.

4 Open a command shell, and change to the directory where you unzipped the package.

5 Change to the `cli` directory, and run the following command to enter the Serengeti CLI.

- For any language other than French of German, run the following command.

  ```
  java –jar serengeti–cli–version_number.jar
  ```

- For French or German languages, which use code page 850 (CP 850) language encoding when running the Serengeti CLI from a Windows command console, run the following command.

  ```
  java –Dfile.encoding=cp850 –jar serengeti–cli–version_number.jar
  ```

6 Connect to the Serengeti service.

  You must run the `connect host` command every time you begin a CLI session, and again after the 30 minute session timeout. If you do not run this command, you cannot run any other commands.

  a Run the `connect` command.

    ```
    connect ––host xx.xx.xx.xx:8443
    ```

  b At the prompt, type your user name, which might be different from your login credentials for the Serengeti Management Server.

    NOTE If you do not create a user name and password for the Serengeti Command-Line Interface Client, you can use the default vCenter Server administrator credentials. The Serengeti Command-Line Interface Client uses the vCenter Server login credentials with read permissions on the Serengeti Management Server.

  c At the prompt, type your password.

A command shell opens, and the Serengeti CLI prompt appears. You can use the `help` command to get help with Serengeti commands and command syntax.

- To display a list of available commands, type `help`.
- To get help for a specific command, append the name of the command to the `help` command.

  ```
  help cluster create
  ```

- Press Tab to complete a command.

Managing vSphere Resources for Hadoop and HBase Clusters

2

Big Data Extensions lets you manage the resource pools, datastores, and networks that you use in the Hadoop and HBase clusters that you create.

Add a Resource Pool with the Serengeti Command-Line Interface

You add resource pools to make them available for use by Hadoop clusters. Resource pools must be located at the top level of a cluster. Nested resource pools are not supported.

When you add a resource pool to Big Data Extensions it symbolically represents the actual vSphere resource pool as recognized by vCenter Server. This symbolic representation lets you use the Big Data Extensions resource pool name, instead of the full path of the resource pool in vCenter Server, in cluster specification files.

Prerequisites

Deploy Big Data Extensions.

Procedure

1 Access the Serengeti Command-Line Interface client.

2 Run the `resourcepool add` command.

The `--vcrp` parameter is optional.

This example adds a Serengeti resource pool named `myRP` to the vSphere rp1 resource pool that is contained by the `cluster1` vSphere cluster.

```
resourcepool add --name myRP --vccluster cluster1 --vcrp rp1
```

What to do next

After you add a resource pool to Big Data Extensions, do not rename the resource pool in vSphere. If you rename it, you cannot perform Serengeti operations on clusters that use that resource pool.

Remove a Resource Pool with the Serengeti Command-Line Interface

You can remove resource pools from Serengeti that are not in use by a Hadoop cluster. You remove resource pools when you do not need them or if you want the Hadoop clusters you create in the Serengeti Management Server to be deployed under a different resource pool. Removing a resource pool removes its reference in vSphere. The resource pool is not deleted.

Procedure

1 Access the Serengeti Command-Line Interface client.

2 Run the `resourcepool delete` command.

If the command fails because the resource pool is referenced by a Hadoop cluster, you can use the `resourcepool list` command to see which cluster is referencing the resource pool.

This example deletes the resource pool named `myRP`.

```
resourcepool delete --name myRP
```

Add a Datastore with the Serengeti Command-Line Interface

You can add shared and local datastores to the Serengeti server to make them available to Hadoop clusters.

Procedure

1 Access the Serengeti CLI.

2 Run the `datastore add` command.

This example adds a new, local storage datastore named `myLocalDS`. The `--spec` parameter's value, `local*`, is a wildcard specifying a set of vSphere datastores. All vSphere datastores whose names begin with "local" are added and managed as a whole by Serengeti.

```
datastore add --name myLocalDS --spec local* --type LOCAL
```

What to do next

After you add a datastore to Big Data Extensions, do not rename the datastore in vSphere. If you rename it, you cannot perform Serengeti operations on clusters that use that datastore.

Remove a Datastore with the Serengeti Command-Line Interface

You can remove any datastore from Serengeti that is not referenced by any Hadoop clusters. Removing a datastore removes only the reference to the vCenter Server datastore. The datastore itself is not deleted.

You remove datastores if you do not need them or if you want to deploy the Hadoop clusters that you create in the Serengeti Management Server under a different datastore.

Procedure

1 Access the Serengeti CLI.

2 Run the `datastore delete` command.

If the command fails because the datastore is referenced by a Hadoop cluster, you can use the `datastore list` command to see which cluster is referencing the datastore.

This example deletes the `myDS` datastore.

```
datastore delete --name myDS
```

Add a Network with the Serengeti Command-Line Interface

You add networks to Serengeti to make their IP addresses available to Hadoop clusters. A network is a port group, as well as a means of accessing the port group through an IP address.

Prerequisites

If your network uses static IP addresses, be sure that the addresses are not occupied before you add the network.

Procedure

1 Access the Serengeti CLI.

2 Run the `network add` command.

This example adds a network named `myNW` to the 10PG vSphere port group. Virtual machines that use this network use DHCP to obtain the IP addresses.

```
network add --name myNW --portGroup 10PG --dhcp
```

This example adds a network named `myNW` to the 10PG vSphere port group. Hadoop nodes use addresses in the 192.168.1.2-100 IP address range, the DNS server IP address is 10.111.90.2, the gateway address is 192.168.1.1, and the subnet mask is 255.255.255.0.

```
network add --name myNW --portGroup 10PG --ip 192.168.1.2-100 --dns 10.111.90.2 --gateway
192.168.1.1 --mask 255.255.255.0
```

To specify multiple IP address segments, use multiple strings to express the IP address range in the format `xx.xx.xx.xx-xx[,xx]*`. For example:

```
xx.xx.xx.xx-xx, xx.xx.xx.xx-xx, single_ip, single_ip
```

What to do next

After you add a network to Big Data Extensions, do not rename it in vSphere. If you rename the network, you cannot perform Serengeti operations on clusters that use that network.

Reconfigure a Static IP Network with the Serengeti Command-Line Interface

You can reconfigure a Serengeti static IP network by adding IP address segments to it. You might need to add IP address segments so that there is enough capacity for a cluster that you want to create.

If the IP range that you specify includes IP addresses that are already in the network, Serengeti ignores the duplicated addresses. The remaining addresses in the specified range are added to the network. If the network is already used by a cluster, the cluster can use the new IP addresses after you add them to the network. If only part of the IP range is used by a cluster, the unused IP address can be used when you create a new cluster.

Prerequisites

If your network uses static IP addresses, be sure that the addresses are not occupied before you add the network.

Procedure

1 Access the Serengeti CLI.

2 Run the `network modify` command.

This example adds IP addresses from 192.168.1.2 to 192.168.1.100 to a network named myNetwork.

```
network modify --name myNetwork --addIP 192.168.1.2-100
```

Remove a Network with the Serengeti Command-Line Interface

You can remove networks from Serengeti that are not referenced by any Hadoop clusters. Removing an unused network frees the IP addresses for reuse.

Procedure

1 Access the Serengeti CLI.

2 Run the `network delete` command.

```
network delete --name network_name
```

If the command fails because the network is referenced by a Hadoop cluster, you can use the `network list --detail` command to see which cluster is referencing the network.

Creating Hadoop and HBase Clusters 3

Big Data Extensions lets you create and deploy Hadoop and HBase clusters. A Hadoop or HBase cluster is a special type of computational cluster designed specifically for storing and analyzing large amounts of unstructured data in a distributed computing environment.

The resource requirements are different for clusters created with the Serengeti Command-Line Interface and the Big Data Extensions plug-in for the vSphere Web Client because the clusters use different default templates. The default clusters created through the Serengeti Command-Line Interface are targeted for Project Serengeti users and proof-of-concept applications, and are smaller than the Big Data Extensions plug-in templates, which are targeted for larger deployments for commercial use.

Additionally, some deployment configurations require more resources than other configurations. For example, if you create a Greenplum HD 1.2 cluster, you cannot use the SMALL size virtual machine. If you create a default MapR or Greenplum HD cluster through the Serengeti Command-Line Interface, at least 550GB of storage and 55GB of memory are recommended. For other Hadoop distributions, at least 350GB of storage and 35GB of memory are recommended.

CAUTION When you create a cluster with Big Data Extensions, Big Data Extensions disables the cluster's virtual machine automatic migration. Although this prevents vSphere from automatically migrating the virtual machines, it does not prevent you from inadvertently migrating cluster nodes to other hosts by using the vCenter Server user interface. Do not use the vCenter Server user interface to migrate clusters. Performing such management functions outside of the Big Data Extensions environment can make it impossible for you to perform some Big Data Extensions operations, such as disk failure recovery.

- Create a Data-Compute Separated Cluster with Topology Awareness and Placement Constraints on page 34

  You can create clusters with separate data and compute nodes, and define topology and placement policy constraints to distribute the nodes among the physical racks and the virtual machines.

- Serengeti's Default HBase Cluster Configuration on page 36

  HBase clusters are required for you to build big table applications. To run HBase MapReduce jobs, configure the HBase cluster to include JobTracker nodes or TaskTracker nodes.

- Create a Default HBase Cluster with the Serengeti Command-Line Interface on page 36

  Serengeti supports deploying HBase clusters on HDFS.

- Create an HBase Cluster with vSphere HA Protection with the Serengeti Command-Line Interface on page 37

  You can create HBase clusters with separated Hadoop NameNode and HBase Master roles, and configure vSphere HA protection for the Masters.

About Hadoop and HBase Cluster Deployment Types

Big Data Extensions lets you deploy several types of Hadoop and HBase clusters. You need to know about the types of clusters that you can create.

You can create the following types of clusters.

Basic Hadoop Cluster	You can create a simple Hadoop deployment for proof of concept projects and other small scale data processing tasks using the basic Hadoop cluster.
HBase Cluster	You can create an HBase cluster. To run HBase MapReduce jobs, configure the HBase cluster to include JobTracker or TaskTracker nodes.
Data-Compute Separated Hadoop Cluster	You can separate the data and compute nodes in a Hadoop cluster, and you can control how nodes are placed on your environment's vSphere ESXi hosts.
Compute-Only Hadoop Cluster	You can create a compute-only cluster to run MapReduce jobs. Compute-only clusters run only MapReduce services that read data from external HDFS clusters and that do not need to store data.
Customized Cluster	You can use an existing cluster specification file to create clusters using the same configuration as your previously created clusters. You can also edit the file to customize the cluster configuration.

Hadoop Distributions Supporting MapReduce v1 and MapReduce v2 (YARN)

If the Hadoop distribution you use supports both MapReduce v1 and MapReduce v2 (YARN), the default Hadoop cluster configuration creates a MapReduce v2 cluster.

In addition, if you are using two different versions of a vendor's Hadoop distribution, and both versions support MapReduce v1 and MapReduce v2, the cluster you create using the latest version with the default Hadoop cluster will use MapReduce v2. Clusters you create with the earlier Hadoop version will use MapReduce v1. For example, if you have both Cloudera CDH 5 and CDH 4 installed within Big Data Extensions, clusters you create with CDH 5 will use MapReduce v2, and clusters you create with CDH 4 will use MapReduce v1.

Serengeti's Default Hadoop Cluster Configuration

For basic Hadoop deployments, such as proof of concept projects, you can use Serengeti's default Hadoop cluster configuration for clusters that are created with the Command-Line Interface.

The resulting cluster deployment consists of the following nodes and virtual machines:

- One master node virtual machine with NameNode and JobTracker services.
- Three worker node virtual machines, each with DataNode and TaskTracker services.
- One client node virtual machine containing the Hadoop client environment: the Hadoop client shell, Pig, and Hive.

Hadoop Distributions Supporting MapReduce v1 and MapReduce v2 (YARN)

If the Hadoop distribution you use supports both MapReduce v1 and MapReduce v2 (YARN), the default Hadoop cluster configuration creates a MapReduce v2 cluster.

In addition, if you are using two different versions of a vendor's Hadoop distribution, and both versions support MapReduce v1 and MapReduce v2, the cluster you create using the latest version with the default Hadoop cluster will use MapReduce v2. Clusters you create with the earlier Hadoop version will use MapReduce v1. For example, if you have both Cloudera CDH 5 and CDH 4 installed within Big Data Extensions, clusters you create with CDH 5 will use MapReduce v2, and clusters you create with CDH 4 will use MapReduce v1.

Create a Default Serengeti Hadoop Cluster with the Serengeti Command-Line Interface

You can create as many clusters as you want in your Serengeti environment, but your environment must meet all prerequisites.

Prerequisites

- Deploy the Serengeti vApp.
- Ensure that you have adequate resources allocated to run the Hadoop cluster.
- To use any Hadoop distribution other than the provided Apache Hadoop, add one or more Hadoop distributions. See the *VMware vSphere Big Data Extensions Administrator's and User's Guide*.

Procedure

1 Access the Serengeti CLI.

2 Deploy a default Serengeti Hadoop cluster on vSphere.

```
cluster create --name cluster_name
```

The only valid characters for cluster names are alphanumeric and underscores. When you choose the cluster name, also consider the applicable vApp name. Together, the vApp and cluster names must be < 80 characters.

During the deployment process, real-time progress updates appear on the command-line.

What to do next

After the deployment finishes, you can run Hadoop commands and view the IP addresses of the Hadoop node virtual machines from the Serengeti CLI.

Create a Cluster with a Custom Administrator Password with the Serengeti Command-Line Interface

When you create a cluster, you can assign a custom administrator password to all the nodes in the cluster. Custom administrator passwords let you directly log in to the cluster's nodes instead of having to first log in to the Serengeti Management server.

Prerequisites

- Deploy the Serengeti vApp.
- Ensure that you have adequate resources allocated to run the Hadoop cluster.
- To use any Hadoop distribution other than the provided Apache Hadoop, add one or more Hadoop distributions. See the *VMware vSphere Big Data Extensions Administrator's and User's Guide*.

Procedure

1 Access the Serengeti CLI.

2 Run the `cluster create` command and include the `--password` parameter.

```
cluster create --name cluster_name --password
```

3 Enter your custom password, and enter it again.

Passwords are from 8 to 128 characters, and include only alphanumeric characters ([0-9, a-z, A-Z]) and the following special characters: _ @ # $ % ^ & *

Your custom password is assigned to all the nodes in the cluster.

Create a Cluster with an Available Distribution with the Serengeti Command-Line Interface

You can choose which Hadoop distribution to use when you deploy a cluster. If you do not specify a Hadoop distribution, the resulting cluster includes the default distribution, Apache Hadoop.

Prerequisites

- Deploy the Serengeti vApp.
- Ensure that you have adequate resources allocated to run the Hadoop cluster.
- To use any Hadoop distribution other than the provided Apache Hadoop, add one or more Hadoop distributions. See the *VMware vSphere Big Data Extensions Administrator's and User's Guide*.

Procedure

1 Access the Serengeti CLI.

2 Run the `cluster create` command, and include the `--distro` parameter.

The `--distro` parameter's value must match a distribution name displayed by the `distro list` command.

NOTE To create an Apache Bigtop, Cloudera CDH4 and CDH5, or Pivotal PHD 1.1 cluster, you must configure a valid DNS and FQDN for the cluster's HDFS and MapReduce traffic. Without valid DNS and FQDN settings, the cluster creation process might fail or the cluster is created but does not function.

This example deploys a cluster with the Cloudera CDH distribution:

```
cluster create --name clusterName --distro cdh
```

This example creates a customized cluster named `mycdh` that uses the CDH4 Hadoop distribution, and is configured according to the `/opt/serengeti/samples/default_cdh4_ha_and_federation_hadoop_cluster.json` sample cluster specification file. In this sample file, `nameservice0` and `nameservice1` are federated. That is, `nameservice0` and `nameservice1` are independent and do not require coordination with each other. The NameNode nodes in the `nameservice0` node group are HDFS2 HA enabled. In Serengeti, name node group names are used as service names for HDFS2.

```
cluster create --name mycdh --distro cdh4 --
specFile /opt/serengeti/samples/default_cdh4_ha_hadoop_cluster.json
```

Create a Hadoop Cluster with Assigned Resources with the Serengeti Command-Line Interface

By default, when you use Serengeti to deploy a Hadoop cluster, the cluster might contain any or all available resources: vCenter Server resource pool for the virtual machine's CPU and memory, datastores for the virtual machine's storage, and a network. You can assign which resources the cluster uses by specifying specific resource pools, datastores, and/or a network when you create the Hadoop cluster.

Prerequisites

- Deploy the Serengeti vApp.
- Ensure that you have adequate resources allocated to run the Hadoop cluster.
- To use any Hadoop distribution other than the provided Apache Hadoop, add one or more Hadoop distributions. See the *VMware vSphere Big Data Extensions Administrator's and User's Guide*.

Procedure

1 Access the Serengeti CLI.

2 Run the `cluster create` command, and specify any or all of the command's resource parameters.

This example deploys a cluster named `myHadoop` on the `myDS` datastore, under the `myRP` resource pool, and uses the `myNW` network for virtual machine communications.

```
cluster create --name myHadoop --rpNames myRP --dsNames myDS --networkName myNW
```

Create a Cluster with Multiple Networks with the Serengeti Command-Line Interface

When you create a cluster, you can distribute the management, HDFS, and MapReduce traffic to separate networks. You might want to use separate networks to improve performance or to isolate traffic for security reasons.

For optimal performance, use the same network for HDFS and MapReduce traffic in Hadoop and Hadoop +HBase clusters. HBase clusters use the HDFS network for traffic related to the HBase Master and HBase RegionServer services.

IMPORTANT You cannot configure multiple networks for clusters that use the MapR Hadoop distribution.

Prerequisites

- Deploy the Serengeti vApp.
- Ensure that you have adequate resources allocated to run the Hadoop cluster.
- To use any Hadoop distribution other than the provided Apache Hadoop, add one or more Hadoop distributions. See the *VMware vSphere Big Data Extensions Administrator's and User's Guide*.

Procedure

1 Access the Serengeti CLI.

2 Run the `cluster create` command and include the `--networkName`, `--hdfsNetworkName`, and `--mapredNetworkName` parameters.

```
cluster create --name cluster_name --networkName management_network [--hdfsNetworkName hdfs_network] [--mapredNetworkName mapred_network]
```

If you omit an optional network parameter, the traffic associated with that network parameter is routed on the management network that you specify by the `--networkName` parameter.

NOTE To create an Apache Bigtop, Cloudera CDH4 and CDH5, or Pivotal PHD 1.1 cluster, you must configure a valid DNS and FQDN for the cluster's HDFS and MapReduce traffic. Without valid DNS and FQDN settings, the cluster creation process might fail or the cluster is created but does not function.

The cluster's management, HDFS, and MapReduce traffic is distributed among the specified networks.

Create a MapReduce v2 (YARN) Cluster with the Serengeti Command-Line Interface

You can create MapReduce v2 (YARN) cluster with the Serengeti Command-Line Interface.

When you create a Hadoop cluster with the Serengeti Command-Line Interface, by default you create a MapReduce v1 cluster. To create a MapReduce v2 (YARN) cluster, create a cluster specification file modeled after the `/opt/serengeti/samples/default_hadoop_yarn_cluster.json` file, and specify the `--specFile` parameter and your cluster specification file in the `cluster create ...` command.

Prerequisites

- Deploy the Serengeti vApp.
- Ensure that you have adequate resources allocated to run the Hadoop cluster.
- To use any Hadoop distribution other than the provided Apache Hadoop, add one or more Hadoop distributions. See the *VMware vSphere Big Data Extensions Administrator's and User's Guide*.

Procedure

1 Access the Serengeti CLI.

2 Run the `cluster create ...` command.

This example creates a customized MapR v2 cluster according to the sample cluster specification file, `default_hadoop_yarn_cluster.json`.

```
cluster create --name cluster_name --distro cdh4 --
specFile /opt/serengeti/samples/default_hadoop_yarn_cluster.json
```

Create a Customized Hadoop or HBase Cluster with the Serengeti Command-Line Interface

You can create clusters that are customized for your requirements, including the number of nodes, virtual machine RAM and disk size, the number of CPUs, and so on.

The Serengeti package includes several annotated sample cluster specification files that you can use as models when you create your custom specification files.

- In the Serengeti Management Server, the sample cluster specification files are in `/opt/serengeti/samples`.
- If you use the Serengeti Remote CLI client, the sample specification files are in the client directory.

Changing a node group role might cause the cluster creation process to fail. For example, workable clusters require a NameNode, so if there are no NameNode nodes after you change node group roles, you cannot create a cluster.

Prerequisites

- Deploy the Serengeti vApp.
- Ensure that you have adequate resources allocated to run the Hadoop cluster.
- To use any Hadoop distribution other than the provided Apache Hadoop, add one or more Hadoop distributions. See the *VMware vSphere Big Data Extensions Administrator's and User's Guide*.

Procedure

1 Create a cluster specification file to define the cluster's characteristics such as the node groups.

2 Access the Serengeti CLI.

3 Run the `cluster create` command, and specify the cluster specification file.

Use the full path to specify the file.

```
cluster create --name cluster_name --specFile full_path/spec_filename
```

NOTE To create an Apache Bigtop, Cloudera CDH4 and CDH5, or Pivotal PHD 1.1 cluster, you must configure a valid DNS and FQDN for the cluster's HDFS and MapReduce traffic. Without valid DNS and FQDN settings, the cluster creation process might fail or the cluster is created but does not function.

Create a Hadoop Cluster with Any Number of Master, Worker, and Client Nodes

You can create a Hadoop cluster with any number of master, worker, and client nodes.

Prerequisites

- Deploy the Serengeti vApp.
- Ensure that you have adequate resources allocated to run the Hadoop cluster.
- To use any Hadoop distribution other than the provided Apache Hadoop, add one or more Hadoop distributions. See the *VMware vSphere Big Data Extensions Administrator's and User's Guide*.

Procedure

1 Create a cluster specification file to define the cluster's characteristics, including the node groups.

NOTE To create an Apache Bigtop, Cloudera CDH4 and CDH5, or Pivotal PHD 1.1 cluster, you must configure a valid DNS and FQDN for the cluster's HDFS and MapReduce traffic. Without valid DNS and FQDN settings, the cluster creation process might fail or the cluster is created but does not function.

In this example, the cluster has one master MEDIUM size virtual machine, five worker SMALL size virtual machines, and one client SMALL size virtual machine. The `instanceNum` attribute configures the number of virtual machines in a node.

```
{
  "nodeGroups" : [
    {
      "name": "master",
      "roles": [
        "hadoop_namenode",
        "hadoop_jobtracker"
      ],
      "instanceNum": 1,
      "instanceType": "MEDIUM"
    },
    {
      "name": "worker",
      "roles": [
        "hadoop_datanode",
        "hadoop_tasktracker"
      ],
      "instanceNum": 5,
      "instanceType": "SMALL"
    },
    {
      "name": "client",
      "roles": [
        "hadoop_client",
        "hive",
        "hive_server",
        "pig"
      ],
      "instanceNum": 1,
```

```
      "instanceType": "SMALL"
    }
  ]
}
```

2 Access the Serengeti CLI.

3 Run the `cluster create` command, and specify the cluster specification file.

```
cluster create --name cluster_name --specFile full_path/spec_filename
```

Create a Data-Compute Separated Cluster with No Node Placement Constraints

You can create a cluster with separate data and compute nodes, without node placement constraints.

Prerequisites

- Deploy the Serengeti vApp.
- Ensure that you have adequate resources allocated to run the Hadoop cluster.
- To use any Hadoop distribution other than the provided Apache Hadoop, add one or more Hadoop distributions. See the *VMware vSphere Big Data Extensions Administrator's and User's Guide*.

Procedure

1 Create a cluster specification file to define the cluster's characteristics.

NOTE To create an Apache Bigtop, Cloudera CDH4 and CDH5, or Pivotal PHD 1.1 cluster, you must configure a valid DNS and FQDN for the cluster's HDFS and MapReduce traffic. Without valid DNS and FQDN settings, the cluster creation process might fail or the cluster is created but does not function.

In this example, the cluster has separate data and compute nodes, without node placement constraints. Four data nodes and eight compute nodes are created and put into individual virtual machines. The number of nodes is configured by the `instanceNum` attribute.

```
{
  "nodeGroups":[
    {
      "name": "master",
      "roles": [
        "hadoop_namenode",
        "hadoop_jobtracker"
      ],
      "instanceNum": 1,
      "cpuNum": 2,
      "memCapacityMB": 7500,
    },
    {
      "name": "data",
      "roles": [
        "hadoop_datanode"
      ],
      "instanceNum": 4,
      "cpuNum": 1,
      "memCapacityMB": 3748,
      "storage": {
```

```
        "type": "LOCAL",
        "sizeGB": 50
      }
    },
    {
      "name": "compute",
      "roles": [
        "hadoop_tasktracker"
      ],
      "instanceNum": 8,
      "cpuNum": 2,
      "memCapacityMB": 7500,
      "storage": {
        "type": "LOCAL",
        "sizeGB": 20
      }
    },
    {
      "name": "client",
      "roles": [
        "hadoop_client",
        "hive",
        "pig"
      ],
      "instanceNum": 1,
      "cpuNum": 1,
      "storage": {
        "type": "LOCAL",
        "sizeGB": 50
      }
    }
  ],
  "configuration": {
  }
}
```

2 Access the Serengeti CLI.

3 Run the `cluster create` command and specify the cluster specification file.

```
cluster create --name cluster_name --specFile full_path/spec_filename
```

Create a Data-Compute Separated Cluster with Placement Policy Constraints

You can create a cluster with separate data and compute nodes, and define placement policy constraints to distribute the nodes among the virtual machines as you want.

CAUTION When you create a cluster with Big Data Extensions, Big Data Extensions disables the cluster's virtual machine automatic migration. Although this prevents vSphere from migrating the virtual machines, it does not prevent you from inadvertently migrating cluster nodes to other hosts by using the vCenter Server user interface. Do not use the vCenter Server user interface to migrate clusters. Performing such management functions outside of the Big Data Extensions environment might break the cluster's placement policy, such as the number of instances per host and the group associations. Even if you do not specify a placement policy, using vCenter Server to migrate clusters can break the default ROUNDROBIN placement policy constraints.

Prerequisites

- Deploy the Serengeti vApp.
- Ensure that you have adequate resources allocated to run the Hadoop cluster.
- To use any Hadoop distribution other than the provided Apache Hadoop, add one or more Hadoop distributions. See the *VMware vSphere Big Data Extensions Administrator's and User's Guide*.

Procedure

1 Create a cluster specification file to define the cluster's characteristics, including the node groups and placement policy constraints.

NOTE To create an Apache Bigtop, Cloudera CDH4 and CDH5, or Pivotal PHD 1.1 cluster, you must configure a valid DNS and FQDN for the cluster's HDFS and MapReduce traffic. Without valid DNS and FQDN settings, the cluster creation process might fail or the cluster is created but does not function.

In this example, the cluster has data-compute separated nodes, and each node group has a `placementPolicy` constraint. After a successful provisioning, four data nodes and eight compute nodes are created and put into individual virtual machines. With the `instancePerHost=1` constraint, the four data nodes are placed on four ESXi hosts. The eight compute nodes are put onto four ESXi hosts: two nodes on each ESXi host.

This cluster specification requires that you configure datastores and resource pools for at least four hosts, and that there is sufficient disk space for Serengeti to perform the necessary placements during deployment.

```
{
  "nodeGroups":[
    {
      "name": "master",
      "roles": [
        "hadoop_namenode",
        "hadoop_jobtracker"
      ],
      "instanceNum": 1,
      "cpuNum": 2,
      "memCapacityMB": 7500,
    },
    {
      "name": "data",
      "roles": [
        "hadoop_datanode"
      ],
      "instanceNum": 4,
      "cpuNum": 1,
      "memCapacityMB": 3748,
      "storage": {
        "type": "LOCAL",
        "sizeGB": 50
      },
      "placementPolicies": {
        "instancePerHost": 1
      }
    },
    {
```

```
      "name": "compute",
      "roles": [
        "hadoop_tasktracker"
      ],
      "instanceNum": 8,
      "cpuNum": 2,
      "memCapacityMB": 7500,
      "storage": {
        "type": "LOCAL",
        "sizeGB": 20
      },
      "placementPolicies": {
        "instancePerHost": 2
      }
    },
    {
      "name": "client",
      "roles": [
        "hadoop_client",
        "hive",
        "pig"
      ],
      "instanceNum": 1,
      "cpuNum": 1,
      "storage": {
        "type": "LOCAL",
        "sizeGB": 50
      }
    }
  ],
  "configuration": {
  }
}
```

2 Access the Serengeti CLI.

3 Run the `cluster create` command, and specify the cluster specification file.

```
cluster create --name cluster_name --specFile full_path/spec_filename
```

Create a Compute-Only Cluster with the Serengeti Command-Line Interface

You can create compute-only clusters to run MapReduce jobs on existing HDFS clusters, including storage solutions that serve as an external HDFS.

Prerequisites

- Deploy the Serengeti vApp.
- Ensure that you have adequate resources allocated to run the Hadoop cluster.
- To use any Hadoop distribution other than the provided Apache Hadoop, add one or more Hadoop distributions. See the *VMware vSphere Big Data Extensions Administrator's and User's Guide*.

Procedure

1 Create a cluster specification file that is modeled on the Serengeti `compute_only_cluster.json` sample cluster specification file found in the Serengeti `cli/samples` directory.

2 Add the following code to your new cluster specification file.

For HDFS clusters, set `port_num` to **`8020`**. For Hadoop 2.0 clusters, such as CDH4 and Pivotal HD distributions, set `port_num` to **`9000`**.

NOTE To create an Apache Bigtop, Cloudera CDH4 and CDH5, or Pivotal PHD 1.1 cluster, you must configure a valid DNS and FQDN for the cluster's HDFS and MapReduce traffic. Without valid DNS and FQDN settings, the cluster creation process might fail or the cluster is created but does not function.

In this example, the `externalHDFS` field points to an HDFS. Assign the `hadoop_jobtracker` role to the `master` node group and the `hadoop_tasktracker` role to the `worker` node group.

The `externalHDFS` field conflicts with node groups that have `hadoop_namenode` and `hadoop_datanode` roles. This conflict might cause the cluster creation to fail or, if successfully created, the cluster might not work correctly. To avoid this problem, define only a single HDFS.

```
{
    "externalHDFS": "hdfs://hostname-of-namenode:port_num",
    "nodeGroups": [
        {
            "name": "master",
            "roles":  [
                "hadoop_jobtracker"
            ],
            "instanceNum": 1,
            "cpuNum":  2,
            "memCapacityMB":  7500,
        },
        {
            "name": "worker",
            "roles": [
                "hadoop_tasktracker",
            ],
            "instanceNum": 4,
            "cpuNum":  2,
            "memCapacityMB": 7500,
            "storage": {
                "type": "LOCAL",
                "sizeGB": 20
            },
        },
        {
            "name": "client",
            "roles":  [
                "hadoop_client",
                "hive",
                "pig"
            ],
            "instanceNum": 1,
            "cpuNum": 1,
            "storage": {
                "type": "LOCAL",
                "sizeGB": 50
            },
        }
```

```
],
“configuration” : {
}
}
```

3 Access the Serengeti CLI.

4 Run the `cluster create` command and include the cluster specification file parameter and associated filename.

```
cluster create --name name_computeOnlyCluster --specFile path/spec_file_name
```

Create a Basic Cluster with the Serengeti Command-Line Interface

You can create a basic cluster in your Serengeti environment. A basic cluster is a group of virtual machines provisioned and managed by Serengeti. Serengeti helps you to plan and provision the virtual machines to your specifications. You can use the basic cluster's virtual machines to install Big Data applications.

The basic cluster does not install the Big Data application packages used when creating a Hadoop or HBase cluster. Instead, you can install and manage Big Data applications with third party application management tools such as Apache Ambari or Cloudera Manager within your Big Data Extensions environment, and integrate it with your Hadoop software. The basic cluster does not deploy a Hadoop or Hbase cluster. You must deploy software into the basic cluster's virtual machines using an external third party application management tool.

The Serengeti package includes an annotated sample cluster specification file that you can use as an example when you create your basic cluster specification file. In the Serengeti Management Server, the sample specification file is located at `/opt/serengeti/samples/basic_cluster.json`. You can modify the configuration values in the sample cluster specification file to meet your requirements. The only value you cannot change is the value assigned to the role for each node group, which must always be `basic`.

You can deploy a basic cluster with the Big Data Extension plug-in using a customized cluster specification file.

To deploy software within the basic cluster virtual machines, use the `cluster list --detail` command, or run `serengeti-ssh.sh` *`cluster_name`* to obtain the IP address of the virtual machine. You can then use the IP address with management applications such as Apache Ambari or Cloudera Manager to provision the virtual machine with software of your choosing. You can configure the management application to use the user name serengeti, and the password you specified when creating the basic cluster within Big Data Extensions when the management tool needs a user name and password to connect to the virtual machines.

Prerequisites

- Deploy the Serengeti vApp.
- Ensure that you have adequate resources allocated to run the cluster, as well as the Big Data software you intend to deploy.

Procedure

1 Create a specification file to define the basic cluster's characteristics.

You must use the `basic` role for each node group you define for the basic cluster.

```
{
  "nodeGroups":[
    {
      "name": "master",
      "roles": [
      "basic"
      ],
      "instanceNum": 1,
```

```
      "cpuNum": 2,
      "memCapacityMB": 3768,
      "storage": {
        "type": "LOCAL",
        "sizeGB": 250
      },
      "haFlag": "on"
    },
    {
      "name": "worker",
      "roles": [
      "basic"
      ],
      "instanceNum": 1,
      "cpuNum": 2,
      "memCapacityMB": 3768,
      "storage": {
        "type": "LOCAL",
        "sizeGB": 250
      },
      "haFlag": "off"
    }
  ]
}
```

2 Access the Serengeti CLI.

3 Run the `cluster create` command, and specify the basic cluster specification file.

```
cluster create --name cluster_name --specFile /opt/serengeti/samples/basic_cluster.json --
password
```

NOTE When creating a basic cluster, you do not need to specify a Hadoop distribution type using the `--distro` option. The reason for this is that there is no Hadoop distribution being installed within the basic cluster to be managed by Serengeti.

About Cluster Topology

You can improve workload balance across your cluster nodes, and improve performance and throughput, by specifying how Hadoop virtual machines are placed using topology awareness. For example, you can have separate data and compute nodes, and improve performance and throughput by placing the nodes on the same set of physical hosts.

To get maximum performance out of your Hadoop or HBase cluster, configure your cluster so that it has awareness of the topology of your environment's host and network information. Hadoop performs better when it uses within-rack transfers, where more bandwidth is available, to off-rack transfers when assigning MapReduce tasks to nodes. HDFS can place replicas more intelligently to trade off performance and resilience. For example, if you have separate data and compute nodes, you can improve performance and throughput by placing the nodes on the same set of physical hosts.

CAUTION When you create a cluster with Big Data Extensions, Big Data Extensions disables the cluster's virtual machine automatic migration. Although this prevents vSphere from migrating the virtual machines, it does not prevent you from inadvertently migrating cluster nodes to other hosts by using the vCenter Server user interface. Do not use the vCenter Server user interface to migrate clusters. Performing such management functions outside of the Big Data Extensions environment might break the cluster's placement policy, such as the number of instances per host and the group associations. Even if you do not specify a placement policy, using vCenter Server to migrate clusters can break the default ROUNDROBIN placement policy constraints.

You can specify the following topology awareness configurations.

Hadoop Virtualization Extensions (HVE)

Enhanced cluster reliability and performance provided by refined Hadoop replica placement, task scheduling, and balancer policies. Hadoop clusters implemented on a virtualized infrastructure have full awareness of the topology on which they are running when using HVE.

To use HVE, your Hadoop distribution must support HVE and you must create and upload a topology rack-hosts mapping file.

RACK_AS_RACK

Standard topology for Apache Hadoop distributions. Only rack and host information are exposed to Hadoop. To use RACK_AS_RACK, create and upload a server topology file.

HOST_AS_RACK

Simplified topology for Apache Hadoop distributions. To avoid placing all HDFS data block replicas on the same physical host, each physical host is treated as a rack. Because data block replicas are never placed on a rack, this avoids the worst case scenario of a single host failure causing the complete loss of any data block.

Use HOST_AS_RACK if your cluster uses a single rack, or if you do not have rack information with which to decide about topology configuration options.

None

No topology is specified.

Topology Rack-Hosts Mapping File

Rack-Hosts mapping files are plain text files that associate logical racks with physical hosts. These files are required to create clusters with HVE or RACK_AS_RACK topology.

The format for every line in a topology rack-hosts mapping file is:

```
rackname: hostname1, hostname2 ...
```

For example, to assign physical hosts a.b.foo.com and a.c.foo.com to rack1, and physical host c.a.foo.com to rack2, include the following lines in your topology rack-hosts mapping file.

```
rack1: a.b.foo.com, a.c.foo.com
rack2: c.a.foo.com
```

Topology Placement Policy Definition Files

The `placementPolicies` field in the cluster specification file controls how nodes are placed in the cluster.

If you specify values for both `instancePerHost` and `groupRacks`, there must be a sufficient number of available hosts. To display the rack hosts information, use the `topology list` command.

The code shows an example placementPolicies field in a cluster specification file.

```
{
  "nodeGroups":[
    …
    {
      "name": "group_name",
       …
      "placementPolicies": {
        "instancePerHost": 2,
        "groupRacks": {
          "type": "ROUNDROBIN",
          "racks": ["rack1", "rack2", "rack3"]
         },
        "groupAssociations": [{
           "reference": "another_group_name",
           "type": "STRICT"   // or "WEAK"
         }]
      }
    },
    …
}
```

Table 3-1. placementPolicies Object Definition

JSON field	Type	Description
instancePerHost	Optional	Number of virtual machine nodes to place for each physical ESXi host. This constraint is aimed at balancing the workload.
groupRacks	Optional	Method of distributing virtual machine nodes among the cluster's physical racks. Specify the following JSON strings: ■ `type`. Specify ROUNDROBIN, which selects candidates fairly and without priority. ■ `racks`. Which racks in the topology map to use.
groupAssociations	Optional	One or more target node groups with which this node group associates. Specify the following JSON strings: ■ `reference`. Target node group name ■ `type`: ■ STRICT. Place the node group on the target group's set or subset of ESXi hosts. If STRICT placement is not possible, the operation fails. ■ WEAK. Attempt to place the node group on the target group's set or subset of ESXi hosts, but if that is not possible, use an extra ESXi host.

Create a Cluster with Topology Awareness with the Serengeti Command-Line Interface

To achieve a balanced workload or to improve performance and throughput, you can control how Hadoop virtual machines are placed by adding topology awareness to the Hadoop clusters. For example, you can have separate data and compute nodes, and improve performance and throughput by placing the nodes on the same set of physical hosts.

Prerequisites

- Deploy the Serengeti vApp.
- Ensure that you have adequate resources allocated to run the Hadoop cluster.
- To use any Hadoop distribution other than the provided Apache Hadoop, add one or more Hadoop distributions. See the *VMware vSphere Big Data Extensions Administrator's and User's Guide*.

Procedure

1 Access the Serengeti Command-Line Interface.

2 (Optional) Run the `topology list` command to view the list of available topologies.

```
topology list
```

3 (Optional) If you want the cluster to use HVE or RACK_AS_RACK toplogies, create a topology rack-hosts mapping file and upload the file to the Serengeti Management Server.

```
topology upload --fileName name_of_rack_hosts_mapping_file
```

4 Run the `cluster create` command to create the cluster.

```
cluster create --name cluster-name ... --topology {HVE|RACK_AS_RACK|HOST_AS_RACK}
```

NOTE To create an Apache Bigtop, Cloudera CDH4 and CDH5, or Pivotal PHD 1.1 cluster, you must configure a valid DNS and FQDN for the cluster's HDFS and MapReduce traffic. Without valid DNS and FQDN settings, the cluster creation process might fail or the cluster is created but does not function.

This example creates an HVE topology.

```
cluster create --name cluster-name --topology HVE --distro name_of_HVE-supported_distro
```

5 View the allocated nodes on each rack.

```
cluster list --name cluster-name --detail
```

Create a Data-Compute Separated Cluster with Topology Awareness and Placement Constraints

You can create clusters with separate data and compute nodes, and define topology and placement policy constraints to distribute the nodes among the physical racks and the virtual machines.

CAUTION When you create a cluster with Big Data Extensions, Big Data Extensions disables the cluster's virtual machine automatic migration. Although this prevents vSphere from migrating the virtual machines, it does not prevent you from inadvertently migrating cluster nodes to other hosts by using the vCenter Server user interface. Do not use the vCenter Server user interface to migrate clusters. Performing such management functions outside of the Big Data Extensions environment might break the cluster's placement policy, such as the number of instances per host and the group associations. Even if you do not specify a placement policy, using vCenter Server to migrate clusters can break the default ROUNDROBIN placement policy constraints.

Prerequisites

- Deploy the Serengeti vApp.
- Ensure that you have adequate resources allocated to run the Hadoop cluster.
- To use any Hadoop distribution other than the provided Apache Hadoop, add one or more Hadoop distributions. See the *VMware vSphere Big Data Extensions Administrator's and User's Guide*.
- Create a rack-host mapping information file.
- Upload the rack-host file to the Serengeti server with the `topology upload` command.

Procedure

1 Create a cluster specification file to define the cluster's characteristics, including the node groups, topology, and placement constraints.

NOTE To create an Apache Bigtop, Cloudera CDH4 and CDH5, or Pivotal PHD 1.1 cluster, you must configure a valid DNS and FQDN for the cluster's HDFS and MapReduce traffic. Without valid DNS and FQDN settings, the cluster creation process might fail or the cluster is created but does not function.

In this example, the cluster has `groupAssociations` and `instancePerHost` constraints for the compute node group, and a `groupRacks` constraint for the data node group.

Four data nodes and eight compute nodes are placed on the same four ESXi hosts, which are fairly selected from rack1, rack2, and rack3. Each ESXi host has one data node and two compute nodes. As defined for the compute node group, compute nodes are placed only on ESXi hosts that have data nodes.

This cluster definition requires that you configure datastores and resource pools for at least four hosts, and that there is sufficient disk space for Serengeti to perform the necessary placements during deployment.

```
{
  "nodeGroups":[
    {
      "name": "master",
      "roles": [
        "hadoop_namenode",
        "hadoop_jobtracker"
      ],
      "instanceNum": 1,
      "cpuNum": 2,
      "memCapacityMB": 7500,
    },
    {
      "name": "data",
      "roles": [
        "hadoop_datanode"
      ],
      "instanceNum": 4,
      "cpuNum": 1,
      "memCapacityMB": 3748,
      "storage": {
        "type": "LOCAL",
        "sizeGB": 50
      },
      "placementPolicies": {
        "instancePerHost": 1,
        "groupRacks": {
          "type": "ROUNDROBIN",
          "racks": ["rack1", "rack2", "rack3"]
         },
      }
    },
    {
      "name": "compute",
      "roles": [
        "hadoop_tasktracker"
      ],
      "instanceNum": 8,
      "cpuNum": 2,
      "memCapacityMB": 7500,
      "storage": {
        "type": "LOCAL",
        "sizeGB": 20
      },
      "placementPolicies": {
        "instancePerHost": 2,
        "groupAssociations": [
```

```
            {
              "reference": "data",
              "type": "STRICT"
            }
          }
        },
        {
          "name": "client",
          "roles": [
            "hadoop_client",
            "hive",
            "pig"
          ],
          "instanceNum": 1,
          "cpuNum": 1,
          "storage": {
            "type": "LOCAL",
            "sizeGB": 50
          }
        }
      ],
      "configuration": {
      }
    }
```

2 Access the Serengeti CLI.

3 Run the `cluster create` command, and specify the cluster specification file.

```
cluster create --name cluster_name --specFile full_path/spec_filename
```

Serengeti's Default HBase Cluster Configuration

HBase clusters are required for you to build big table applications. To run HBase MapReduce jobs, configure the HBase cluster to include JobTracker nodes or TaskTracker nodes.

When you create an HBase cluster with the Command-Line Interface, according to the default Serengeti HBase template the resulting cluster consists of the following nodes:

- One master node, which runs the NameNode and HBaseMaster services.
- Three zookeeper nodes, each running the ZooKeeper service.
- Three data nodes, each running the DataNode and HBase Regionserver services.
- One client node, from which you can run Hadoop or HBase jobs.

The default Serengeti-deployed HBase cluster does not contain Hadoop JobTracker or Hadoop TaskTracker daemons. To run an HBase MapReduce job, deploy a customized, nondefault HBase cluster.

Create a Default HBase Cluster with the Serengeti Command-Line Interface

Serengeti supports deploying HBase clusters on HDFS.

Prerequisites

- Deploy the Serengeti vApp.
- Ensure that you have adequate resources allocated to run the Hadoop cluster.

- To use any Hadoop distribution other than the provided Apache Hadoop, add one or more Hadoop distributions. See the *VMware vSphere Big Data Extensions Administrator's and User's Guide*.

Procedure

1 Access the Serengeti CLI.

2 Run the cluster `create command`, and specify the `--type` parameter's value as **`hbase`**.

```
cluster create --name cluster_name --type hbase
```

What to do next

After you deploy the cluster, you can access an HBase database by using several methods. See the *VMware vSphere Big Data Extensions Administrator's and User's Guide*.

Create an HBase Cluster with vSphere HA Protection with the Serengeti Command-Line Interface

You can create HBase clusters with separated Hadoop NameNode and HBase Master roles, and configure vSphere HA protection for the Masters.

Prerequisites

- Deploy the Serengeti vApp.
- Ensure that you have adequate resources allocated to run the Hadoop cluster.
- To use any Hadoop distribution other than the provided Apache Hadoop, add one or more Hadoop distributions. See the *VMware vSphere Big Data Extensions Administrator's and User's Guide*.

Procedure

1 Create a cluster specification file to define the cluster's characteristics, including the node group roles and vSphere HA protection.

In this example, the cluster has JobTracker and TaskTracker nodes, which let you run HBase MapReduce jobs. The Hadoop NameNode and HBase Master roles are separated, and both are protected by vSphere HA.

```
{
  "nodeGroups" : [
    {
      "name" : "zookeeper",
      "roles" : [
        "zookeeper"
      ],
      "instanceNum" : 3,
      "instanceType" : "SMALL",
      "storage" : {
        "type" : "shared",
        "sizeGB" : 20
      },
      "cpuNum" : 1,
      "memCapacityMB" : 3748,
      "haFlag" : "on",
      "configuration" : {
      }
    },
    {
      "name" : "hadoopmaster",
```

```
    "roles" : [
      "hadoop_namenode",
      "hadoop_jobtracker"
    ],
    "instanceNum" : 1,
    "instanceType" : "MEDIUM",
    "storage" : {
      "type" : "shared",
      "sizeGB" : 50
    },
    "cpuNum" : 2,
    "memCapacityMB" : 7500,
    "haFlag" : "on",
    "configuration" : {
    }
  },
  {
    "name" : "hbasemaster",
    "roles" : [
      "hbase_master"
    ],
    "instanceNum" : 1,
    "instanceType" : "MEDIUM",
    "storage" : {
      "type" : "shared",
      "sizeGB" : 50
    },
    "cpuNum" : 2,
    "memCapacityMB" : 7500,
    "haFlag" : "on",
    "configuration" : {
    }
  },

  {
    "name" : "worker",
    "roles" : [
      "hadoop_datanode",
      "hadoop_tasktracker",
      "hbase_regionserver"
    ],
    "instanceNum" : 3,
    "instanceType" : "SMALL",
    "storage" : {
      "type" : "local",
      "sizeGB" : 50
    },
    "cpuNum" : 1,
    "memCapacityMB" : 3748,
    "haFlag" : "off",
    "configuration" : {
    }
  },
  {
    "name" : "client",
```

```
      "roles" : [
        "hadoop_client",
        "hbase_client"
      ],
      "instanceNum" : 1,
      "instanceType" : "SMALL",
      "storage" : {
        "type" : "shared",
        "sizeGB" : 50
      },
      "cpuNum" : 1,
      "memCapacityMB" : 3748,
      "haFlag" : "off",
      "configuration" : {
      }
    }
  ],
  // we suggest running convert-hadoop-conf.rb to generate "configuration" section and paste
the output here
  "configuration" : {
    "hadoop": {
      "core-site.xml": {
        // check for all settings at http://hadoop.apache.org/common/docs/stable/core-
default.html
        // note: any value (int, float, boolean, string) must be enclosed in double quotes
and here is a sample:
        // "io.file.buffer.size": "4096"
      },
      "hdfs-site.xml": {
        // check for all settings at http://hadoop.apache.org/common/docs/stable/hdfs-
default.html
      },
      "mapred-site.xml": {
        // check for all settings at http://hadoop.apache.org/common/docs/stable/mapred-
default.html
      },
      "hadoop-env.sh": {
        // "HADOOP_HEAPSIZE": "",
        // "HADOOP_NAMENODE_OPTS": "",
        // "HADOOP_DATANODE_OPTS": "",
        // "HADOOP_SECONDARYNAMENODE_OPTS": "",
        // "HADOOP_JOBTRACKER_OPTS": "",
        // "HADOOP_TASKTRACKER_OPTS": "",
        // "HADOOP_CLASSPATH": "",
        // "JAVA_HOME": "",
        // "PATH": ""
      },
      "log4j.properties": {
        // "hadoop.root.logger": "DEBUG,DRFA",
        // "hadoop.security.logger": "DEBUG,DRFA"
      },
      "fair-scheduler.xml": {
        // check for all settings at
http://hadoop.apache.org/docs/stable/fair_scheduler.html
        // "text": "the full content of fair-scheduler.xml in one line"
```

```
      },
      "capacity-scheduler.xml": {
        // check for all settings at
http://hadoop.apache.org/docs/stable/capacity_scheduler.html
      },
      "mapred-queue-acls.xml": {
        // check for all settings at
http://hadoop.apache.org/docs/stable/cluster_setup.html#Configuring+the+Hadoop+Daemons
        // "mapred.queue.queue-name.acl-submit-job": "",
        // "mapred.queue.queue-name.acl-administer-jobs", ""
      }
    },
    "hbase": {
      "hbase-site.xml": {
        // check for all settings at http://hbase.apache.org/configuration.html#hbase.site
      },
      "hbase-env.sh": {
        // "JAVA_HOME": "",
        // "PATH": "",
        // "HBASE_CLASSPATH": "",
        // "HBASE_HEAPSIZE": "",
        // "HBASE_OPTS": "",
        // "HBASE_USE_GC_LOGFILE": "",
        // "HBASE_JMX_BASE": "",
        // "HBASE_MASTER_OPTS": "",
        // "HBASE_REGIONSERVER_OPTS": "",
        // "HBASE_THRIFT_OPTS": "",
        // "HBASE_ZOOKEEPER_OPTS": "",
        // "HBASE_REGIONSERVERS": "",
        // "HBASE_SSH_OPTS": "",
        // "HBASE_NICENESS": "",
        // "HBASE_SLAVE_SLEEP": ""
      },
      "log4j.properties": {
        // "hbase.root.logger": "DEBUG,DRFA"
      }
    },
    "zookeeper": {
      "java.env": {
        // "JVMFLAGS": "-Xmx2g"
      },
      "log4j.properties": {
        // "zookeeper.root.logger": "DEBUG,DRFA"
      }
    }
  }
}
```

2 Access the Serengeti CLI.

3 Run the `cluster create` command, and specify the cluster specification file.

```
cluster create --name cluster_name --specFile full_path/spec_filename
```

Managing Hadoop and HBase Clusters

4

You can use the vSphere Web Client to start and stop your Hadoop or HBase cluster and to modify cluster configuration. You can also manage a cluster using the Serengeti Command-Line Interface.

CAUTION Do not use vSphere management functions such as migrating cluster nodes to other hosts for clusters that you create with Big Data Extensions. Performing such management functions outside of the Big Data Extensions environment can make it impossible for you to perform some Big Data Extensions operations, such as disk failure recovery.

- Recover from Disk Failure with the Serengeti Command-Line Interface Client on page 51

  If there is a disk failure in a Hadoop cluster, and the disk does not perform management roles such as NameNode, JobTracker, ResourceManager, HMaster, or ZooKeeper, you can recover by running the Serengeti `cluster fix` command.

Stop and Start a Hadoop or HBase Cluster with the Serengeti Command-Line Interface

You can stop a currently running cluster and start a stopped cluster from the Serengeti CLI.

Prerequisites

- Verify that the cluster is provisioned.
- Verify that enough resources, especially CPU and memory, are available to start the virtual machines in the Hadoop cluster.

Procedure

1 Access the Serengeti CLI.

2 Run the `cluster stop` command.

```
cluster stop --name name_of_cluster_to_stop
```

3 Run the `cluster start` command.

```
cluster start --name name_of_cluster_to_start
```

Scale Out a Hadoop or HBase Cluster with the Serengeti Command-Line Interface

You specify the number of nodes in the cluster when you create Hadoop and HBase clusters. You can later scale out the cluster by increasing the number of worker nodes and client nodes.

IMPORTANT Even if you changed the user password on the cluster's nodes, the changed password is not used for the new nodes that are created when you scale out a cluster. If you set the cluster's initial administrator password when you created the cluster, that initial administrator password is used for the new nodes. If you did not set the cluster's initial administrator password when you created the cluster, new random passwords are used for the new nodes.

Prerequisites

If the cluster is stopped, start it. The cluster status must be RUNNING.

Procedure

1 Access the Serengeti CLI.

2 Run the `cluster resize` command.

For *node_type*, specify worker or client. For the `instanceNum` parameter's *num_nodes* value, use any number that is larger than the current number of *node_type* instances.

```
cluster resize --name name_of_cluster_to_resize --nodeGroup node_type --instanceNum num_nodes
```

Scale CPU and RAM with the Serengeti Command-Line Interface

You can increase or decrease a Hadoop or HBase cluster's compute capacity and RAM to prevent memory resource contention of running jobs.

Serengeti lets you adjust compute and memory resources without increasing the workload on the master node. If increasing or decreasing the cluster's CPU is unsuccessful for a node, which is commonly due to insufficient resources being available, the node is returned to its original CPU setting. If increasing or decreasing the cluster's RAM is unsuccessful for a node, which is commonly due to insufficient resources, the swap disk retains its new setting anyway. The disk is not returned to its original memory setting.

Although all node types support CPU and RAM scaling, do not scale a cluster's master node because Serengeti powers down the virtual machine during the scaling process.

The maximum CPU and RAM settings depend on the virtual machine's version.

Table 4-1. Maximum CPU and RAM Settings

Virtual Machine Version	Maximum Number of CPUs	Maximum RAM, in GB
7	8	255
8	32	1011
9	64	1011
10	64	1011

Prerequisites

Start the cluster if it is not running.

Procedure

1 Access the Serengeti Command-Line Interface.

2 Run the `cluster resize` command to change the number of CPUs or the amount of RAM of a cluster.

- Node types are either worker or client.
- Specify one or both scaling parameters: `--cpuNumPerNode` or `--memCapacityMbPerNode`.

```
cluster resize --name cluster_name --nodeGroup node_type [--cpuNumPerNode vCPUs_per_node] [--
memCapacityMbPerNode memory_per_node]
```

Reconfigure a Hadoop or HBase Cluster with the Serengeti Command-Line Interface

You can reconfigure any Hadoop or HBase cluster that you create with Big Data Extensions.

The cluster configuration is specified by attributes in Hadoop distribution XML configuration files such as: `core-site.xml`, `hdfs-site.xml`, `mapred-site.xml`, `hadoop-env.sh`, `yarn-env.sh`, `yarn-site.sh`, and `hadoop-metrics.properties`.

NOTE Always use the `cluster config` command to change the parameters specified by these configuration files. If you manually modify these files, your changes will be erased if the virtual machine is rebooted, or you use the `cluster config`, `cluster start`, `cluster stop`, or `cluster resize` commands.

Procedure

1 Use the `cluster export` command to export the cluster specification file for the cluster that you want to reconfigure.

```
cluster export --name    cluster_name --specFile file_path/cluster_spec_file_name
```

Option	Description
cluster_name	Name of the cluster that you want to reconfigure.
file_path	The file system path at which to export the specification file.
cluster_spec_file_name	The name with which to label the exported cluster specification file.

2 Edit the configuration information located near the end of the exported cluster specification file.

If you are modeling your configuration file on existing Hadoop XML configuration files, use the `convert-hadoop-conf.rb` conversion tool to convert Hadoop XML configuration files to the required JSON format.

```
…
"configuration": {
    "hadoop": {
      "core-site.xml": {
        // check for all settings at http://hadoop.apache.org/common/docs/stable/core-
default.html
        // note: any value (int, float, boolean, string) must be enclosed in double quotes
and here is a sample:
        // "io.file.buffer.size": "4096"
      },
      "hdfs-site.xml": {
        // check for all settings at http://hadoop.apache.org/common/docs/stable/hdfs-
default.html
      },
      "mapred-site.xml": {
        // check for all settings at http://hadoop.apache.org/common/docs/stable/mapred-
default.html
      },
      "hadoop-env.sh": {
        // "HADOOP_HEAPSIZE": "",
        // "HADOOP_NAMENODE_OPTS": "",
        // "HADOOP_DATANODE_OPTS": "",
        // "HADOOP_SECONDARYNAMENODE_OPTS": "",
        // "HADOOP_JOBTRACKER_OPTS": "",
        // "HADOOP_TASKTRACKER_OPTS": "",
        // "HADOOP_CLASSPATH": "",
        // "JAVA_HOME": "",
        // "PATH": "",
      },
      "log4j.properties": {
        // "hadoop.root.logger": "DEBUG, DRFA ",
        // "hadoop.security.logger": "DEBUG, DRFA ",
      },
      "fair-scheduler.xml": {
        // check for all settings at
http://hadoop.apache.org/docs/stable/fair_scheduler.html
        // "text": "the full content of fair-scheduler.xml in one line"
      },
```

```
      "capacity-scheduler.xml": {
        // check for all settings at
http://hadoop.apache.org/docs/stable/capacity_scheduler.html
      }
    }
  }
…
```

3 (Optional) If your Hadoop distribution's JAR files are not in the `$HADOOP_HOME/lib` directory, add the full path of the JAR file in `$HADOOP_CLASSPATH` to the cluster specification file.

This action lets the Hadoop daemons locate the distribution JAR files.

For example, the Cloudera CDH3 Hadoop Fair Scheduler JAR files are in `/usr/lib/hadoop/contrib/fairscheduler/`. Add the following to the cluster specification file to enable Hadoop to use the JAR files.

```
…
"configuration": {
  "hadoop": {
    "hadoop-env.sh": {
      "HADOOP_CLASSPATH": "/usr/lib/hadoop/contrib/fairscheduler/*:$HADOOP_CLASSPATH"
    },
    "mapred-site.xml": {
      "mapred.jobtracker.taskScheduler": "org.apache.hadoop.mapred.FairScheduler"
      …
    },
    "fair-scheduler.xml": {
      …
    }
  }
}
…
```

4 Access the Serengeti Command-Line Interface.

5 Run the `cluster config` command to apply the new Hadoop configuration.

```
cluster config --name cluster_name --specFile file_path/cluster_spec_file_name
```

6 (Optional) Reset an existing configuration attribute to its default value.

a Remove the attribute from the cluster configuration file's configuration section, or comment out the attribute using double back slashes (//).

b Re-run the `cluster config` command.

About Resource Usage and Elastic Scaling

Scaling lets you adjust the compute capacity of Hadoop data-compute separated clusters. When you enable elastic scaling for a Hadoop cluster, the Serengeti Management Server can stop and start compute nodes to match resource requirements to available resources. You can use manual scaling for more explicit cluster control.

Manual scaling is appropriate for static environments where capacity planning can predict resource availability for workloads. Elastic scaling is best suited for mixed workload environments where resource requirements and availability fluctuate.

When you select manual scaling, Big Data Extensions disables elastic scaling. You can configure the target number of compute nodes for manual scaling. If you do not configure the target number of compute nodes, Big Data Extensions sets the number of active compute nodes to the current number of active compute nodes. If nodes become unresponsive, they remain in the cluster and the cluster operates with fewer functional nodes. In contrast, when you enable elastic scaling, Big Data Extensions manages the number of active TaskTracker nodes according to the range that you specify, replacing unresponsive or faulty nodes with live, responsive nodes.

For both manual and elastic scaling, Big Data Extensions, not vCenter Server, controls the number of active nodes. However, vCenter Server applies the usual reservations, shares, and limits to the cluster's resource pool according to the cluster's vSphere configuration. vSphere DRS operates as usual, allocating resources between competing workloads, which in turn influences how Big Data Extensions dynamically adjusts the number of active nodes in competing Hadoop clusters while elastic scaling is in effect.

Big Data Extensions also lets you adjust cluster nodes' access priority for datastores by using the vSphere Storage I/O Control feature. Clusters configured for HIGH I/O shares receive higher priority access than clusters with NORMAL priority. Clusters configured for NORMAL I/O shares receive higher priority access than clusters with LOW priority. In general, higher priority provides better disk I/O performance.

Scaling Modes

To change between manual and elastic scaling, you change the scaling mode.

- MANUAL. Big Data Extensions disables elastic scaling. When you change to manual scaling, you can configure the target number of compute nodes. If you do not configure the target number of compute nodes, Big Data Extensions sets the number of active compute nodes to the current number of active compute nodes.
- AUTO. Enables elastic scaling. Big Data Extensions manages the number of active compute nodes, maintaining the number of compute nodes in the range from the configured minimum to the configured maximum number of compute nodes in the cluster. If the minimum number of compute nodes is undefined, the lower limit is 0. If the maximum number of compute nodes is undefined, the upper limit is the number of available compute nodes.

  Elastic scaling operates on a per-host basis, at a node-level granularity. That is, the more compute nodes a Hadoop cluster has on a host, the finer the control that Big Data Extensions elasticity can exercise. The tradeoff is that the more compute nodes you have, the higher the overhead in terms of runtime resource cost, disk footprint, I/O requirements, and so on.

  When resources are overcommitted, elastic scaling reduces the number of powered on compute nodes. Conversely, if the cluster receives all the resources it requested from vSphere, and Big Data Extensions determines that the cluster can make use of additional capacity, elastic scaling powers on additional compute nodes.

  Resources can become overcommitted for many reasons, such as:

  - The compute nodes have lower resource entitlements than a competing workload, according to how vCenter Server applies the usual reservations, shares, and limits as configured for the cluster.
  - Physical resources are configured to be available, but another workload is consuming those resources.

  In elastic scaling, Big Data Extensions has two different behaviors for deciding how many active compute nodes to maintain. In both behaviors, Big Data Extensions replaces unresponsive or faulty nodes with live, responsive nodes.

  - Variable. The number of active, healthy TaskTracker compute nodes is maintained from the configured minimum number of compute nodes to the configured maximum number of compute nodes. The number of active compute nodes varies as resource availability fluctuates.
  - Fixed. The number of active, healthy TaskTracker compute nodes is maintained at a fixed number when the same value is configured for the minimum and maximum number of compute nodes.

Default Cluster Scaling Parameter Values

When you create a cluster, its scaling configuration is as follows.

- The cluster's scaling mode is MANUAL, for manual scaling.
- The cluster's minimum number of compute nodes is -1. It appears as "Unset" in the Serengeti CLI displays. Big Data Extensions elastic scaling treats a `minComputeNodeNum` value of -1 as if it were zero (0).
- The cluster's maximum number of compute nodes is -1. It appears as "Unset" in the Serengeti CLI displays. Big Data Extensions elastic scaling treats a `maxComputeNodeNum` value of -1 as if it were unlimited.
- The cluster's target number of nodes is not applicable. Its value is -1. Big Data Extensions manual scaling operations treat a `targetComputeNodeNum` value of -1 as if it were unspecified upon a change to manual scaling.

Interactions Between Scaling and Other Cluster Operations

Some cluster operations cannot be performed while Big Data Extensions is actively scaling a cluster.

If you try to perform the following operations while Big Data Extensions is scaling a cluster in MANUAL mode, Big Data Extensions warns you that in the cluster's current state, the operation cannot be performed.

- Concurrent attempt at manual scaling
- Switch to AUTO mode while manual scaling operations are in progress

If a cluster is in AUTO mode for elastic scaling when you perform the following cluster operations on it, Big Data Extensions changes the scaling mode to MANUAL and changes the cluster to manual scaling. You can re-enable the AUTO mode for elastic scaling after the cluster operation finishes, except if you delete the cluster.

- Delete the cluster
- Repair the cluster
- Stop the cluster

If a cluster is in AUTO mode for elastic scaling when you perform the following cluster operations on it, Big Data Extensions temporarily switches the cluster to MANUAL mode. When the cluster operation finishes, Big Data Extensions returns the scaling mode to AUTO, which re-enables elastic scaling.

- Resize the cluster
- Reconfigure the cluster

If Big Data Extensions is scaling a cluster when you perform an operation that changes the scaling mode to MANUAL, your requested operation waits until the scaling finishes, and then the requested operation begins.

Enable Elastic Scaling for a Hadoop Cluster with the Serengeti Command-Line Interface

When you enable elastic scaling for a data-compute separated Hadoop cluster, Big Data Extensions optimizes cluster performance and utilization of TaskTracker compute nodes.

To enable elastic scaling, set a data-compute separated Hadoop cluster's scaling mode to AUTO and configure the minimum and maximum number of compute nodes. If you do not configure the minimum or maximum number of compute nodes, the previous minimum or maximum setting, respectively, is retained.

To ensure that under contention, elastic scaling keeps a cluster operating with more than a cluster's initial default setting of zero compute nodes, configure the `minComputeNodeNum` parameter value to a nonzero number. To limit the maximum number of compute nodes that can be used in a Hadoop cluster, configure the `maxComputeNodeNum` parameter value to less than the total available compute nodes.

In elastic scaling, Big Data Extensions has two different behaviors for deciding how many active compute nodes to maintain. In both behaviors, Big Data Extensions replaces unresponsive or faulty nodes with live, responsive nodes.

- Variable. The number of active, healthy TaskTracker compute nodes is maintained from the configured minimum number of compute nodes to the configured maximum number of compute nodes. The number of active compute nodes varies as resource availability fluctuates.
- Fixed. The number of active, healthy TaskTracker compute nodes is maintained at a fixed number when the same value is configured for the minimum and maximum number of compute nodes.

Prerequisites

- Understand how elastic scaling and resource usage work. See "About Resource Usage and Elastic Scaling," on page 45.
- Verify that the cluster you want to optimize is data-compute separated. See "About Hadoop and HBase Cluster Deployment Types," on page 17.

Procedure

1 Access the Serengeti CLI.

2 Run the `cluster setParam` command, and set the `--elasticityMode` parameter value to AUTO.

```
cluster setParam --name cluster_name --elasticityMode AUTO [--minComputeNodeNum minNum] [--
maxComputeNodeNum maxNum]
```

Enable Manual Scaling for a Hadoop Cluster with the Serengeti Command-Line Interface

When you enable manual scaling for a cluster, Big Data Extensions disables elastic scaling. When you enable manual scaling, you can configure the target number of compute nodes. If you do not configure the target number of compute nodes, Big Data Extensions sets the number of active compute nodes to the current number of active compute nodes.

Procedure

1 Access the Serengeti CLI.

2 Run the `cluster setParam` command, and set the `--elasticityMode` parameter value to **`MANUAL`**.

```
cluster setParam --name cluster_name --elasticityMode MANUAL [--targetComputeNodeNum
numTargetNodes]
```

Configure Scaling Parameters with the Serengeti Command-Line Interface

You can configure scaling parameters, such as the target number of nodes, with or without changing the scaling mode.

Procedure

1 Access the Serengeti CLI.

2 To display a cluster's scaling settings, run the `cluster list` command.

```
cluster list --detail --name cluster_name
```

3 To configure one or more scaling parameters, run the `cluster setParam` command.

The `--name` parameter is required, and you can include as few or as many of the other parameters as you want. You can repeatedly run the command to configure or reconfigure additional scaling parameters.

```
cluster setParam --name cluster_name --elasticityMode mode --targetComputeNodeNum
numTargetNodes --minComputeNodeNum minNum --maxComputeNodeNum maxNum --ioShares level
```

Paremeter Option	Description
cluster_name	Name of the cluster. Specify this parameter every time you run the `cluster setParam` command.
mode	MANUAL or AUTO.
numTargetNodes	Number of nodes. This parameter is applicable only for MANUAL scaling mode.
minNum	Lower limit of the range of active compute nodes to maintain in the cluster. This parameter is applicable only for AUTO scaling mode.
maxNum	Upper limit of the range of active compute nodes to maintain in the cluster. This parameter is applicable only for AUTO scaling mode.
level	LOW, NORMAL, or HIGH.

4 To reset one or more scaling parameters to their default values, run the `cluster resetParam` command.

The `--name` parameter is required, and you can include as few or as many of the other parameters as you want. You can repeatedly run the command to reset additional scaling parameters.

For data-compute separated nodes, you can reset all the scaling parameters to their defaults by using the `--all` parameter.

```
cluster resetParam --name cluster_name [--all] [--elasticityMode] [--targetComputeNodeNum]
[--minComputeNodeNum] [--maxComputeNodeNum] [--ioShares]
```

Parameter	Description
cluster_name	Name of the cluster. Specify this parameter every time you run the `cluster resetParam` command.
--all	Reset all scaling parameters to their defaults.
--elasticityMode	Sets the scaling mode to MANUAL.
--targetComputeNodeNum	Reset `targetComputeNodeNum` to -1. Big Data Extensions manual scaling operations treat a `targetComputeNodeNum` value of -1 as if it were unspecified upon a change to manual scaling.
--minComputeNodeNum	Reset `minComputeNodeNum` to 0. It appears as "Unset" in the Serengeti CLI displays. Big Data Extensions elastic scaling treats a `minComputeNodeNum` value of -1 as if it were zero (0).
--maxComputeNodeNum	Reset `maxComputeNodeNum` to unlimited. It appears as "Unset" in the Serengeti CLI displays. Big Data Extensions elastic scaling treats a `maxComputeNodeNum` value of -1 as if it were unlimited.
--ioShares	Reset `ioShares` to NORMAL.

Schedule Fixed Elastic Scaling for a Hadoop Cluster

You can enable fixed, elastic scaling according to a preconfigured schedule. Scheduled fixed, elastic scaling provides more control than variable, elastic scaling while still improving efficiency, allowing explicit changes in the number of active compute nodes during periods of predictable usage.

For example, in an office with typical workday hours, there is likely a reduced load on a VMware View resource pool after the office staff goes home. You could configure scheduled fixed, elastic scaling to specify a greater number of compute nodes from 8 PM to 4 AM, when you know that the workload would otherwise be very light.

Prerequisites

From the Serengeti Command-Line Interface, enable the cluster for elastic scaling, and set the `minComputeNodeNum` and `MaxComputeNodeNum` parameters to the same value: the number of active TaskTracker nodes that you want during the period of scheduled fixed elasticity.

Procedure

1 Open a command shell, such as Bash or PuTTY, and log in to the Serengeti Management Server as user `serengeti`.

2 Use any scheduling mechanism that you want to call the `/opt/serengeti/sbin/set_compute_node_num.sh` script to set the number of active TaskTracker compute nodes that you want.

```
/opt/serengeti/sbin/set_compute_node_num.sh --name cluster_name --computeNodeNum
num_TT_to_maintain
```

After the scheduling mechanism calls the `set_compute_node_num.sh` script, fixed, elastic scaling remains in effect with the configured number of active TaskTracker compute nodes until the next scheduling mechanism change or until a user changes the scaling mode or parameters in either the vSphere Web Client or the Serengeti Command-Line Interface.

This example shows how to use a crontab file on the Serengeti Management Server to schedule specific numbers of active TaskTracker compute nodes.

```
# cluster_A: use 20 active TaskTracker compute nodes from 11:00 to 16:00, and 30 compute
nodes the rest of the day
00 11   *  *  *   /opt/serengeti/sbin/set_compute_node_num.sh --name cluster_A --
computeNodeNum 20 >> $HOME/schedule_elasticity.log 2>&1
00 16   *  *  *   /opt/serengeti/sbin/set_compute_node_num.sh --name cluster_A --
computeNodeNum 30 >> $HOME/schedule_elasticity.log 2>&1

# cluster_B: use 3 active TaskTracker compute nodes beginning at 10:00 every weekday
0  10   *  *  1-5 /opt/serengeti/sbin/set_compute_node_num.sh --name cluster_B --
computeNodeNum 3  >> $HOME/schedule_elasticity.log 2>&1

# cluster_C: reset the number of active TaskTracker compute nodes every 6 hours to 15
0  */6  *  *  *   /opt/serengeti/sbin/set_compute_node_num.sh --name cluster_B --
computeNodeNum 15 >> $HOME/schedule_elasticity.log 2>&1
```

Delete a Hadoop or HBase Cluster with the Serengeti Command-Line Interface

You can delete a Hadoop cluster that you no longer need, regardless of whether it is running. When a Hadoop cluster is deleted, all its virtual machines and resource pools are destroyed.

Procedure

1 Access the Serengeti CLI.

2 Run the `cluster delete` command.

```
cluster delete --name cluster_name
```

About vSphere High Availability and vSphere Fault Tolerance

The Serengeti Management Server leverages vSphere HA to protect the Hadoop master node virtual machine, which can be monitored by vSphere.

When a Hadoop NameNode or JobTracker service stops unexpectedly, vSphere restarts the Hadoop virtual machine in another host, reducing unplanned downtime. If vsphere Fault Tolerance is configured and the master node virtual machine stops unexpectedly because of host failover or loss of network connectivity, the secondary node is used, without downtime.

Reconfigure a Node Group with the Serengeti Command-Line Interface

You can reconfigure node groups by modifying node group configuration data in the associated cluster specification file. When you configure a node group, its configuration overrides any cluster level configuration of the same name.

Procedure

1 Access the Serengeti CLI.

2 Run the `cluster export` command to export the cluster's cluster specification file.

```
cluster export --name cluster_name --specFile path_name/spec_file_name
```

3 In the specification file, modify the node group's configuration section with the same content as a cluster-level configuration.

4 Add the customized Hadoop configuration for the node group that you want to reconfigure.

5 Run the `cluster config` command to apply the new Hadoop configuration.

```
cluster config --name cluster_name --specFile path_name/spec_file_name
```

Recover from Disk Failure with the Serengeti Command-Line Interface Client

If there is a disk failure in a Hadoop cluster, and the disk does not perform management roles such as NameNode, JobTracker, ResourceManager, HMaster, or ZooKeeper, you can recover by running the Serengeti `cluster fix` command.

Big Data Extensions uses a large number of inexpensive disk drives for data storage (configured as Just a Bunch of Disks). If several disks fail, the Hadoop data node might shutdown. Big Data Extensions lets you to recover from disk failures.

Serengeti supports recovery from swap and data disk failure on all supported Hadoop distributions. Disks are recovered and started in sequence to avoid the temporary loss of multiple nodes at once. A new disk matches the corresponding failed disk's storage type and placement policies.

The MapR distribution does not support recovery from disk failure by using the `cluster fix` command.

IMPORTANT Even if you changed the user password on the cluster's nodes, the changed password is not used for the new nodes that are created by the disk recovery operation. If you set the cluster's initial administrator password when you created the cluster, that initial administrator password is used for the new nodes. If you did not set the cluster's initial administrator password when you created the cluster, new random passwords are used for the new nodes.

Procedure

1 Access the Serengeti CLI.

2 Run the `cluster fix` command.

The `nodeGroup` parameter is optional.

```
cluster fix --name cluster_name --disk [--nodeGroup nodegroup_name]
```

Monitoring the Big Data Extensions Environment 5

You can monitor the status of Serengeti-deployed clusters, including their datastores, networks, and resource pools through the Serengeti Command-Line Interface. You can also view a list of available Hadoop distributions. Monitoring capabilities are also in the vSphere Web Client.

View Available Hadoop Distributions with the Serengeti Command-Line Interface

You can view a list of Hadoop distributions that are available in your Serengeti deployment. When you create clusters, you can use any available Hadoop distribution.

Procedure

1 Access the Serengeti CLI.

2 Run the `distro list` command.

The available Hadoop distributions are listed, along with their packages.

What to do next

Before you use a distribution, verify that it includes the services that you want to deploy. If services are missing , add the appropriate packages to the distribution.

View Provisioned Hadoop and HBase Clusters with the Serengeti Command-Line Interface

From the Serengeti Command-Line Interface, you can list the provisioned Hadoop and HBase clusters that are in the Serengeti deployment.

Procedure

1 Access the Serengeti CLI.

2 Run the `cluster list` command.

```
cluster list
```

This example displays a specific cluster by including the `--name` parameter.

```
cluster list --name cluster_name
```

This example displays detailed information about a specific cluster by including the `--name` and `--detail` parameters.

```
cluster list --name cluster_name --detail
```

View Datastores with the Serengeti Command-Line Interface

From the Serengeti CLI, you can see the datastores that are in the Serengeti deployment.

Procedure

1 Access the Serengeti CLI.

2 Run the `datastore list` command.

This example displays detailed information by including the `--detail` parameter.

```
datastore list --detail
```

This example displays detailed information about a specific datastore by including the `--name` and `--detail` parameters.

```
datastore list --name datastore_name --detail
```

View Networks with the Serengeti Command-Line Interface

From the Serengeti CLI, you can see the networks that are in the Serengeti deployment.

Procedure

1 Access the Serengeti CLI.

2 Run the `network list` command.

This example displays detailed information by including the `--detail` parameter.

```
network list --detail
```

This example displays detailed information about a specific network by including the `--name` and `--detail` parameters.

```
network list --name network_name --detail
```

View Resource Pools with the Serengeti Command-Line Interface

From the Serengeti CLI, you can see the resource pools that are in the Serengeti deployment.

Procedure

1 Access the Serengeti CLI.

2 Run the `resourcepool list` command.

This example displays detailed information by including the `--detail` parameter.

```
resourcepool list --detail
```

This example displays detailed information about a specific datastore by including the `--name` and `--detail` parameters.

```
resourcepool list --name resourcepool_name --detail
```

Using Hadoop Clusters from the Serengeti Command-Line Interface

6

The Serengeti Command-Line Interface lets you perform Hadoop operations. You can run Hive and Pig scripts, HDFS commands, and MapReduce jobs.

The procedures in this section describe how to use the Serengeti Command-Line Interface, which is typically used by administrators. You can also access the client nodes that are provisioned by the Serengeti Management Server to perform standard Hadoop operations such as moving files and running jobs. And if you have other systems integrated with your Hadoop clusters, those systems can directly communicate with the clusters.

Run HDFS Commands with the Serengeti Command-Line Interface

You can run Hadoop Distributed File System (HDFS) commands from the Serengeti Command-Line Interface Client. The HDFS commands let you directly interact with the HDFS as well as other file systems that Hadoop supports.

The File System (FS) shell provides various commands that let you interact with the HDFS, as well as other file systems that Hadoop supports such as Local FS, HFTP FS, S3 FS, and others. You invoke the FS shell using the `fs` command.

Procedure

1 Access the Serengeti CLI.

2 Run the `cluster list` command to list the available clusters.

3 Connect to the target cluster.

```
cluster target --name cluster_name
```

4 Run HDFS commands.

This example uses the `fs put` command to move files from `/home/serengeti/data` to the HDFS path `/tmp`.

```
fs put --from /home/serengeti/data --to /tmp
```

This example uses the `fs get` command to download a file from a specific HDFS to your local filesystem.

```
fs get --from source_path_and_file --to dest_path_and_file
```

This example uses the `fs ls` command to display the contents of the `dir1` directory in `/user/serengeti`.

```
fs ls /user/serengeti/dir1
```

This example uses the `fs mkdir` command to create the `dir2` directory in the `/user/serengeti` directory.

```
fs mkdir /user/serengeti/dir2
```

Run MapReduce Jobs with the Serengeti Command-Line Interface

You can run MapReduce jobs on your Hadoop cluster.

Procedure

1 Access the Serengeti CLI, and connect to a Hadoop cluster.

2 Show available clusters.

```
cluster list
```

3 Connect to the cluster where you want to run a MapReduce job.

```
cluster target --name cluster_name
```

4 Run the `mr jar` command.

```
mr jar --jarfile path_and_jar_filename --mainclass class_with_main_method [--args
double_quoted_arg_list]
```

This example runs the MapReduce job located in the `hadoop-examples-1.2.1.jar` JAR, which is located in the `/serengeti/cli/lib` directory. Two arguments pass to the MapReduce job: `/tmp/input` and `/tmp/output`.

```
mr jar --jarfile /opt/serengeti/cli/lib/hadoop-examples-1.2.1.jar --mainclass
org.apache.hadoop.examples.WordCount --args "/tmp/input /tmp/output"
```

5 Show the output of the MapReduce job.

```
fs cat file_to_display_to_stdout
```

6 Download the output of the MapReduce job from HDFS to the local file system.

```
fs get --from HDFS_file_path_and_name --to local_file_path_and_name
```

Run Pig and PigLatin Scripts with the Serengeti Command-Line Interface

You can run Pig and PigLatin scripts from the Serengeti Command-Line Interface.

Pig lets you write PigLatin statements. PigLatin statements are converted by the Pig service into MapReduce jobs, which are executed across your Hadoop cluster.

Prerequisites

Create a PigLatin script to execute against your Hadoop cluster.

Procedure

1 Access the Serengeti CLI.

2 Show available clusters.

```
cluster list
```

3 Connect to the cluster where you want to run a Pig script.

```
cluster target --name cluster_name
```

4 Run the `pig script` command to run an existing PigLatin script.

This example runs the PigLatin script `data.pig` located in the `/pig/scripts` directory.

```
pig script --location /pig/scripts/data.pig
```

What to do next

If the PigLatin script stores its results in a file, you might want to copy that file to your local file system.

Run Hive and Hive Query Language Scripts with the Serengeti Command-Line Interface

You can run Hive and Hive Query Language (HQL) scripts from the Serengeti Command-Line Interface Client.

Hive lets you write HQL statements. HQL statements are converted by the Hive service into MapReduce jobs, which are executed across your Hadoop cluster.

Prerequisites

Create an HQL script to execute against your Hadoop cluster.

Procedure

1 Access the Serengeti CLI.

2 Show available clusters by running the `cluster list` command.

```
cluster list
```

3 Connect to the cluster where you want to run a Hive script.

```
cluster target --name cluster_name
```

4 Run the `hive script` command to run an existing Hive script.

This example runs the Hive script `data.hive` located in the `/hive/scripts` directory. `hive script --location /hive/scripts/hive.data`

Cluster Specification Reference 7

This information describes the Serengeti cluster specification file's attributes and their mapping to Hadoop attributes, and how to convert a Hadoop XML configuration file to a Serengeti configuration file.

Cluster Specification File Requirements

A cluster specification file is a text file with the configuration attributes provided in a JSON-like formatted structure. Cluster specification files must adhere to requirements concerning syntax, quotation mark usage, and comments.

- To parse cluster specification files, Serengeti uses the Jackson JSON Processor. For syntax requirements, such as the truncation policy for float types, see the Jackson JSON Processor Wiki.

- Always enclose digital values in quotation marks. For example:

  ```
  "mapred.tasktracker.reduce.tasks.maximum" : "2"
  ```

  The quotation marks ensure that integers are correctly interpreted instead of being converted to double-precision floating point, which can cause unintended consequences.

- Do not include any comments.

Cluster Definition Requirements

Cluster specification files contain configuration definitions for clusters, such as their roles and node groups. Cluster definitions must adhere to requirements concerning node group roles, cluster roles, and instance numbers.

A cluster definition has the following requirements:

- Node group roles cannot be empty. You can determine the valid role names for your Hadoop distribution by using the `distro list` command.
- The hadoop_namenode and hadoop_jobtracker roles must be configured in a single node group.
  - In Hadoop 2.0 clusters, such as CDH4 or Pivotal HD, the instance number can be greater than 1 to create an HDFS HA or Federation cluster.
  - Otherwise, the total instance number must be 1.
- Node group instance numbers must be positive numbers.

Annotated Cluster Specification File

The Serengeti cluster specification file defines the nodes, resources, and so on for a cluster. You can use this annotated cluster specification file, and the sample files in `/opt/serengeti/samples`, as models when you create your clusters.

The following code is a typical cluster specification file. For code annotations, see Table 7-1.

```
 1 {
 2   "nodeGroups" : [
 3     {
 4       "name": "master",
 5       "roles": [
 6         "hadoop_namenode",
 7         "hadoop_jobtracker"
 8       ],
 9       "instanceNum": 1,
10       "instanceType": "LARGE",
11       "cpuNum": 2,
12       "memCapacityMB":4096,
13       "storage": {
14         "type": "SHARED",
15         "sizeGB": 20
16       },
17       "haFlag":"on",
18       "rpNames": [
19         "rp1"
20       ]
21     },
22     {
23       "name": "data",
24       "roles": [
```

```
25          "hadoop_datanode"
26        ],
27        "instanceNum": 3,
28        "instanceType": "MEDIUM",
29        "cpuNum": 2,
30        "memCapacityMB":2048,
31        "storage": {
32          "type": "LOCAL",
33          "sizeGB": 50,
34          "dsNames4Data": ["DSLOCALSSD"],
35          "dsNames4System": ["DSNDFS"]
36        }
37        "placementPolicies": {
38          "instancePerHost": 1,
39          "groupRacks": {
40            "type": "ROUNDROBIN",
41            "racks": ["rack1", "rack2", "rack3"]
42           }
43        }
44      },
45      {
46        "name": "compute",
47        "roles": [
48          "hadoop_tasktracker"
49        ],
50        "instanceNum": 6,
51        "instanceType": "SMALL",
52        "cpuNum": 2,
53        "memCapacityMB":2048,
54        "storage": {
55          "type": "LOCAL",
56          "sizeGB": 10
57        }
58        "placementPolicies": {
59          "instancePerHost": 2,
60           "groupAssociations": [{
61             "reference": "data",
62             "type": "STRICT"
63           }]
64        }
65      },
66      {
67        "name": "client",
68        "roles": [
69          "hadoop_client",
70          "hive",
71          "hive_server",
72          "pig"
73        ],
74        "instanceNum": 1,
75        "instanceType": "SMALL",
76        "memCapacityMB": 2048,
77        "storage": {
78          "type": "LOCAL",
79          "sizeGB": 10,
```

```
80         "dsNames": [“ds1”, “ds2”]
81       }
82     }
83  ],
84  "configuration": {
85  }
86 }
```

The cluster definition elements are defined in the table.

Table 7-1. Example Cluster Specification Annotation

Line(s)	Attribute	Example Value	Description
4	name	master	Node group name.
5-8	role	hadoop_namenode, hadoop_jobtracker	Node group role. `hadoop_namenode` and `hadoop_jobtracker` are deployed to the node group's virtual machine.
9	instanceNum	1	Number of instances in the node group. Only one virtual machine is created for the group. ■ You can have multiple instances for hadoop_tasktracker, hadoop_datanode, hadoop_client, pig, and hive. ■ For HDFS1 clusters, you can have only one instance of hadoop_namenode and hadoop_jobtracker. ■ For HDFS2 clusters, you can have two hadoop_namenode instances. ■ With a MapR distribution, you can configure multiple instances of hadoop_jobtracker.
10	instanceType	LARGE	Node group instance type. Instance types are predefined virtual machine specifications, which are combinations of the number of CPUs, RAM sizes, and storage size. The predefined numbers can be overridden by the `cpuNum`, `memCapacityMB`, and storage attributes in the Serengeti server specification file.

Table 7-1. Example Cluster Specification Annotation (Continued)

Line(s)	Attribute	Example Value	Description
11	cpuNum	2	Number of CPUs per virtual machine. This attribute overrides the number of vCPUs in the predefined virtual machine specification.
12	memCapacityMB	4096	RAM size, in MB, per virtual machine. This attribute overrides the RAM size in the predefined virtual machine specification.
13-16	storage	See lines 14-15 for one group's storage attributes	Node group storage requirements.
14	type	SHARED	Storage type. The node group is deployed using only shared storage.
15	sizeGB	20	Storage size. Each node in the node group is deployed with 20GB available disk space.
17	haFlag	on	HA protection for the node group. The node group is deployed with vSphere HA protection.
18-20	rpNames	rp1	Resource pools under which the node group virtual machines are deployed. These pools can be an array of values.
22-36	Node group definition for the data node		See lines 3-21, which define the same attributes for the master node. In lines 34-35, data disks are placed on `dsNames4Data` datastores, and system disks are placed on `dsNames4System` datastores.
37-44	placementPolicies	See code sample	Data node group's placement policy constraints. You need at least three ESXi hosts because there are three instances and a requirement that each instance be on its own host. This group is provisioned on hosts on rack1, rack2, and rack3 by using a ROUNDROBIN algorithm.
45-57	Node group definition for the compute node		See lines 4-16, which define the same attributes for the master node.

Table 7-1. Example Cluster Specification Annotation (Continued)

Line(s)	Attribute	Example Value	Description
58-65	placementPolicies	See code sample	Compute node group's placement policy constraints. You need at least three ESXi hosts to meet the instance requirements. The compute node group references a data node group through STRICT typing. The two compute instances use a data instance on the ESXi host. The STRICT association provides better performance.
66-82	Node group definition for the client node		See previous node group definitions.
83-86	configuration	Empty in the code sample	Hadoop configuration customization.

Cluster Specification Attribute Definitions

Cluster definitions include attributes for the cluster itself and for each of the cluster's node groups.

Cluster Specification Outer Attributes

Cluster specification outer attributes apply to the cluster as a whole.

Table 7-2. Cluster Specification Outer Attributes

Attribute	Type	Mandatory/ Optional	Description
nodeGroups	object	Mandatory	One or more group specifications. See Table 7-3.
configuration	object	Optional	Customizable Hadoop configuration key/value pairs.
externalHDFS	string	Optional	Valid only for compute-only clusters. URI of external HDFS.

Cluster Specification Node Group Objects and Attributes

Node group objects and attributes apply to one node group in a cluster.

Table 7-3. Cluster Specification's Node Group Objects and Attributes

Attribute	Type	Mandatory/ Optional	Description
name	string	Mandatory	User defined node group name.
roles	list of string	Mandatory	List of software packages or services to install on the node group's virtual machine. Values must match the roles displayed by the `distro list` command.

Table 7-3. Cluster Specification's Node Group Objects and Attributes (Continued)

Attribute	Type	Mandatory/ Optional	Description
instanceNumber	integer	Mandatory	Number of virtual machines in the node group: ■ Positive integer. ■ Generally, you can have multiple instances for `hadoop_tasktracker`, `hadoop_datanode`, `hadoop_client`, `pig`, and `hive`. ■ For HDFS1 clusters, you can have only one instance of `hadoop_namenode` and `hadoop_jobtracker`. ■ For HDFS2 clusters, you can have two `hadoop_namenode` instances. ■ With a MapR distribution, you can configure multiple instances of `hadoop_jobtracker`.
instanceType	string	Optional	Size of virtual machines in the node group, expressed as the name of a predefined virtual machine template. See Table 7-4. ■ SMALL ■ MEDIUM ■ LARGE ■ EXTRA_LARGE If you specify `cpuNum` , `memCapacityMB` , or `sizeGB` attributes, they override the corresponding value of your selected virtual machine template for the applicable node group.
cpuNum	integer	Optional	Number of CPUs per virtual machine. If the `haFlag` value is FT, the `cpuNum` value must be 1.
memCapacityMB	integer	Optional	RAM size, in MB, per virtual machine. **Note** When using MapR 3.1, you must specify a minimum of 5120 MBs of memory capacity for the zookeeper, worker, and client nodes.
Storage	object	Optional	Storage settings.
type	string	Optional	Storage type: ■ LOCAL. For local storage ■ SHARED. For shared storage.
sizeGB	integer	Optional	Data storage size. Must be a positive integer.
dsNames	list of string	Optional	Array of datastores the node group can use.
dnNames4Data	list of string	Optional	Array of datastores the data node group can use.
dsNames4System	list of string	Optional	Array of datastores the system can use.
rpNames	list of string	Optional	Array of resource pools the node group can use.
haFlag	string	Optional	By default, NameNode and JobTracker nodes are protected by vSphere HA. ■ on. Protect the node with vSphere HA. ■ ft. Protect the node with vSphere FT. ■ off. Do not use vSphere HA or vSphere FT.
placementPolicies	object	Optional	Up to three optional constraints: ■ instancePerHost ■ groupRacks ■ groupAssociations

Serengeti Predefined Virtual Machine Sizes

Serengeti provides predefined virtual machine sizes to use for defining the size of virtual machines in a cluster node group.

Table 7-4. Serengeti Predefined Virtual Machine Sizes

	SMALL	MEDIUM	LARGE	EXTRA_LARGE
Number of CPUs per virtual machine	1	2	4	8
RAM, in GB	3.75	7.5	15	30
Hadoop master data disk size, in GB	25	50	100	200
Hadoop worker data disk size, in GB	50	100	200	400
Hadoop client data disk size, in GB	50	100	200	400
Zookeeper data disk size, in GB	20	40	80	120

White Listed and Black Listed Hadoop Attributes

White listed attributes are Apache Hadoop attributes that you can configure from Serengeti with the `cluster config` command. The majority of Apache Hadoop attributes are white listed. However, there are a few black listed Apache Hadoop attributes, which you cannot configure from Serengeti.

If you use an attribute in the cluster specification file that is neither a white listed nor a black listed attribute, and then run the `cluster config` command, a warning appears and you must answer yes to continue or no to cancel.

If your cluster includes a NameNode or JobTracker, Serengeti configures the `fs.default.name` and `dfs.http.address` attributes. You can override these attributes by defining them in your cluster specification.

Table 7-5. Configuration Attribute White List

File	Attributes
`core-site.xml`	All `core-default` configuration attributes listed on the Apache Hadoop 2.x documentation Web page. For example, `http://hadoop.apache.org/docs/`*`branch_name`*`/core-default.html`. Exclude the attributes defined in the black list.
`hdfs-site.xml`	All `hdfs-default` configuration attributes listed on the Apache Hadoop 2.x documentation Web page. For example, `http://hadoop.apache.org/docs/`*`branch_name`*`/hdfs-default.html`. Exclude the attributes defined in the black list.
`mapred-site.xml`	All `mapred-default` configuration attributes listed on the Apache Hadoop 2.x documentation Web page. For example, `http://hadoop.apache.org/docs/`*`branch_name`*`/mapred-default.html`. Exclude the attributes defined in the black list.

Table 7-5. Configuration Attribute White List (Continued)

File	Attributes
hadoop-env.sh	JAVA_HOME PATH HADOOP_CLASSPATH HADOOP_HEAPSIZE HADOOP_NAMENODE_OPTS HADOOP_DATANODE_OPTS HADOOP_SECONDARYNAMENODE_OPTS HADOOP_JOBTRACKER_OPTS HADOOP_TASKTRACKER_OPTS HADOOP_LOG_DIR
log4j.properties	hadoop.root.logger hadoop.security.logger log4j.appender.DRFA.MaxBackupIndex log4j.appender.RFA.MaxBackupIndex log4j.appender.RFA.MaxFileSize
fair-scheduler.xml	text All fair_scheduler configuration attributes listed on the Apache Hadoop 2.x documentation Web page that can be used inside the text field. For example, http://hadoop.apache.org/docs/*branch_name*/fair_scheduler.html. Exclude the attributes defined in the black list.
capacity-scheduler.xml	All capacity_scheduler configuration attributes listed on the Apache Hadoop 2.x documentation Web page. For example, http://hadoop.apache.org/docs/*branch_name*/capacity_scheduler.html. Exclude attributes defined in black list
mapred-queue-acls.xml	All mapred-queue-acls configuration attributes listed on the Apache Hadoop 2.x Web page. For example, http://hadoop.apache.org/docs/*branch_name*/cluster_setup.html#Configuring+the+Hadoop+Daemons. Exclude the attributes defined in the black list.

Table 7-6. Configuration Attribute Black List

File	Attributes
core-site.xml	net.topology.impl net.topology.nodegroup.aware dfs.block.replicator.classname topology.script.file.name
hdfs-site.xml	dfs.http.address dfs.name.dir dfs.data.dir
mapred-site.xml	mapred.job.tracker mapred.local.dir mapred.task.cache.levels mapred.jobtracker.jobSchedulable mapred.jobtracker.nodegroup.aware
hadoop-env.sh	HADOOP_HOME HADOOP_COMMON_HOME HADOOP_MAPRED_HOME HADOOP_HDFS_HOME HADOOP_CONF_DIR HADOOP_PID_DIR

Table 7-6. Configuration Attribute Black List (Continued)

File	Attributes
log4j.properties	None
fair-scheduler.xml	None
capacity-scheduler.xml	None
mapred-queue-acls.xml	None

Convert Hadoop XML Files to Serengeti JSON Files

If you defined a lot of attributes in your Hadoop configuration files, you can convert that configuration information into the JSON format that Serengeti can use.

Procedure

1 Copy the directory `$HADOOP_HOME/conf/` from your Hadoop cluster to the Serengeti Management Server.

2 Open a command shell, such as Bash or PuTTY, log in to the Serengeti Management Server, and run the `convert-hadoop-conf.rb` Ruby conversion script.

```
convert-hadoop-conf.rb path_to_hadoop_conf
```

The converted Hadoop configuration attributes, in JSON format, appear.

3 Open the cluster specification file for editing.

4 Replace the cluster level configuration or group level configuration items with the output that was generated by the `convert-hadoop-conf.rb` Ruby conversion script.

What to do next

Access the Serengeti CLI, and use the new specification file.

- To apply the new configuration to a cluster, run the `cluster config` command. Include the `--specFile` parameter and its value: the new specification file.
- To create a cluster with the new configuration, run the `cluster create` command. Include the `--specFile` parameter and its value: the new specification file.

Serengeti CLI Command Reference

8

This section provides descriptions and syntax requirements for every Serengeti CLI command.

- resourcepool Commands on page 94

  The `resourcepool {*}` commands let you manage resource pools.

- topology Commands on page 95

  The `topology {*}` commands let you manage cluster topology.

cfg Commands

The `cfg {*}` commands let you view and specify configuration information for your MapReduce jobs. These commands let you set the configuration properties and their values in the `core-site.xml`, `hdfs-site.xml`, and `mapred-site.xml` configuration files for jobs started from the command-line interface using `mr` commands.

- cfg fs Command on page 72

  The `cfg fs` command lets you specify the node running the Hadoop NameNode service.

- cfg info Command on page 73

  The `cfg info` command lets you get information about your Hadoop cluster's configuration.

- cfg jt Command on page 73

  The `cfg jt` command lets you specify which node to use for the Hadoop JobTracker service.

- cfg load Command on page 73

  The `cfg load` command lets you load (or copy) Hadoop configuration parameters from an exiting configuration file. The resource can be a local Hadoop configuration file such as `core-site.xml`, `hdfs-site.xml`, or `mapred-site.xml`, or a URL.

- cfg props get Command on page 73

  The `cfg props get` command lets you view the value of a Hadoop property that you specify.

- cfg props list Command on page 73

  The `cfg props list` command lets you view the value of all the Hadoop cluster configuration properties.

- cfg props set Command on page 73

  The `cfg props set` command lets you set the value of the Hadoop property you specify.

cfg fs Command

The `cfg fs` command lets you specify the node running the Hadoop NameNode service.

The NameNode manages the filesystem namespace. It maintains the filesystem tree and the metadata for all the files and directories in the tree. This information is stored persistently on the local disk in the form of two files: the namespace image and the edit log. The NameNode also knows the DataNodes on which all the blocks for a given file are located.

The `cfg fs` command sets the variable `fs.default.name` to the hostname and port number on which you intend to run the NameNode. Typically there is a single master node which runs both the NameNode and HBaseMaster services.

Parameter	Mandatory/Optional	Description
--namenode *hostname-of-namenode:port_num*	Mandatory	Specifies the NameNode address, which can be `local` or *`namenode:port_num`*

cfg info Command

The `cfg info` command lets you get information about your Hadoop cluster's configuration.

You can use the `cfg info` command to view information about your Hadoop cluster's configuration.

cfg jt Command

The `cfg jt` command lets you specify which node to use for the Hadoop JobTracker service.

There are two types of nodes that control the job execution process: a JobTracker and a number of TaskTrackers. The JobTracker coordinates all the jobs run on the system by scheduling tasks to run on TaskTrackers. Tasktrackers run tasks and send progress reports to the JobTracker, which keeps a record of the overall progress of each job. If a task fails, the JobTracker can reschedule it on a different TaskTracker.

The `cfg jt` command sets the variable `mapred.job.tracker` to the hostname and port number on which you intend to run the JobTracker.

Parameter	Mandatory/Optional	Description
--jobtracker *hostname-of-jobtracker:port_num*	Mandatory	Specifies the JobTracker address, which can be `local` or *`jobtracker:port_num`*

cfg load Command

The `cfg load` command lets you load (or copy) Hadoop configuration parameters from an exiting configuration file. The resource can be a local Hadoop configuration file such as `core-site.xml`, `hdfs-site.xml`, or `mapred-site.xml`, or a URL.

Parameter	Mandatory/Optional	Description
--location *file_path*	Mandatory	Path and file from which to load the configuration data. For example: **`"local_dir/hdfs-site.xml"`**. You can also specify the URL of a file.

cfg props get Command

The `cfg props get` command lets you view the value of a Hadoop property that you specify.

Parameter	Mandatory/Optional	Description
--key	Mandatory	The name of the property whose value you want to view. For example: `fs.default.name`

cfg props list Command

The `cfg props list` command lets you view the value of all the Hadoop cluster configuration properties.

You can use the `cfg props list` command to view the value of all Hadoop cluster configuration properties.

cfg props set Command

The `cfg props set` command lets you set the value of the Hadoop property you specify.

Parameter	Mandatory/Optional	Description
--property *name=value*	Mandatory	Specify the property name and the value to assign.

cluster Commands

The `cluster {*}` commands let you connect to Hadoop and HBase clusters, create and delete clusters, stop and start clusters, and perform cluster management operations.

- cluster config Command on page 75

  The `cluster config` command lets you modify the configuration of an existing Hadoop or HBase cluster, whether the cluster is the configured according to the Serengeti defaults or you have customized the cluster.

- cluster create Command on page 75

  The `cluster create` command lets you create a Hadoop or HBase cluster.

- cluster delete Command on page 76

  The `cluster delete` command lets you delete a Hadoop or HBase cluster in Serengeti. When a cluster is deleted, all its virtual machines and resource pools are destroyed.

- cluster export Command on page 76

  The `cluster export` command lets you export cluster configuration information to a cluster specification file or to display the cluster configuration information to the console.

- cluster fix Command on page 77

  The `cluster fix` command lets you recover from a failed disk.

- cluster list Command on page 77

  The `cluster list` command lets you view a list of provisioned clusters in Serengeti. You can see the following information: name, distribution, status, and each node group's information. The node group information consists of the instance count, CPU, memory, type, and size.

- cluster resetParam Command on page 77

  The `cluster resetParam` command lets you reset a cluster's scaling parameters and ioShares level to default values. You must specify at least one optional parameter.

- cluster resize Command on page 78

  The `cluster resize` command lets you change the number of nodes in a node group or scale up/down cluster CPU or RAM. You must specify at least one optional parameter.

- cluster setParam Command on page 78

  The `cluster setParam` command lets you set scaling parameters and the ioShares priority for a Hadoop cluster in Serengeti. You must specify at least one optional parameter.

- cluster start Command on page 79

  The `cluster start` command lets you start a cluster in Serengeti.

- cluster stop Command on page 79

  The `cluster stop` command lets you stop a Hadoop cluster in Serengeti.

- cluster target Command on page 80

  The `cluster target` command lets you connect to a cluster to run `fs`, `mr`, `pig`, and `hive` commands with the Serengeti CLI. You must rerun the *cluster target* command if it has been more than 30 minutes since you ran it in your current Serengeti Command-Line Interface session.

cluster config Command

The `cluster config` command lets you modify the configuration of an existing Hadoop or HBase cluster, whether the cluster is the configured according to the Serengeti defaults or you have customized the cluster.

You can use the `cluster config` command with the `cluster export` command to return cluster services and the original Hadoop configuration to normal in the following situations:

- A service such as NameNode, JobTracker, DataNode, or TaskTracker goes down.
- You manually changed the Hadoop configuration of one or more of the cluster's nodes.

Run the `cluster export` command, and then run the `cluster config` command. Include the new cluster specification file that you just exported.

Table 8-1.

Parameter	Mandatory/Optional	Description
`--name` *cluster_name_in_Serengeti*	Mandatory	Name of Hadoop cluster to configure.
`--specFile` *spec_file_path*	Optional	File name of Hadoop cluster specification
`--yes`	Optional	Answer Y to Y/N confirmation. If not specified, manually type y or n.
`--skipConfigValidation`	Optional	Skip cluster configuration validation.

cluster create Command

The `cluster create` command lets you create a Hadoop or HBase cluster.

If the cluster specification does not include the required nodes, for example a master node, Serengeti creates the cluster according to the default Serengeti-deployed cluster configuration.

When you create a MapReduce cluster with the Serengeti Command-Line Interface, by default you create a MapReduce v1 cluster. To create a MapReduce v2 (YARN) cluster, create a cluster specification file modeled after the `/opt/serengeti/samples/default_hadoop_yarn_cluster.json` file, and specify the `--specFile` parameter and your cluster specification file in the `cluster create ...` command.

Table 8-2.

Parameter	Mandatory/Optional	Description
`--name` *cluster_name_in_Serengeti*	Mandatory	Cluster name
`--type` *cluster_type*	Optional	Cluster type: ■ Hadoop, which is the default ■ HBase
`--password`	Optional Do not use if you use the `--resume` parameter	Custom password for all the nodes in the cluster. Passwords are from 8 to 128 characters, and include only alphanumeric characters ([0-9, a-z, A-Z]) and the following special characters: _ @ # $ % ^ & *
`--specFile` *spec_file_path*	Optional	Cluster specification filename
`--distro` *Hadoop_distro_name*	Optional	Hadoop distribution for the cluster.

Table 8-2. (Continued)

Parameter	Mandatory/Optional	Description
`--dsNames` *datastore_names*	Optional	Datastore to use to deploy Hadoop cluster in Serengeti. Multiple datastores can be used, separated by ",". By default, all available datastores are used. When you specify the `--dsNames` parameter, the cluster can use only those datastores that you provide in this command.
`--networkName` *management_network_name*	Mandatory If you omit any of the optional network parameters, the traffic associated with that parameter is routed on the management network that you specify by the `--networkName` parameter.	Network to use for management traffic in Hadoop clusters.
`--hdfsNetworkName` *hdfs_network_name*	Optional	Network to use for HDFS traffic in Hadoop clusters.
`--mapredNetworkName` *mapred_network_name*	Optional	Network to use for MapReduce traffic in Hadoop clusters.
`--rpNames` *resource_pool_name*	Optional	Resource pool to use for Hadoop clusters. Multiple resource pools can be used, separated by ",".
`--resume`	Optional Do not use if you use the `--password` parameter	Recover from a failed deployment process.
`--topology` *topology_type*	Optional	Topology type for rack awareness: HVE, RACK_AS_RACK, or HOST_AS_RACK.
`--yes`	Optional	Answer Y to Y/N confirmation. If not specified, manually type y or n.
`--skipConfigValidation`	Optional	Skip cluster configuration validation.

cluster delete Command

The `cluster delete` command lets you delete a Hadoop or HBase cluster in Serengeti. When a cluster is deleted, all its virtual machines and resource pools are destroyed.

Table 8-3.

Parameter	Mandatory/Optional	Description
`--name` *cluster_name*	Mandatory	Name of cluster to delete.

cluster export Command

The `cluster export` command lets you export cluster configuration information to a cluster specification file or to display the cluster configuration information to the console.

Table 8-4.

Parameter	Mandatory/Optional	Description
`--name` *cluster_name*	Mandatory	Name of cluster to export
`--specFile`	Optional	File name for exported cluster specification. If not specified, the output appears in the console.

cluster fix Command

The `cluster fix` command lets you recover from a failed disk.

IMPORTANT Even if you changed the user password on the cluster's nodes, the changed password is not used for the new nodes that are created by the disk recovery operation. If you set the cluster's initial administrator password when you created the cluster, that initial administrator password is used for the new nodes. If you did not set the cluster's initial administrator password when you created the cluster, new random passwords are used for the new nodes.

Table 8-5.

Parameter	Mandatory/Optional	Description
`--name` *cluster_name*	Mandatory	Name of cluster that has a failed disk.
`--disk`	Required	Recover node disks.
`--nodeGroup` *nodegroup_name*	Optional	Perform scan and recovery only on the specified node group, not on all the management nodes in the cluster.

cluster list Command

The `cluster list` command lets you view a list of provisioned clusters in Serengeti. You can see the following information: name, distribution, status, and each node group's information. The node group information consists of the instance count, CPU, memory, type, and size.

Table 8-6.

Parameter	Mandatory/Optional	Description
`--name` *cluster_name_in_Serengeti*	Optional	Name of cluster to list.
`--detail`	Optional	List cluster details, including name in Serengeti, distribution, deploy status, each node's information in different roles. If you specify this option, Serengeti queries the vCenter Server to get the latest node status.

cluster resetParam Command

The `cluster resetParam` command lets you reset a cluster's scaling parameters and ioShares level to default values. You must specify at least one optional parameter.

Table 8-7.

Parameter	Mandatory/Optional	Description
`--name` *cluster_name*	Mandatory	Name of cluster for which to reset scaling parameters.
`--all`	Optional	Reset all scaling parameters to their defaults.
`--elasticityMode`	Optional	Reset auto to false.
`--targetComputeNodeNum`	Optional	Reset to -1. Big Data Extensions manual scaling operations treat a `targetComputeNodeNum` value of -1 as if it were unspecified upon a change to manual scaling.
`--minComputeNodeNum`	Optional	Reset to -1. It appears as "Unset" in the Serengeti CLI displays. Big Data Extensions elastic scaling treats a `minComputeNodeNum` value of -1 as if it were zero (0).

Table 8-7. (Continued)

Parameter	Mandatory/Optional	Description
`--maxComputeNodeNum`	Optional	Reset to -1. It appears as "Unset" in the Serengeti CLI displays. Big Data Extensions elastic scaling treats a `maxComputeNodeNum` value of -1 as if it were unlimited.
`--ioShares`	Optional	Reset to NORMAL.

cluster resize Command

The `cluster resize` command lets you change the number of nodes in a node group or scale up/down cluster CPU or RAM. You must specify at least one optional parameter.

If you specify the `--instanceNum` parameter, you cannot specify either the `--cpuNumPerNode` parameter or the `--memCapacityMbPerNode` parameter.

You can specify the `--cpuNumPerNode` and the `--memCapacityMbPerNode` parameters at the same time to scale the CPU and RAM with a single command.

IMPORTANT Even if you changed the user password on the cluster's nodes, the changed password is not used for the new nodes that are created when you scale out a cluster. If you set the cluster's initial administrator password when you created the cluster, that initial administrator password is used for the new nodes. If you did not set the cluster's initial administrator password when you created the cluster, new random passwords are used for the new nodes.

Table 8-8.

Parameter	Mandatory/Optional	Description
`--name` *cluster_name_in_Serengeti*	Mandatory	Target Hadoop cluster in Serengeti.
`--nodeGroup` *name_of_the_node_group*	Mandatory	Target role to scale out in the cluster deployed by Serengeti.
`--instanceNum` *instance_number*	Optional	Target count to which to scale out. Must be greater than the original count.
`--cpuNumPerNode` *num_of_vCPUs*	Optional	Number of vCPUs to use for a node group
`--memCapacityMbPerNode` *size_in_MB*	Optional	Total memory, in MB, to use for the node group

cluster setParam Command

The `cluster setParam` command lets you set scaling parameters and the ioShares priority for a Hadoop cluster in Serengeti. You must specify at least one optional parameter.

In elastic scaling, Big Data Extensions has two different behaviors for deciding how many active compute nodes to maintain. In both behaviors, Big Data Extensions replaces unresponsive or faulty nodes with live, responsive nodes.

- Variable. The number of active, healthy TaskTracker compute nodes is maintained from the configured minimum number of compute nodes to the configured maximum number of compute nodes. The number of active compute nodes varies as resource availability fluctuates.
- Fixed. The number of active, healthy TaskTracker compute nodes is maintained at a fixed number when the same value is configured for the minimum and maximum number of compute nodes.

Table 8-9.

Parameter	Mandatory/Optional	Description
`--name` *cluster_name*	Mandatory	Name of cluster for which to set elasticity parameters.
`--elasticityMode` *mode*	Optional	MANUAL or AUTO.
`--targetComputeNodeNum` *numTargetNodes*	Optional This parameter is applicable only for the MANUAL scaling mode. If the cluster is in AUTO mode or you are changing it to AUTO mode, this parameter is ignored.	Number of compute nodes for the specified Hadoop cluster or node group within that cluster. Must be an integer >= 0. ■ If zero (0), all the nodes in the specific Hadoop cluster, or if `nodeGroup` is specified, in the node group, are decommissioned and powered off. ■ For integers from one to the max number of nodes in the Hadoop cluster, the specified number of nodes remain commissioned and powered on, and the remaining nodes are decommissioned. ■ For integers > the max number of nodes in the Hadoop cluster, all the nodes in the specified Hadoop cluster or node group are re-commissioned and powered on.
`--minComputeNodeNum` *minNum*	Optional This parameter is applicable only for the AUTO scaling mode. If the cluster is in MANUAL mode or you are changing it to MANUAL mode, this parameter is ignored.	Lower limit of the range of active compute nodes to maintain in the cluster.
`--maxComputeNodeNum` *maxNum*	Optional This parameter is applicable only for the AUTO scaling mode. If the cluster is in MANUAL mode or you are changing it to MANUAL mode, this parameter is ignored.	Upper limit of the range of active compute nodes to maintain in the cluster.
`--ioShares` *level*	Optional	Priority access level: LOW, NORMAL, or HIGH.

cluster start Command

The `cluster start` command lets you start a cluster in Serengeti.

Table 8-10.

Parameter	Mandatory/Optional	Description
`--name` *cluster_name*	Mandatory	Name of cluster to start.

cluster stop Command

The `cluster stop` command lets you stop a Hadoop cluster in Serengeti.

Table 8-11.

Parameter	Mandatory/Optional	Description
`--name` *cluster_name*	Mandatory	Name of cluster to stop.

cluster target Command

The `cluster target` command lets you connect to a cluster to run `fs`, `mr`, `pig`, and `hive` commands with the Serengeti CLI. You must rerun the *cluster target* command if it has been more than 30 minutes since you ran it in your current Serengeti Command-Line Interface session.

Table 8-12.

Parameter	Mandatory/Optional	Description
`--name` *cluster_name*	Optional	Name of the cluster to which to connect. If unspecified, the first cluster listed by the `cluster list` command is connected. The `--name` and `--info` parameters are mutually exclusive. You can use either parameter, but not both.
`--info`	Optional	Show targeted cluster information, such as the HDFS URL, Job Tracker URL and Hive server URL. The `--info` and `--name` parameters are mutually exclusive. You can use either parameter, but not both.

cluster upgrade Command

The `cluster upgrade` command lets you upgrade the components in each Big Data cluster's virtual machines created in a previous version of Big Data Extensions.

To enable the Serengeti Management Server to manage clusters created in a previous version of Big Data Extensions, you must upgrade the components in each cluster's virtual machines. The Serengeti Management Server uses these components to control the cluster nodes.

You can identify clusters you need to upgrade using the `cluster list` command. When you run the `cluster list` command the message "Earlier" displays where the cluster version normally appears. See "cluster list Command," on page 77.

Table 8-13.

Parameter	Mandatory/Optional	Description
`--name` *cluster_name_in_Serengeti*	Mandatory	Name of cluster to upgrade.

connect Command

The `connect` command lets you connect and log in to a remote Serengeti server.

The `connect` command reads the user name and password in interactive mode. You must run the connect command every time you begin a Serengeti Command-Line Interface session, and again after the 30 minute session timeout. If you do not run this command, you cannot run any other commands.

Table 8-14.

Parameter	Mandatory/Optional	Description
--host	Mandatory	Serengeti Web service URL, formatted as *serengeti_management_server_ip_or_host*:*port*. By default, the Serengeti web service is started at port 8443.

datastore Commands

The `datastore {*}` commands let you add and delete datastores, and view the list of datastores in a Serengeti deployment.

- datastore add Command on page 81

  The `datastore add` command lets you add a datastore to Serengeti.

- datastore delete Command on page 81

  The `datastore delete` command lets you delete a datastore from Serengeti.

- datastore list Command on page 81

  The `datastore list` command lets you view a list of datastores in Serengeti. If you do not specify a datastore name, all datastores are listed.

datastore add Command

The `datastore add` command lets you add a datastore to Serengeti.

Table 8-15.

Parameter	Mandatory/Optional	Description
--name *datastore_name_in_Serengeti*	Mandatory	Datastore name in Serengeti.
--spec *datastore_name_in_vCenter_Server*	Mandatory	Datastore name in vSphere. You can use a wildcard to specify multiple vmfs stores. Supported wildcards are * and ?.
--type **{LOCAL|SHARED}**	Optional	(Default=SHARED) Type of datastore: LOCAL or SHARED.

datastore delete Command

The `datastore delete` command lets you delete a datastore from Serengeti.

Table 8-16.

Parameter	Mandatory/Optional	Description
--name *datastore_name_in_Serengeti*	Mandatory	Name of datastore to delete.

datastore list Command

The `datastore list` command lets you view a list of datastores in Serengeti. If you do not specify a datastore name, all datastores are listed.

Table 8-17.

Parameter	Mandatory/Optional	Description
--name *Name_of_datastore_name_in_Serengeti*	Optional	Name of datastore to list.
--detail	Optional	List the datastore details, including the datastore path in vSphere.

disconnect Command

The `disconnect` command lets you disconnect and log out from a remote Serengeti server. After you disconnect from the server, you cannot run any Serengeti commands until you reconnect with the `connect` command.

There are no command parameters.

distro list Command

The `distro list` command lets you view the list of roles in a Hadoop distribution.

Table 8-18.

Parameter	Mandatory/Optional	Description
`--name` *distro_name*	Optional	Name of distribution to show.

fs Commands

The `fs {*}` FileSystem (FS) shell commands let you manage files on the HDFS and local systems. Before you can run an FS command in a Command-Line Interface session, or after the 30 minute session timeout, you must run the `cluster target` command

- fs cat Command on page 83

  The `fs cat` command lets you copy source paths to stdout.

- fs chgrp Command on page 84

  The `fs chgrp` command lets you change group associations of one or more files.

- fs chmod Command on page 84

  The `fs chmod` command lets you change permissions of one or more files.

- fs chown Command on page 84

  The `fs chown` command lets you change the owner of one or more files.

- fs copyFromLocal Command on page 84

  The `fs copyFromLocal` command lets you copy one or more source files from the local file system to the destination file system. The result of this command is the same as the `put` command.

- fs copyToLocal Command on page 85

  The `fs copyToLocal` command lets you copy one or more files to the local file system. The result of this command is the same as the `get` command.

- fs copyMergeToLocal Command on page 85

  The `fs copyMergeToLocal` command lets you concatenate one or more HDFS files to a local file.

- fs count Command on page 85

  The `fs count` command lets you count the number of directories, files, bytes, quota, and remaining quota.

- fs cp Command on page 85

  The `fs cp` command lets you copy one or more files from source to destination. This command allows multiple sources, in which case the destination must be a directory.

- fs du Command on page 86

  The `fs du` command lets you display the size of files and directories that are in the given directory, or if just a file is specified, the file size.

fs cat Command

The `fs cat` command lets you copy source paths to stdout.

Table 8-19.

Parameter	Mandatory/Optional	Description
file_name	Mandatory	File to be showed in the console. Multiple files must use quotes. For example: `"/path/file1 /path/file2"`.

fs chgrp Command

The `fs chgrp` command lets you change group associations of one or more files.

Table 8-20.

Parameter	Mandatory/Optional	Description
`--group` *group_name*	Mandatory	Group name of the file.
`--recursive` {true|false}	Optional	Perform the operation recursively.
file_name	Mandatory	Name of file to change. Multiple files must use quotes. For example: `"/path/file1 /path/file2"`.

fs chmod Command

The `fs chmod` command lets you change permissions of one or more files.

Table 8-21.

Parameter	Mandatory/Optional	Description
`--mode` `<permission mode>`	Mandatory	File permission mode. For example: 755.
`--recursive` {true|false}	Optional	Perform the operation recursively.
file_name	Mandatory	Name of file to change. Multiple files must use quotes. For example: `"/path/file1 /path/file2"/path/file1 /path/"`.

fs chown Command

The `fs chown` command lets you change the owner of one or more files.

Table 8-22.

Parameter	Mandatory/Optional	Description
`--owner` *permission_mode*	Mandatory	Name of file owner.
`--recursive` {true|false}	Optional	Change the owner recursively through the directory structure.
file_name	Mandatory	Name of file to change. Multiple files must use quotes. For example: `"/path/file1 /path/file2"`.

fs copyFromLocal Command

The `fs copyFromLocal` command lets you copy one or more source files from the local file system to the destination file system. The result of this command is the same as the `put` command.

Table 8-23.

Parameter	Mandatory/Optional	Description
--from *local_file_path*	Mandatory	Local file path. Multiple files must use quotes. For example: `"/path/file1 /path/file2"`.
--to *HDFS_file_path*	Mandatory	File path in local. If the `from` parameter value lists multiple files, the `to` parameter value is the directory name.

fs copyToLocal Command

The `fs copyToLocal` command lets you copy one or more files to the local file system. The result of this command is the same as the `get` command.

Table 8-24.

Parameter	Mandatory/Optional	Description
--from *HDFS_file_path*	Mandatory	File path in HDFS. Multiple files must use quotes. For example: `"/path/file1 /path/file"`.
--to *local_file_path*	Mandatory	File path in local. If the `from` parameter value lists multiple files, the `to` parameter value is the directory name.

fs copyMergeToLocal Command

The `fs copyMergeToLocal` command lets you concatenate one or more HDFS files to a local file.

Table 8-25.

Parameter	Mandatory/Optional	Description
--from *HDFS_file_path*	Mandatory	File path in HDFS. Multiple files must use quotes. For example: `"/path/file1 /path/file2"`.
--to *local_file_path*	Mandatory	File path in local.
--endline **`{true|false}`**	Optional	Add end of line (EOL) character.

fs count Command

The `fs count` command lets you count the number of directories, files, bytes, quota, and remaining quota.

Table 8-26.

Parameter	Mandatory/Optional	Description
--path *HDFS_path*	Mandatory	Path to be counted.
--quota **`{true|false}`**	Optional	Include quota information.

fs cp Command

The `fs cp` command lets you copy one or more files from source to destination. This command allows multiple sources, in which case the destination must be a directory.

Table 8-27.

Parameter	Mandatory/Optional	Description
--from *HDFS_source_file_path*	Mandatory	File path in local. Multiple files must use quotes. For example: `"/path/file1 /path/file2"`.
--to *HDFS_destination_file_path*	Mandatory	File path in local. If the `from` parameter value lists multiple files, the `to` parameter value is the directory name.

fs du Command

The `fs du` command lets you display the size of files and directories that are in the given directory, or if just a file is specified, the file size.

Table 8-28.

Parameter	Mandatory/Optional	Description
file_name	Mandatory	File to display. Multiple files must use quotes. For example: `"/path/file1 /path/file2"`.

fs expunge Command

The `fs expunge` command lets you empty the HDFS trash bin. There are no command parameters.

fs get Command

The `fs get` command lets you copy one or more HDFS files to the local file system.

Table 8-29.

Parameter	Mandatory/Optional	Description
--from *HDFS_file_path*	Mandatory	File path in HDFS. Multiple files must use quotes. For example: `"/path/file1 /path/file2"`.
--to *local_file_path*	Mandatory	File path in local. If the `from` parameter value lists multiple files, the `to` parameter value is the directory name.

fs ls Command

The `fs ls`command lets you view a list of a directory's files.

Table 8-30.

Parameter	Mandatory/Optional	Description
path_name	Mandatory	Path for which to view the list of files. Multiple paths must use quotes. For example: `"/path/file1 /path/file2"`.
--recursive **`{true|false}`**	Optional	Perform the operation recursively.

fs mkdir Command

The `fs mkdir` command lets you create a directory.

Table 8-31.

Parameter	Mandatory/Optional	Description
dir_name	Mandatory	Name of directory to create.

fs moveFromLocal Command

The `fs moveFromLocal` command copies files similarly to the `put` command, except that the local source file is deleted after it is copied.

Table 8-32.

Parameter	Mandatory/Optional	Description
--from *local_file_path*	Mandatory	File path in local. Multiple files must use quotes. For example: `"/path/file1 /path/file2"`.
--to *HDFS_file_path*	Mandatory	File path in HDFS. If the `from` parameter value lists multiple files, the `to` parameter value is the directory name.

fs mv Command

The `fs mv` command lets you move one or more local source files to an HDFS destination.

Table 8-33.

Parameter	Mandatory/Optional	Description
--from *source_file_path*	Mandatory	Local source path and file. Multiple files must use quotes. For example: `"/path/file1 /path/file2"`.
--to *HDFS_file_path*	Mandatory	HDFS destination path and file. If the `from` parameter value lists multiple files, the `to` parameter value is the directory name.

fs put Command

The `fs put` command lets you copy one or more local file system sources to an HDFS.

Table 8-34.

Parameter	Mandatory/Optional	Description
--from *local_file_path*	Mandatory	Local source path and file. Multiple files must use quotes. For example: `"/path/file1 /path/file2"`.
--to *HDFS_file_path*	Mandatory	HDFS destination path and file. If the `from` parameter value lists multiple files, the `to` parameter value is the directory name.

fs rm Command

The `fs rm` command lets you remove files from the HDFS.

Table 8-35.

Parameter	Mandatory/Optional	Description
file_path	Mandatory	File to remove.
--recursive **`{true|false}`**	Optional	Perform the operation recursively.
--skipTrash **`{true|false}`**	Optional	Bypass trash.

fs setrep Command

The `fs setrep` command lets you change a file's replication factor.

Table 8-36.

Parameter	Mandatory/Optional	Description
--path *file_path*	Mandatory	Path and file for which to change the replication factor.
--replica *replica_number*	Mandatory	Number of replicas.
--recursive **{true|false}**	Optional	Perform the operation recursively.
--waiting **{true|false}**	Optional	Wait for the replica number to equal the specified number.

fs tail Command

The `fs tail` command lets you display a file's last kilobyte of content to stdout.

Table 8-37.

Parameter	Mandatory/Optional	Description
file_path	Mandatory	File path to display.
`--file` {true|false}	Optional	Show content while the file grows.

fs text Command

The `fs text` command lets you output a source file in text format.

Table 8-38.

Parameter	Mandatory/Optional	Description
file_path	Mandatory	File path to display.

fs touchz Command

The `fs touchz` command lets you create a zero length file.

Table 8-39.

Parameter	Mandatory/Optional	Description
file_path	Mandatory	Name of file to create.

hive script Command

The `hive script` command lets you run a Hive or Hive Query Language (HQL) script.

Before you can run the `hive script` command in a Command-Line Interface session, or after the 30 minute session timeout, you must run the `cluster target` command.

Table 8-40.

Parameter	Mandatory/Optional	Description
--location *script_path*	Mandatory	Hive or HQL script file name.

mr Commands

The `mr {*}` commands let you manage MapReduce jobs.

Before you can run an `mr {*}` command in a Command-Line Interface session, or after the 30 minute session timeout, you must run the `cluster target` command

- mr jar Command on page 89

  The `mr jar` command lets you run a MapReduce job located inside the provided jar.

- mr job counter Command on page 90

  The `mr job counter` command lets you print the counter value of the MapReduce job.

- mr job events Command on page 90

  The `mr job events` command lets you print the events' details received by JobTracker for the given range.

- mr job history Command on page 90

  The `mr job history` command lets you print job details for failed and killed jobs.

- mr job kill Command on page 90

  The `mr job kill` command lets you kill a MapReduce job.

- mr job list Command on page 90

  The `mr job list` command lets you view the list of MapReduce jobs.

- mr job set priority Command on page 91

  The `mr job set priority` command lets you change a MapReduce job's priority.

- mr job status Command on page 91

  The `mr job status` command lets you query a MapReduce job's status.

- mr job submit Command on page 91

  The `mr job submit` command lets you submit a MapReduce job that is defined in a MapReduce job file.

- mr task fail Command on page 92

  The `mr task fail` command lets you fail the MapReduce task.

- mr task kill Command on page 92

  The `mr task kill` command lets you kill a MapReduce task.

mr jar Command

The `mr jar` command lets you run a MapReduce job located inside the provided jar.

Table 8-41.

Parameter	Mandatory/Optional	Description
--jarfile *jar_file_path*	Mandatory	JAR file path.
--mainclass *main_class_name*	Mandatory	Class that contains the `main()` method.
--args *arg*	Optional	Arguments to send to the `main` class. To send multiple arguments, double quote them.

mr job counter Command

The `mr job counter` command lets you print the counter value of the MapReduce job.

Table 8-42.

Parameter	Mandatory/Optional	Description
--jobid *job_id*	Mandatory	MR job id.
--groupname *group_name*	Mandatory	Counter's group name.
--countername *counter_name*	Mandatory	Counter's name.

mr job events Command

The `mr job events` command lets you print the events' details received by JobTracker for the given range.

Table 8-43.

Parameter	Mandatory/Optional	Description
--jobid *job_id*	Mandatory	MapReduce job id.
--from *from-event-#*	Mandatory	Event number of the first event to print.
--number *#-of-events*	Mandatory	Total number of events to print.

mr job history Command

The `mr job history` command lets you print job details for failed and killed jobs.

Table 8-44.

Parameter	Mandatory/Optional	Description
job_history_directory	Mandatory	Directory into which to place the job history files.
--all **{true|false}**	Optional	Print all jobs information.

mr job kill Command

The `mr job kill` command lets you kill a MapReduce job.

Table 8-45.

Parameter	Mandatory/Optional	Description
--jobid *job_id*	Mandatory	Job id of job to kill.

mr job list Command

The `mr job list` command lets you view the list of MapReduce jobs.

Table 8-46.

Parameter	Mandatory/Optional	Description
--all **{true|false}**	Optional	List all jobs.

mr job set priority Command

The `mr job set priority` command lets you change a MapReduce job's priority.

Table 8-47.

Parameter	Mandatory/Optional	Description
--jobid *jobid*	Mandatory	Job id of job to change.
--priority **{VERY_HIGH|HIGH|NORMAL|LOW|VERY_LOW}**	Mandatory	Job priority.

mr job status Command

The `mr job status` command lets you query a MapReduce job's status.

Table 8-48.

Parameter	Mandatory/Optional	Description
--jobid *jobid*	Mandatory	Job id of job to query.

mr job submit Command

The `mr job submit` command lets you submit a MapReduce job that is defined in a MapReduce job file.

Table 8-49.

Parameter	Mandatory/Optional	Description
--jobfile *jobfile*	Mandatory	File that defines the MapReduce job.

Sample MapReduce Job File

A MapReduce job file is a standard Hadoop configuration file.

```
<configuration>
<property>
  <name>fs.default.name</name>
  <value>hdfs://localhost:9000</value>
</property>
<property>
  <name>mapred.job.tracker</name>
  <value>localhost:9001</value>
</property>

<property>
  <name>mapred.jar</name>
  <value>/home/hadoop/hadoop-1.0.1/hadoop-examples-1.0.1.jar</value>
</property>
<property>
  <name>mapred.input.dir</name>
  <value>/user/hadoop/input</value>
</property>
<property>
  <name>mapred.output.dir</name>
  <value>/user/hadoop/output</value>
</property>
<property>
```

```
  <name>mapred.job.name</name>
  <value>wordcount</value>
</property>
<property>
  <name>mapreduce.map.class</name>
  <value>org.apache.hadoop.examples.WordCount.TokenizerMapper</value>
</property>
<property>
  <name>mapreduce.reduce.class</name>
  <value>org.apache.hadoop.examples.WordCount.IntSumReducer</value>
</property>
</configuration>
```

mr task fail Command

The `mr task fail` command lets you fail the MapReduce task.

Table 8-50.

Parameter	Mandatory/Optional	Description
--taskid *taskid*	Mandatory	Task id of task to fail.

mr task kill Command

The `mr task kill` command lets you kill a MapReduce task.

Table 8-51.

Parameter	Mandatory/Optional	Description
--taskid *taskid*	Mandatory	Task id of task to kill.

network Commands

The `network {*}` commands let you manage your networks.

- network add Command on page 93

  The `network add` command lets you add a network to Serengeti so that The network's IP addresses are available to clusters that you create.

- network delete Command on page 93

  The `network delete` command lets you delete a network from Serengeti. Deleting an unused network frees the network's IP addresses for use by other services.

- network list Command on page 93

  The `network list` command lets you view a list of available networks in Serengeti. The name, port group in vSphere, IP address assignment type, assigned IP address, and so on appear.

- network modify Command on page 94

  The `network modify` command lets you reconfigure a Serengeti static IP network by adding IP address segments to it. You might need to add IP address segments so that there is enough capacity for a cluster that you want to create.

network add Command

The `network add` command lets you add a network to Serengeti so that The network's IP addresses are available to clusters that you create.

NOTE If your network uses static IP addresses, be sure that the addresses are not occupied before you add the network.

This example adds a network with statically assigned IP addresses.

```
network add --name ipNetwork --ip 192.168.1.1-100,192.168.1.120-180 --portGroup pg1
--dns 202.112.0.1 --gateway 192.168.1.255 --mask 255.255.255.1
```

This example adds a network with DHCP assigned IP addresses.

```
network add --name dhcpNetwork --dhcp --portGroup pg1
```

Specify either the `--dhcp` parameter for dynamic addresses or the combination of parameters that are required for static addresses, but not parameters for both dynamic and static addresses.

Table 8-52.

Parameter	Mandatory/Optional	Description
`--name` *network_name_in_Serengeti*	Mandatory	Name of network resource to add.
`--portGroup` *port_group_name_in_vSphere*	Mandatory	Name of port group in vSphere to add.
`--dhcp`	Mandatory for dynamic addresses. Do not use for static addresses.	Assign dynamic DHCP IP addresses.
`--ip` *IP_range* `--dns` *dns_server_ip_addr* `--secondDNS` *dns_server_ip_addr* `--gateway` *gateway_IP_addr* `--mask` *network_IP_addr_mask*	Mandatory for static addresses. Do not use for dynamic addresses.	Assign static IP addresses. Express the *IP_range* in the format `xx.xx.xx.xx-xx[,xx]*` . Express IP addresses in the format `xx.xx.xx.xx`.

network delete Command

The `network delete` command lets you delete a network from Serengeti. Deleting an unused network frees the network's IP addresses for use by other services.

Table 8-53.

Parameter	Mandatory/Optional	Description
`--name` *network_name_in_Serengeti*	Mandatory	Delete the specified network in Serengeti.

network list Command

The `network list` command lets you view a list of available networks in Serengeti. The name, port group in vSphere, IP address assignment type, assigned IP address, and so on appear.

Table 8-54.

Parameter	Mandatory/Optional	Description
`--name` *network_name_in_Serengeti*	Optional	Name of network to display.
`--detail`	Optional	List network details.

network modify Command

The `network modify` command lets you reconfigure a Serengeti static IP network by adding IP address segments to it. You might need to add IP address segments so that there is enough capacity for a cluster that you want to create.

NOTE If your network uses static IP addresses, be sure that the addresses are not occupied before you add the network.

Table 8-55.

Parameter	Mandatory/Optional	Description
`--name` *network_name_in_Serengeti*	Mandatory	Modify the specified static IP network in Serengeti.
`--addIP` *IP_range*	Mandatory	IP address segments, in the format `xx.xx.xx.xx-xx[,xx]*`.

pig script Command

The `pig script` command lets you run a Pig or PigLatin script.

Before you can run the `pig script` command in a Command-Line Interface session, or after the 30 minute session timeout, you must run the `cluster target` command.

Table 8-56.

Parameter	Mandatory/Optional	Description
`--location` *script_path*	Mandatory	Name of the script to execute.

resourcepool Commands

The `resourcepool {*}` commands let you manage resource pools.

- resourcepool add Command on page 94

  The `resourcepool add` command lets you add a vSphere resource pool to Serengeti.

- resourcepool delete Command on page 95

  The `resourcepool delete` command lets you remove a resource pool from Serengeti.

- resourcepool list Command on page 95

  The `resourcepool list` command lets you view a list of Serengeti resource pools. If you do not specify a name, all Serengeti resource pools are listed.

resourcepool add Command

The `resourcepool add` command lets you add a vSphere resource pool to Serengeti.

When you add a resource pool in Serengeti, it represents the actual vSphere resource pool as recognized by vCenter Server. This symbolic representation enables you to use the Serengeti resource pool name, instead of the full path of the resource pool in vCenter Server, in cluster specification files.

Table 8-57.

Parameter	Mandatory/Optional	Description
`--name` *resource_pool_name_in_Serengeti*	Mandatory	Name of resource pool to add.
`--vccluster` *vSphere_cluster_of_the_resource_pool*	Optional	Name of vSphere cluster that contains the resource pool.
`--vcrp` *vSphere_resource_pool_name*	Mandatory	vSphere resource pool.

resourcepool delete Command

The `resourcepool delete` command lets you remove a resource pool from Serengeti.

Table 8-58.

Parameter	Mandatory/Optional	Description
`--name` *resource_pool_name_in_Serengeti*	Mandatory	Resource pool to remove.

resourcepool list Command

The `resourcepool list` command lets you view a list of Serengeti resource pools. If you do not specify a name, all Serengeti resource pools are listed.

Table 8-59.

Parameter	Mandatory/Optional	Description
--name *resource_pool_name_in_Serengeti*	Optional	Name and path of resource pool to list.
`--detail`	Optional	Include resource pool details.

topology Commands

The `topology {*}` commands let you manage cluster topology.

- topology list Command on page 95

  The `topology list` command lets you list the RACK-HOSTS mapping topology stored in Serengeti.

- topology upload Command on page 95

  The `topology upload` command lets you upload a rack-hosts mapping topology file to Serengeti. The uploaded file overwrites any previous file.

topology list Command

The `topology list` command lets you list the RACK-HOSTS mapping topology stored in Serengeti.

There are no command parameters.

topology upload Command

The `topology upload` command lets you upload a rack-hosts mapping topology file to Serengeti. The uploaded file overwrites any previous file.

The file format for every line is: `rackname: hostname1, hostname2…`

Table 8-60.

Parameter	Mandatory/Optional	Description
`--fileName` *topology_file_name*	Mandatory	Name of topology file.
`--yes`	Optional	Answer Y to Y/N confirmation. If not specified, manually type y or n.

Index