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
Contents

About This Book 7

Updated Information 9

1 Using the Serengeti Remote Command-Line Interface Client 11
   Access the Serengeti CLI By Using the Remote CLI Client 11
   Log in to Hadoop Nodes with the Serengeti Command-Line Interface Client 12

2 Managing Application Managers 15
   About Application Managers 15
   Add an Application Manager by Using the Serengeti Command-Line Interface 16
   View List of Application Managers by using the Serengeti Command-Line Interface 16
   Modify an Application Manager by Using the Serengeti Command-Line Interface 17
   View Supported Distributions for All Application Managers by Using the Serengeti Command-
   Line Interface 17
   View Configurations or Roles for Application Manager and Distribution by Using the Serengeti
   Command-Line Interface 17
   Delete an Application Manager by Using the Serengeti Command-Line Interface 18

3 Managing the Big Data Extensions Environment by Using the Serengeti
   Command-Line Interface 19
   About Application Managers 19
   Add a Resource Pool with the Serengeti Command-Line Interface 22
   Remove a Resource Pool with the Serengeti Command-Line Interface 23
   Add a Datastore with the Serengeti Command-Line Interface 23
   Remove a Datastore with the Serengeti Command-Line Interface 23
   Add a Network with the Serengeti Command-Line Interface 24
   Remove a Network with the Serengeti Command-Line Interface 24
   Reconfigure a Static IP Network with the Serengeti Command-Line Interface 25
   Reconfigure the DNS Type with the Serengeti Command-Line Interface 25
   Increase Cloning Performance and Resource Usage of Virtual Machines 26

4 Managing Users and User Accounts 29
   Create an LDAP Service Configuration File Using the Serengeti Command-Line Interface 29
   Activate Centralized User Management Using the Serengeti Command-Line Interface 31
   Create a Cluster With LDAP User Authentication Using the Serengeti Command-Line Interface 31
   Change User Management Modes Using the Serengeti Command-Line Interface 32
   Modify LDAP Configuration Using the Serengeti Command-Line Interface 33

5 Creating Hadoop and HBase Clusters 35
   About Hadoop and HBase Cluster Deployment Types 37
Default Hadoop Cluster Configuration for Serengeti  37
Default HBase Cluster Configuration for Serengeti  38
About Cluster Topology  38
About HBase Clusters  41
About MapReduce Clusters  48
About Data Compute Clusters  51
About Customized Clusters  62

6 Managing Hadoop and HBase Clusters  71
Stop and Start a Cluster with the Serengeti Command-Line Interface  71
Scale Out a Cluster with the Serengeti Command-Line Interface  72
Scale CPU and RAM with the Serengeti Command-Line Interface  72
Reconfigure a Cluster with the Serengeti Command-Line Interface  73
Delete a Cluster by Using the Serengeti Command-Line Interface  75
About vSphere High Availability and vSphere Fault Tolerance  75
Reconfigure a Node Group with the Serengeti Command-Line Interface  75
Expanding a Cluster with the Command-Line Interface  76
Recover from Disk Failure with the Serengeti Command-Line Interface Client  77
Recover a Cluster Node Virtual Machine  77
Enter Maintenance Mode to Perform Backup and Restore with the Serengeti Command-Line Interface Client  78
Perform Backup and Restore with the Serengeti Command-Line Interface Client  79

7 Monitoring the Big Data Extensions Environment  81
View List of Application Managers by using the Serengeti Command-Line Interface  81
View Available Hadoop Distributions with the Serengeti Command-Line Interface  82
View Supported Distributions for All Application Managers by Using the Serengeti Command-Line Interface  82
View Configurations or Roles for Application Manager and Distribution by Using the Serengeti Command-Line Interface  82
View Provisioned Clusters with the Serengeti Command-Line Interface  83
View Datastores with the Serengeti Command-Line Interface  83
View Networks with the Serengeti Command-Line Interface  83
View Resource Pools with the Serengeti Command-Line Interface  84

8 Cluster Specification Reference  85
Cluster Specification File Requirements  85
Cluster Definition Requirements  85
Annotated Cluster Specification File  86
Cluster Specification Attribute Definitions  89
White Listed and Black Listed Hadoop Attributes  92
Convert Hadoop XML Files to Serengeti JSON Files  94

9 Serengeti CLI Command Reference  95
appmanager Commands  95
cluster Commands  97
connect Command  104
datastore Commands  104
disconnect Command 105
distro list Command 105
mgmtvmcfg Commands 105
network Commands 106
resourcepool Commands 108
template Commands 109
topology Commands 109
usermgmt Commands 109

Index 111
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 VMware 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.
Updated Information

This *vSphere Big Data Extensions Command-Line Interface Guide* is updated with each release of the product or when necessary.

This table provides the update history of the *vSphere Big Data Extensions Command-Line Interface Guide*.

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN-001702-01</td>
<td>Added information on performing backup and restore operations. See “Perform Backup and Restore with the Serengeti Command-Line Interface Client,” on page 79</td>
</tr>
<tr>
<td>EN-001702-00</td>
<td>Initial release.</td>
</tr>
</tbody>
</table>
Using the Serengeti Remote Command-Line Interface Client

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

This chapter includes the following topics:
- “Access the Serengeti CLI By Using the Remote CLI Client,” on page 11
- “Log in to Hadoop Nodes with the Serengeti Command-Line Interface Client,” on page 12

Access the Serengeti CLI By Using the Remote CLI Client

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

Prerequisites

- Use the VMware vSphere Web Client to log in to the VMware 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. Download the Serengeti CLI package from the Serengeti Management Server.
   Open a Web browser and navigate to the following URL: https://server_ip_address/cli/VMware-Serengeti-CLI.zip

2. Download the ZIP file.
   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 or 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.

**Log in to Hadoop Nodes with the Serengeti Command-Line Interface Client**

To perform troubleshooting or to run your management automation scripts, log in to Hadoop master, worker, and client nodes with SSH from the Serengeti Management Server using SSH client tools such as SSH, PDSH, ClusterSSH, and Mussh, which do not require password authentication.

To connect to Hadoop cluster nodes over SSH, you can use a user name and password authenticated login. All deployed nodes are password-protected with either a random password or a user-specified password that was assigned when the cluster was created.

**Prerequisites**

Use the vSphere Web Client to log in to vCenter Server, and verify that the Serengeti Management Server virtual machine is running.
**Procedure**

1. Right-click the Serengeti Management Server virtual machine and select **Open Console**.
   
The password for the Serengeti Management Server appears.

   **Note**  If the password scrolls off the console screen, press Ctrl+D to return to the command prompt.

2. Use the vSphere Web Client to log in to the Hadoop node.
   
The password for the **root** user appears on the virtual machine console in the vSphere Web Client.

3. Change the password of the Hadoop node by running the `set-password -u` command.
   
   ```bash
   sudo /opt/serengeti/sbin/set-password -u
   ```
A key to managing your Hadoop clusters is understanding how to manage the different application managers that you use in your Big Data Extensions environment.

This chapter includes the following topics:

- “About Application Managers,” on page 15
- “Add an Application Manager by Using the Serengeti Command-Line Interface,” on page 16
- “View List of Application Managers by using the Serengeti Command-Line Interface,” on page 16
- “Modify an Application Manager by Using the Serengeti Command-Line Interface,” on page 17
- “View Supported Distributions for All Application Managers by Using the Serengeti Command-Line Interface,” on page 17
- “View Configurations or Roles for Application Manager and Distribution by Using the Serengeti Command-Line Interface,” on page 17
- “Delete an Application Manager by Using the Serengeti Command-Line Interface,” on page 18

### About Application Managers

You can use Cloudera Manager, Apache Ambari, and the default application manager to provision and manage clusters with VMware vSphere Big Data Extensions.

After you add a new Cloudera Manager or Ambari application manager to Big Data Extensions, you can redirect your software management tasks, including monitoring and managing clusters, to that application manager.

You can use an application manager to perform the following tasks:

- List all available vendor instances, supported distributions, and configurations or roles for a specific application manager and distribution.
- Create clusters.
- Monitor and manage services from the application manager console.

Check the documentation for your application manager for tool-specific requirements.

### Restrictions

The following restrictions apply to Cloudera Manager and Ambari application managers:

- To add an application manager with HTTPS, use the FQDN instead of the URL.
- You cannot rename a cluster that was created with a Cloudera Manager or Ambari application manager.
You cannot change services for a big data cluster from Big Data Extensions if the cluster was created with Ambari or Cloudera Manager application manager.

To change services, configurations, or both, you must make the changes from the application manager on the nodes.

If you install new services, Big Data Extensions starts and stops the new services together with old services.

If you use an application manager to change services and big data cluster configurations, those changes cannot be synced from Big Data Extensions. The nodes that you create with Big Data Extensions do not contain the new services or configurations.

Add an Application Manager by Using the Serengeti Command-Line Interface

To use either Cloudera Manager or Ambari application managers, you must add the application manager and add server information to Big Data Extensions.

**Note** If you want to add a Cloudera Manager or Ambari application manager with HTTPS, use the FQDN in place of the URL.

**Procedure**

1. Access the Serengeti CLI.
2. Run the `appmanager add` command.

   ```bash
   appmanager add --name application_manager_name --type [ClouderaManager|Ambari] --url http[s]://server:port
   ```

   Application manager names can include only alphanumeric characters ([0-9, a-z, A-Z]) and the following special characters; underscores, hyphens, and blank spaces.

   You can use the optional `description` variable to include a description of the application manager instance.
3. Enter your username and password at the prompt.
4. If you specified SSL, enter the file path of the SSL certificate at the prompt.

**What to do next**

To verify that the application manager was added successfully, run the `appmanager list` command.

View List of Application Managers by using the Serengeti Command-Line Interface

You can use the `appManager list` command to list the application managers that are installed on the Big Data Extensions environment.

**Prerequisites**

Verify that you are connected to an application manager.

**Procedure**

1. Access the Serengeti CLI.
2. Run the `appmanager list` command.

   ```bash
   appmanager list
   ```
The command returns a list of all application managers that are installed on the Big Data Extensions environment.

**Modify an Application Manager by Using the Serengeti Command-Line Interface**

You can modify the information for an application manager with the Serengeti CLI, for example, you can change the manager server IP address if it is not a static IP, or you can upgrade the administrator account.

**Prerequisites**

Verify that you have at least one external application manager installed on your Big Data Extensions environment.

**Procedure**

1. Access the Serengeti CLI.
2. Run the `appmanager modify` command.

    ```bash
    appmanager modify --name application_manager_name --url <http[s]://server:port>
    ```

    Additional parameters are available for this command. For more information about this command, see “appmanager modify Command,” on page 96.

**View Supported Distributions for All Application Managers by Using the Serengeti Command-Line Interface**

Supported distributions are those distributions that are supported by Big Data Extensions. Available distributions are those distributions that have been added into your Big Data Extensions environment. You can view a list of the Hadoop distributions that are supported in the Big Data Extensions environment to determine if a particular distribution is available for a particular application manager.

**Prerequisites**

Verify that you are connected to an application manager.

**Procedure**

1. Access the Serengeti CLI.
2. Run the `appmanager list` command.

    ```bash
    appmanager list --name application_manager_name [--distros]
    ```

    If you do not include the `--name` parameter, the command returns a list of all the Hadoop distributions that are supported on each of the application managers in the Big Data Extensions environment.

    The command returns a list of all distributions that are supported for the application manager of the name that you specify.

**View Configurations or Roles for Application Manager and Distribution by Using the Serengeti Command-Line Interface**

You can use the `appManager list` command to list the Hadoop configurations or roles for a specific application manager and distribution.

The configuration list includes those configurations that you can use to configure the cluster in the cluster specifications.
The role list contains the roles that you can use to create a cluster. You should not use unsupported roles to create clusters in the application manager.

**Prerequisites**
Verify that you are connected to an application manager.

**Procedure**
1. Access the Serengeti CLI.
2. Run the `appmanager list` command.
   ```bash
   appmanager list --name application_manager_name [--distro distro_name (--configurations | --roles)]
   ```
   The command returns a list of the Hadoop configurations or roles for a specific application manager and distribution.

**Delete an Application Manager by Using the Serengeti Command-Line Interface**
You can use the Serengeti CLI to delete an application manager when you no longer need it.

**Prerequisites**
- Verify that you have at least one external application manager installed on your Big Data Extensions environment.
- Verify that application manager you want to delete does not contain any clusters, or the deletion process will fail.

**Procedure**
1. Access the Serengeti CLI.
2. Run the `appmanager delete` command.
   ```bash
   appmanager delete --name application_manager_name
   ```
Managing the Big Data Extensions Environment by Using the Serengeti Command-Line Interface

You must manage your Big Data Extensions, which includes ensuring that if you choose not to add the resource pool, datastore, and network when you deploy the Serengeti vApp, you add the vSphere resources before you create a Hadoop or HBase cluster. You must also add additional application managers, if you want to use either Ambari or Cloudera Manager to manage your Hadoop clusters. You can remove resources that you no longer need.

This chapter includes the following topics:

- “About Application Managers,” on page 19
- “Add a Resource Pool with the Serengeti Command-Line Interface,” on page 22
- “Remove a Resource Pool with the Serengeti Command-Line Interface,” on page 23
- “Add a Datastore with the Serengeti Command-Line Interface,” on page 23
- “Remove a Datastore with the Serengeti Command-Line Interface,” on page 23
- “Add a Network with the Serengeti Command-Line Interface,” on page 24
- “Remove a Network with the Serengeti Command-Line Interface,” on page 24
- “Reconfigure a Static IP Network with the Serengeti Command-Line Interface,” on page 25
- “Reconfigure the DNS Type with the Serengeti Command-Line Interface,” on page 25
- “Increase Cloning Performance and Resource Usage of Virtual Machines,” on page 26

About Application Managers

You can use Cloudera Manager, Apache Ambari, and the default application manager to provision and manage clusters with VMware vSphere Big Data Extensions.

After you add a new Cloudera Manager or Ambari application manager to Big Data Extensions, you can redirect your software management tasks, including monitoring and managing clusters, to that application manager.

You can use an application manager to perform the following tasks:

- List all available vendor instances, supported distributions, and configurations or roles for a specific application manager and distribution.
- Create clusters.
- Monitor and manage services from the application manager console.

Check the documentation for your application manager for tool-specific requirements.
Restrictions

The following restrictions apply to Cloudera Manager and Ambari application managers:

- To add an application manager with HTTPS, use the FQDN instead of the URL.
- You cannot rename a cluster that was created with a Cloudera Manager or Ambari application manager.
- You cannot change services for a big data cluster from Big Data Extensions if the cluster was created with Ambari or Cloudera Manager application manager.
- To change services, configurations, or both, you must make the changes from the application manager on the nodes.
  If you install new services, Big Data Extensions starts and stops the new services together with old services.
- If you use an application manager to change services and big data cluster configurations, those changes cannot be synced from Big Data Extensions. The nodes that you create with Big Data Extensions do not contain the new services or configurations.

Add an Application Manager by Using the Serengeti Command-Line Interface

To use either Cloudera Manager or Ambari application managers, you must add the application manager and add server information to Big Data Extensions.

**Note** If you want to add a Cloudera Manager or Ambari application manager with HTTPS, use the FQDN in place of the URL.

**Procedure**

1. Access the Serengeti CLI.
2. Run the `appmanager add` command.

```
appmanager add --name application_manager_name --type [ClouderaManager|Ambari] --url http[s]://server:port
```

Application manager names can include only alphanumeric characters ([0-9, a-z, A-Z]) and the following special characters; underscores, hyphens, and blank spaces.

You can use the optional `description` variable to include a description of the application manager instance.

3. Enter your username and password at the prompt.

4. If you specified SSL, enter the file path of the SSL certificate at the prompt.

**What to do next**

To verify that the application manager was added successfully, run the `appmanager list` command.

Modify an Application Manager by Using the Serengeti Command-Line Interface

You can modify the information for an application manager with the Serengeti CLI, for example, you can change the manager server IP address if it is not a static IP, or you can upgrade the administrator account.

**Prerequisites**

Verify that you have at least one external application manager installed on your Big Data Extensions environment.
**Procedure**

1. Access the Serengeti CLI.
2. Run the `appmanager modify` command.
   ```
   appmanager modify --name application_manager_name
   --url <http[s]://server:port>
   ```
   Additional parameters are available for this command. For more information about this command, see “appmanager modify Command,” on page 96.

**View Supported Distributions for All Application Managers by Using the Serengeti Command-Line Interface**

Supported distributions are those distributions that are supported by Big Data Extensions. Available distributions are those distributions that have been added into your Big Data Extensions environment. You can view a list of the Hadoop distributions that are supported in the Big Data Extensions environment to determine if a particular distribution is available for a particular application manager.

**Prerequisites**

Verify that you are connected to an application manager.

**Procedure**

1. Access the Serengeti CLI.
2. Run the `appmanager list` command.
   ```
   appmanager list --name application_manager_name [--distros]
   ```
   If you do not include the `--name` parameter, the command returns a list of all the Hadoop distributions that are supported on each of the application managers in the Big Data Extensions environment.

   The command returns a list of all distributions that are supported for the application manager of the name that you specify.

**View Configurations or Roles for Application Manager and Distribution by Using the Serengeti Command-Line Interface**

You can use the `appManager list` command to list the Hadoop configurations or roles for a specific application manager and distribution.

The configuration list includes those configurations that you can use to configure the cluster in the cluster specifications.

The role list contains the roles that you can use to create a cluster. You should not use unsupported roles to create clusters in the application manager.

**Prerequisites**

Verify that you are connected to an application manager.

**Procedure**

1. Access the Serengeti CLI.
2. Run the `appmanager list` command.
   ```
   appmanager list --name application_manager_name [--distro distro_name
   (**--configurations | --roles**) ]
   ```
The command returns a list of the Hadoop configurations or roles for a specific application manager and distribution.

View List of Application Managers by using the Serengeti Command-Line Interface

You can use the `appManager list` command to list the application managers that are installed on the Big Data Extensions environment.

**Prerequisites**
Verify that you are connected to an application manager.

**Procedure**
1. Access the Serengeti CLI.
2. Run the `appManager list` command.

   ```
   appManager list
   ```

   The command returns a list of all application managers that are installed on the Big Data Extensions environment.

Delete an Application Manager by Using the Serengeti Command-Line Interface

You can use the Serengeti CLI to delete an application manager when you no longer need it.

**Prerequisites**
- Verify that you have at least one external application manager installed on your Big Data Extensions environment.
- Verify that application manager you want to delete does not contain any clusters, or the deletion process will fail.

**Procedure**
1. Access the Serengeti CLI.
2. Run the `appManager delete` command.

   ```
   appManager delete --name application_manager_name
   ```

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.

**Note** 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.

**Procedure**
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
```

**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**

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.

**Note** 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.

**Procedure**

1. Access the Serengeti CLI.
2. Run the `datastore add` command.

   This example adds a new, local storage datastore named `myLocalDS`. The value of the `--spec` parameter, `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 Big Data Extensions 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.

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.

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 `myNetwork` to the `10PG` vSphere port group. Virtual machines that use this network use DHCP to obtain the IP addresses.
     ```
     network add --name myNetwork --portGroup 10PG --dhcp
     ```
   - This example adds a network named `myNetwork` 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 myNetwork --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.xx.xx.xx-xx, single_ip, single_ip`
     ```
     xx.xx.xx.xx-xx, xx.xx.xx.xx-xx, single_ip, single_ip
     ```
   - This example adds a dynamic network with DHCP assigned IP addresses and meaningful host name.
     ```
     network add --name ddnsNetwork --dhcp --portGroup pg1 --dnsType DYNAMIC
     ```

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.

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`

Reconfigure the DNS Type with the Serengeti Command-Line Interface

You can reconfigure a network's Domain Name System (DNS) type, and specify that Big Data Extensions generate meaningful host names for the nodes in a Hadoop cluster.

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.
There are three DNS options you can specify:

**Normal**

The DNS server provides both forward and reverse FQDN to IP resolution. Reverse DNS is IP address to domain name mapping. The opposite of forward (normal) DNS which maps domain names to IP addresses. Normal is the default DNS type.

**Dynamic**

Dynamic DNS (DDNS or DynDNS) is a method of automatically updating a name server in the Domain Name System (DNS) with the active DNS configuration of its configured hostnames, addresses or other information. Big Data Extensions integrates with a Dynamic DNS server in its network through which it provides meaningful host names to the nodes in a Hadoop cluster. The cluster will then automatically register with the DNS server.

**Others**

There is no DNS server, or the DNS server doesn’t provide normal DNS resolution or Dynamic DNS services. In this case, you must add FQDN/IP mapping for all nodes in the /etc/hosts file for each node in the cluster. Through this mapping of hostnames to IP addresses each node can contact another node in the cluster.

Host names provide easier visual identification, as well as allowing you to use services such as Single Sign-On, which requires the use of a properly configured DNS.

**Procedure**

1. Access the Serengeti CLI.
2. Run the `network modify` command.

   There are three DNS types you can specify: NORMAL, DYNAMIC, and OTHERS. NORMAL is the default value.

   This example modifies a network named `myNetwork` to use a Dynamic DNS type. Virtual machines that use this network will use DHCP to obtain the IP addresses.

   ```
   network modify --name myNetwork --dnsType DYNAMIC
   ```

**Increase Cloning Performance and Resource Usage of Virtual Machines**

You can rapidly clone and deploy virtual machines using Instant Clone, a feature of vSphere 6.0.

Using Instant Clone, a parent virtual machine is forked, and then a child virtual machine (or instant clone) is created. The child virtual machine leverages the storage and memory of the parent, reducing resource usage.

When provisioning a cluster, Big Data Extensions creates a parent virtual machine for each host on which a cluster node has been placed. After provisioning a new resource pool labeled `BDE-ParentVMs-$serengeti.uuid-$template.name` is visible in vCenter Server. This resource pool contains several parent virtual machines. Normal cluster nodes are instantly cloned from these parent virtual machines. Once the parent virtual machines are created on the cluster hosts, the time required to provision and scale a cluster is significantly reduced.

When scaling a cluster the clone type you specified during cluster creation continues to be used, regardless of what the current clone type is. For example, if you create a cluster using instant clone, then change your Big Data Extensions clone type to fast clone, the cluster you provisioned using instant clone will continue to use instant clone to scale out the cluster.

If you create clusters and later want to make changes to the template virtual machine used to provision those clusters, you must first delete all the existing parent virtual machines before using the new template virtual machine. When you create clusters using the new template, Big Data Extensions creates new parent virtual machines based on the new template.
Prerequisites

Your Big Data Extensions deployment must use vSphere 6.0 to take advantage of Instant Clone.

Procedure

1. Log into the Serengeti Management Server.

2. Edit the /opt/serengeti/conf/serengeti.properties file and change the value of
   cluster.clone.service=fast.
   
   The default clone type when running vSphere 6.0 is Instant Clone.
   
   cluster.clone.service = instant

3. To enable Instant Clone, restart the Serengeti Management Server.

   sudo /sbin/service tomcat restart

   The Serengeti Management Server reads the revised serengeti.properties file and applies the Fast Clone feature to all new clusters you create.

What to do next

All clusters you create will now use Instant Clone to deploy virtual machines. See Chapter 5, “Creating Hadoop and HBase Clusters,” on page 35.
Managing Users and User Accounts

By default Big Data Extensions provides authentication only for local user accounts. If you want to use LDAP (either Active Directory or an OpenLDAP compatible directory) to authenticate users, you must configure Big Data Extensions for use with your LDAP or Active Directory service.

This chapter includes the following topics:

- “Create an LDAP Service Configuration File Using the Serengeti Command-Line Interface,” on page 29
- “Activate Centralized User Management Using the Serengeti Command-Line Interface,” on page 31
- “Create a Cluster With LDAP User Authentication Using the Serengeti Command-Line Interface,” on page 31
- “Change User Management Modes Using the Serengeti Command-Line Interface,” on page 32
- “Modify LDAP Configuration Using the Serengeti Command-Line Interface,” on page 33

Create an LDAP Service Configuration File Using the Serengeti Command-Line Interface

Create a configuration file that identifies your LDAP or Active Directory server environment.

**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 default distribution, 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. Navigate to a directory on the Serengeti Management Server where you want to create and store the configuration file.

You can use the directory `/opt/serengeti/etc` to store your configuration file.
3 Using a text editor, create a JavaScript Object Notation (JSON) file containing the configuration settings for your LDAP or Active Directory service.

The format of the configuration file is shown below.

```json
{
    "type": "user_mode_type",
    "primaryUrl": "ldap://AD_LDAP_server_IP_address:network_port",
    "baseUserDn": "DN_information",
    "baseGroupDn": "DN_information",
    "userName": "username",
    "password": "password",
    "mgmtVMUserGroupDn": "DN_information"
}
```

<table>
<thead>
<tr>
<th>Table 4-1. LDAP Connection Information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>type</strong></td>
</tr>
<tr>
<td><strong>baseUserDn</strong></td>
</tr>
<tr>
<td><strong>baseGroupDn</strong></td>
</tr>
<tr>
<td><strong>primaryUrl</strong></td>
</tr>
<tr>
<td><strong>mgmtVMUserGroupDn</strong></td>
</tr>
<tr>
<td><strong>userName</strong></td>
</tr>
<tr>
<td><strong>password</strong></td>
</tr>
</tbody>
</table>

4 When you complete the file, save your work.

**Example: Example LDAP Configuration File**

The following example illustrates the configuration file for an LDAP server within the acme.com domain.

```json
{
    "type": "LDAP",
    "primaryUrl": "ldap://acme.com:8888",
    "baseUserDn": "ou=users,dc=dev,dc=acme,dc=com",
    "baseGroupDn": "ou=users,dc=dev,dc=acme,dc=com",
    "userName": "jsmith",
    "password": "MyPassword",
    "mgmtVMUserGroupDn": "cn=Administrators,cn=Builtin,dc=dev,dc=acme,dc=com"
}
```

**What to do next**

With an LDAP configuration file created, you can now activate centralized user management for your Big Data Extensions environment. See “Activate Centralized User Management Using the Serengeti Command-Line Interface,” on page 31.
Activate Centralized User Management Using the Serengeti Command-Line Interface

You must specify that Big Data Extensions use an external user identity source before you can manage users through your LDAP or Active Directory.

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 default distribution, add one or more Hadoop distributions. See the VMware vSphere Big Data Extensions Administrator’s and User’s Guide.
- Create a configuration file identifying your LDAP or Active Directory environment for use with Big Data Extensions. See “Create an LDAP Service Configuration File Using the Serengeti Command-Line Interface,” on page 29

Procedure
1. Access the Serengeti CLI.
2. Run the command `usermgmtserver add --cfgfile config_file_path`
   This example activates centralized user management, specifying the file `/opt/serengeti/LDAPConfigFile.cfg` as the file containing your LDAP configuration settings.
   `usermgmtserver add --cfgfile /opt/serengeti/LDAPConfigFile.cfg`
3. Run the `mgmtvmcfg get` to verify successful configuration of your environment by printing out the LDAP or Active Directory configuration information.
   The contents of the active configuration file in use by your Big Data Extensions environment prints to the terminal.

What to do next
When you activate centralized user management, you can create clusters and assign user management to roles using the users and user groups defined by your LDAP or Active Directory service. See “Create a Cluster With LDAP User Authentication Using the Serengeti Command-Line Interface,” on page 31.

Create a Cluster With LDAP User Authentication Using the Serengeti Command-Line Interface

With centralized user management configured and activated, you can grant privileges to users and user groups in your LDAP or Active Directory service to individual Hadoop clusters that you create.

As an example of how you can use centralized user management in your Big Data Extensions environment, you can assign groups with administrative privileges in your LDAP or Active Directory service access to the Serengeti Management Server. This allows those users to administer Big Data Extensions and the Serengeti Management Server. You can then give another user group access to Hadoop cluster nodes, allowing them to run Hadoop jobs.

To access the Serengeti CLI and Serengeti commands, users must change to the user `serengeti` after they login. For example, you can use the command `su` to change to the `serengeti` user, after which you can access the Serengeti CLI.

`su serengeti`
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 default distribution, 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 value of the `--adminGroupName` parameter and `--userGroupName` parameter using the names of administrative groups and user groups to whom you want to grant privileges for the cluster you are creating.

   ```bash
   cluster create --name cluster_name --type hbase --adminGroupName AdminGroupName --userGroupName UserGroupName
   ```

What to do next

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

Change User Management Modes Using the Serengeti Command-Line Interface

You can change the user management mode of your Big Data Extensions environment. You can choose to use local user management, LDAP, or a combination of the two.

Big Data Extensions lets you authenticate local users, those managed by LDAP or Active Directory, or a combination of these authentication methods.

**Table 4-2. User Authentication Modes**

<table>
<thead>
<tr>
<th>User Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local</td>
<td>Specify LOCAL to create and manage users and groups that are stored locally in your Big Data Extensions environment. Local is the default user management solution.</td>
</tr>
<tr>
<td>LDAP user</td>
<td>Specify LDAP to create and manage users and groups that are stored in your organization’s identity source, such as Active Directory or LDAP. If you choose LDAP user you must configure Big Data Extensions to use an LDAP or Active Directory service (Active Directory as LDAP).</td>
</tr>
<tr>
<td>Mixed mode</td>
<td>Specify MIXED to use a combination of both local users and users stored in an external identity source. If you choose mixed mode you must configure Big Data Extensions to use an LDAP or Active Directory service (Active Directory as LDAP).</td>
</tr>
</tbody>
</table>

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 default distribution, 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 command `mgmtvmcfg modify` to specify the user authentication mode you want to use.

- Specify LOCAL to create and manage users and groups that are stored locally in your Big Data Extensions environment. LOCAL is the default user management solution when no Active Directory or LDAP service is available.
  
  `mgmtvmcfg modify LOCAL`

- Specify MIXED to use a combination of both local users and users stored in an external identity source. If you choose mixed mode you must configure Big Data Extensions to use an LDAP or Active Directory service.
  
  `mgmtvmcfg modify MIXED`

- Specify LDAP to create and manage users and groups that are stored in your organization's identity source, such as Active Directory as LDAP or LDAP. If you use LDAP you must configure Big Data Extensions to use an LDAP or Active Directory service.
  
  `mgmtvmcfg modify LDAP`

Big Data Extensions uses the user authentication mode you specify.

Modify LDAP Configuration Using the Serengeti Command-Line Interface

You can modify your LDAP settings and make those changes available to your Big Data Extensions environment.

You can populate changes you make to your LDAP configuration settings to Big Data Extensions. This lets you update your LDAP service information.

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 default distribution, add one or more Hadoop distributions. See the VMware vSphere Big Data Extensions Administrator's and User's Guide.
- Modify the LDAP configuration file to account for any changes you want to make to your user management settings. See “Create an LDAP Service Configuration File Using the Serengeti Command-Line Interface,” on page 29

Procedure

1 Access the Serengeti CLI.

2 Run the command `usermgmtserver modify --cfgfile config_file_path`

`usermgmtserver modify --cfgfile config_file_path`

Any changes you made to the LDAP configuration file are applied to your Big Data Extensions environment. Clusters you create will use the new LDAP settings.

What to do next

You can create clusters and assign user management roles using the users and user groups defined by your LDAP or Active Directory service. See “Create a Cluster With LDAP User Authentication Using the Serengeti Command-Line Interface,” on page 31.
Creating Hadoop and HBase Clusters

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

Restrictions

- When you create an HBase only cluster, you must use the default application manager because the other application managers do not support HBase only clusters.
- You cannot rename a cluster that was created with Cloudera Manager or Ambari application manager.
- Temporarily powering off hosts will cause Big Data clusters to fail during cluster creation.

When creating Big Data clusters, Big Data Extensions calculates virtual machine placement according to available resources, Hadoop best practices, and user defined placement policies prior to creating the virtual machines. When performing placement calculations, if some hosts are powered off or set to stand-by, either manually, or automatically by VMware Distributed Power Management (VMware DPM), those hosts will not be considered as available resources when Big Data Extensions calculates virtual machine placement for use with a Big Data cluster.

If a host is powered off or set to stand-by after Big Data Extensions calculates virtual machine placement, but before it creates the virtual machines, the cluster fails to create until you power on those hosts. The following workarounds can help you both prevent and recover from this issue.

- Disable VMware DPM on those vSphere clusters where you deploy and run Big Data Extensions.
- Put hosts in maintenance mode before you power them off.
- If a Big Data cluster fails to create due to its assigned hosts being temporarily unavailable, resume the cluster creation after you power-on the hosts.

Requirements

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 by using the Serengeti CLI 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.
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 by using the Serengeti CLI, at least 550 GB of storage and 55 GB of memory are recommended. For other Hadoop distributions, at least 350 GB of storage and 35 GB of memory are recommended.

**Caution** When you create a cluster with Big Data Extensions, Big Data Extensions disables the virtual machine automatic migration on the cluster. 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.

Passwords must be from 8 to 20 characters, use only visible lower ASCII characters (no spaces), and must contain at least one uppercase alphabetic character (A - Z), at least one lowercase alphabetic character (a - z), at least one digit (0 - 9), and at least one of the following special characters: \_ @ $ % ^ & *

This chapter includes the following topics:

- “About Hadoop and HBase Cluster Deployment Types,” on page 37
- “Default Hadoop Cluster Configuration for Serengeti,” on page 37
- “Default HBase Cluster Configuration for Serengeti,” on page 38
- “About Cluster Topology,” on page 38
- “About HBase Clusters,” on page 41
- “About MapReduce Clusters,” on page 48
- “About Data Compute Clusters,” on page 51
- “About Customized Clusters,” on page 62
About Hadoop and HBase Cluster Deployment Types

With Big Data Extensions, you can create and use several types of big data clusters.

**Basic Hadoop Cluster**
Simple Hadoop deployment for proof of concept projects and other small-scale data processing tasks. The Basic Hadoop cluster contains HDFS and the MapReduce framework. The MapReduce framework processes problems in parallel across huge datasets in the HDFS.

**HBase Cluster**
Runs on top of HDFS and provides a fault-tolerant way of storing large quantities of sparse data.

**Data and Compute Separation Cluster**
Separates the data and compute nodes, or clusters that contain compute nodes only. In this type of cluster, the data node and compute node are not on the same virtual machine.

**Compute Only Cluster**
You can create a cluster that contain only compute nodes, for example Jobtracker, Tasktracker, ResourceManager and NodeManager nodes, but not Namenode and Datanodes. A compute only cluster is used to run MapReduce jobs on an external HDFS cluster.

**Compute Workers Only Cluster**
Contains only compute worker nodes, for example, Tasktracker and NodeManager nodes, but not Namenodes and Datanodes. A compute workers only cluster is used to add more compute worker nodes to an existing Hadoop cluster.

**HBase Only Cluster**
Contains HBase Master, HBase RegionServer, and Zookeeper nodes, but not Namenodes or Datanodes. Multiple HBase only clusters can use the same external HDFS cluster.

**Customized Cluster**
Uses a cluster specification file to create clusters using the same configuration as your previously created clusters. You can edit the cluster specification file to customize the cluster configuration.

**Default Hadoop Cluster Configuration for Serengeti**

For basic Hadoop deployments, such as proof of concept projects, you can use the default Hadoop cluster configuration for Serengeti for clusters that are created with the CLI.

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 you use either Cloudera CDH4 or CDH5 Hadoop distributions, which support both MapReduce v1 and MapReduce v2 (YARN), the default Hadoop cluster configurations are different. The default hadoop cluster configuration for CDH4 is a MapReduce v1 cluster. The default hadoop cluster configuration for CDH5 is a MapReduce v2 cluster. All other distributions support either MapReduce v1 or MapReduce v2 (YARN), but not both.
Default HBase Cluster Configuration for Serengeti

HBase is an open source distributed columnar database that uses MapReduce and HDFS to manage data. You can use HBase 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 CLI, 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 HBase cluster deployed by Serengeti does not contain Hadoop JobTracker or Hadoop TaskTracker daemons. To run an HBase MapReduce job, deploy a customized, nondefault HBase cluster.

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 big data 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 virtual machine automatic migration of the cluster. 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 placement policy of the cluster, 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.

```json
{
    "nodeGroups": [
        ...
        {
            "name": "group_name",
            "placementPolicies": {
                "instancePerHost": 2,
                "groupRacks": {
                    "type": "ROUNDROBIN",
                    "racks": ["rack1", "rack2", "rack3"]
                }
            }
        }
    ]
}
```
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 default distribution, 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. (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 topologies, 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, Hortonworks HDP 2.x, or Pivotal PHD 1.1 or later cluster, you must configure a valid DNS and FQDN for the cluster's HDFS and MapReduce network traffic. If the DNS server cannot provide valid forward and reverse FQDN/IP resolution, 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
   ```

About HBase Clusters

HBase runs on top of HDFS and provides a fault-tolerant way of storing large quantities of sparse data.

Create a Default HBase Cluster by Using the Serengeti Command-Line Interface

You can use the Serengeti CLI to deploy HBase clusters on HDFS.

This task creates a default HBase cluster which does not contain the MapReduce framework. To run HBase MapReduce jobs, add Jobtracker and TaskTracker or ResourceManager and NodeManager nodes to the default HBase cluster sample specification file `/opt/serengeti/samples/default_hbase_cluster.json`, then create a cluster using this specification file.

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 default distribution, 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 value of the --type parameter 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 Only Cluster in Big Data Extensions

With Big Data Extensions, you can create an HBase only cluster, which contain only HBase Master, HBase RegionServer, and Zookeeper nodes, but not Namenodes and Datanodes. The advantage of having an HBase only cluster is that multiple HBase clusters can use the same external HDFS.

Procedure

1. **Prerequisites for Creating an HBase Only Cluster** on page 42
   Before you can create an HBase only cluster, you must verify that your system meets all of the prerequisites.

2. **Prepare the EMC Isilon OneFS as the External HDFS Cluster** on page 42
   If you use EMC Isilon OneFS as the external HDFS cluster to the HBase only cluster, you must create and configure users and user groups, and prepare your Isilon OneFS environment.

3. **Create an HBase Only Cluster by Using the Serengeti Command-Line Interface** on page 43
   You can use the Serengeti CLI to create an HBase only cluster.

Prerequisites for Creating an HBase Only Cluster

Before you can create an HBase only cluster, you must verify that your system meets all of the prerequisites.

**Prerequisites**

- Verify that you started the Serengeti vApp.
- Verify that you have more than one distribution if you want to use a distribution other than the default distribution.
- Verify that you have an existing HDFS cluster to use as the external HDFS cluster.
  To avoid conflicts between the HBase only cluster and the external HDFS cluster, the clusters should use the same Hadoop distribution and version.
- If the external HDFS cluster was not created using Big Data Extensions, verify that the HDFS directory /hadoop/hbase, the group hadoop, and the following users exist in the external HDFS cluster:
  - hdfs
  - hbase
  - serengeti
- If you use the EMC Isilon OneFS as the external HDFS cluster, verify that your Isilon environment is prepared.
  For information about how to prepare your environment, see “Prepare the EMC Isilon OneFS as the External HDFS Cluster,” on page 42.

Prepare the EMC Isilon OneFS as the External HDFS Cluster

If you use EMC Isilon OneFS as the external HDFS cluster to the HBase only cluster, you must create and configure users and user groups, and prepare your Isilon OneFS environment.

Procedure

1. Log in to one of the Isilon HDFS nodes as user root.
2. Create the users.
   - hdfs
   - hbase
Create the user group `hadoop`.

4 Create the directory `tmp` under the root HDFS directory.

5 Set the owner as `hdfs:hadoop` with the read and write permissions set as 777.

6 Create the directory `hadoop` under the root HDFS directory.

7 Set the owner as `hdfs:hadoop` with the read and write permissions set as 775.

8 Create the directory `hbase` under the directory `hadoop`.

9 Set the owner as `hbase:hadoop` with the read and write permissions set as 775.

10 Set the owner of the root HDFS directory as `hdfs:hadoop`.

### Example: Configuring the EMC Isilon OneFS Environment

```bash
isi auth users create --name="hdfs"
isi auth users create --name="hbase"
isi auth users create --name="serengeti"
isi auth groups create --name="hadoop"
pw useradd mapred -G wheel
pw useradd yarn -G wheel
chown hdfs:hadoop /ifs
mkdir /ifs/tmp
chmod 777 /ifs/tmp
chown hdfs:hadoop /ifs/tmp
mkdir -p /ifs/hadoop/hbase
chmod -R 775 /ifs/hadoop/hbase
chown hdfs:hadoop /ifs/hadoop/hbase
chown hbase:hadoop /ifs/hadoop/hbase
```

### What to do next

You are now ready to create the HBase only cluster with the EMC Isilon OneFS as the external cluster.

### Create an HBase Only Cluster by Using the Serengeti Command-Line Interface

You can use the Serengeti CLI to create an HBase only cluster.

You must use the default application manager because the other application managers do not support HBase only clusters.

#### Procedure

1 To define the characteristics of the new cluster, make a copy of the following cluster specification file: `/opt/serengeti/samples/hbase_only_cluster.json`

2 Replace `hdfs://hostname-of-namenode:8020` in the specification file with the namenode uniform resource identifier (URI) of the external HDFS cluster.

3 Access the Serengeti CLI.

4 Run the cluster `create` command.

   ```bash
   cluster create --name clusternamle --distro distroname
                  --specfile specfile_location
   ```
The `/opt/serengeti/samples/hbase_only_cluster.json` file is a sample specification file for HBase only clusters. It contains the zookeeper, hbase_master, and hbase_regionserver roles, but not the hadoop_namenode/hadoop_datanode role.

5 To verify that the cluster was created, run the `cluster list` command.

    cluster list --name name

After the cluster is created, the system returns `Cluster clustername created`.

### 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. You can configure vSphere HA protection for the Master roles.

**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 default distribution, add one or more Hadoop distributions. See the [VMware vSphere Big Data Extensions Administrator's and User's Guide](VMware-vSphere-Big-Data-Extensions-Command-Line-Interface-Guide).

**Procedure**

1. Create a cluster specification file to define the characteristics of the cluster, 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.

   ```json
   {
       "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": {
      // 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": {
    },
    "mapred-site.xml": {
    },
    "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": {
      // "text": "the full content of fair-scheduler.xml in one line"
    },
    "capacity-scheduler.xml": {
    }
  }
}
"mapred-queue-acls.xml": {
    // check for all settings at
    // "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

Create an HBase Only Cluster with External Namenode HA HDFS Cluster

You can create an HBase only cluster with two namenodes in an active-passive HA configuration. The HA namenode provides a hot standby name node that, in the event of a failure, can perform the role of the active namenode with no downtime.

- Worker only clusters are not supported on Ambari and Cloudera Manager application managers.
MapReduce v1 worker only clusters and HBase only clusters created using the MapR distribution are not supported.

**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 default distribution, add one or more Hadoop distributions. See the VMware vSphere Big Data Extensions Administrator’s and User’s Guide.

**Procedure**

1. To define the characteristics of the new cluster, make a copy of the following cluster specification file: `/opt/serengeti/samples/hbase_only_cluster.json`
2. Replace `hdfs://hostname-of-namenode:8020` in this spec file with the namenode uniform resource identifier (URI) of the external namenode HA HDFS cluster. The namenode URI is the value of the `fs.defaultFS` parameter in the `core-site.xml` of the external cluster.
3. Change the configuration section of the HBase only cluster specification file as shown in the following example. All the values can be found in `hdfs-site.xml` of the external cluster.

```json
"configuration": {
  "hadoop": {
    "hdfs-site.xml": {
      "dfs.nameservices": "dataMaster",
      "dfs.ha.namenodes.dataMaster": "namenode0,namenode1",
      "dfs.namenode.rpc-address.dataMaster.namenode0": "10.555.xx.xxx:xxx1",
      "dfs.namenode.http-address.dataMaster.namenode0": "10.555.xx.xxx:xxx2",
      "dfs.namenode.rpc-address.dataMaster.namenode1": "10.555.xx.xxx:xxx3",
      "dfs.namenode.http-address.dataMaster.namenode1": "10.555.xx.xxx:xxx4"
    }
  }
}
```

**About MapReduce Clusters**

MapReduce is a framework for processing problems in parallel across huge data sets. The MapReduce framework distributes a number of operations on the data set to each node in the network.

**Create a MapReduce v2 (YARN) Cluster by Using the Serengeti Command-Line Interface**

You can create MapReduce v2 (YARN) clusters if you want to create a cluster that separates the resource management and processing components.

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 default distribution, 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 MapReduce v2 cluster using the CDH4 distribution 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 MapReduce v1 Worker Only Cluster with External Namenode HA HDFS Cluster

You can create a MapReduce v1 worker only cluster with two namenodes in an active-passive HA configuration. The HA namenode provides a hot standby namenode that, in the event of a failure, can perform the role of the active namenode with no downtime.

The following restrictions apply to this task:

- Worker only clusters are not supported on Ambari and Cloudera Manager application managers.
- You cannot use MapR distribution to create MapReduce v1 worker only clusters and HBase only clusters

Prerequisites

- Start the Big Data Extensions vApp.
- Ensure that you have adequate resources allocated to run the Hadoop cluster.
- Ensure that you have an External Namenode HA HDFS cluster.
- To use any Hadoop distribution other than the default distribution, add one or more Hadoop distributions. See the VMware vSphere Big Data Extensions Administrator’s and User’s Guide.

Procedure

1. To define the characteristics of the new cluster, open the following cluster specification file to modify: /opt/serengeti/samples/compute_workers_only_mr1.json
2. Replace hdfs://hostname-of-namenode:8020 in this spec file with the namenode uniform resource identifier (URI) of the external namenode HA HDFS cluster. The namenode URI is the value of the fs.defaultFS parameter in the core-site.xml of the external cluster.
3. Replace the hostname-of-jobtracker in the specification file with the FQDN or IP address of the JobTracker in the external cluster.
4. Change the configuration section of the MapReduce Worker only cluster specification file as shown in the following example. All the values can be found in hdfs-site.xml of the external cluster.

```json
{
    "externalHDFS": "hdfs://dataMaster",
    "externalMapReduce": "xx.xxx.xxx.xxx:8021",
    "nodeGroups": [
        {
            "name": "worker",
            "roles": [
                "hadoop_tasktracker"
            ]
        }
    ]
}
```
Create a MapReduce v2 Worker Only Cluster with External Namenode HA HDFS Cluster

You can create a MapReduce v2 (Yarn) worker only cluster with two namenodes in an active-passive HA configuration. The HA namenode provides a hot standby namenode that, in the event of a failure, can perform the role of the active namenode with no downtime.

The following restrictions apply to this task:

- Worker only clusters are not supported on Ambari and Cloudera Manager application managers.
- You cannot use a MapR distribution to deploy MapReduce v1 worker only clusters and HBase only clusters.

**Prerequisites**

- Start the Big Data Extensions vApp.
- Ensure that you have an external Namenode HA HDFS cluster.
- Ensure that you have adequate resources allocated to run the Hadoop cluster.
- To use any Hadoop distribution other than the default distribution, add one or more Hadoop distributions. See the *VMware vSphere Big Data Extensions Administrator’s and User’s Guide*.

**Procedure**

1. To define the characteristics of the new cluster, open the following cluster specification file to modify: `/opt/serengeti/samples/compute_workers_only_yarn.json`

2. Replace `hdfs://hostname-of-namenode:8020` in this spec file with the namenode uniform resource identifier (URI) of the external namenode HA HDFS cluster. The namenode URI is the value of the `fs.defaultFS` parameter in the `core-site.xml` of the external cluster.

3. Replace the `hostname-of-resourcemanager` in the specification file with the FQDN or IP address of the ResourceManager in the external cluster.
Change the configuration section of the Yarn Worker only cluster specification file as shown in the following example. All the values can be found in hdfs-site.xml of the external cluster.

```json
{
  "externalHDFS": "hdfs://dataMaster",
  "externalMapReduce": "xx.xxx.xxx.xxx:8021",
  "nodeGroups": [
    {
      "name": "worker",
      "roles": [
        "hadoop_nodemanager"
      ],
      "instanceNum": 3,
      "cpuNum": 2,
      "memCapacityMB": 7500,
      "storage": {
        "type": "LOCAL",
        "sizeGB": 20
      }
    }
  ],
  "configuration": {
    "hadoop": {
      "hdfs-site.xml": {
        "dfs.nameservices": "dataMaster",
        "dfs.ha.namenodes.dataMaster": "namenode0,namenode1",
        "dfs.namenode.rpc-address.dataMaster.namenode0": "10.555.xx.xxx:xxx1",
        "dfs.namenode.http-address.dataMaster.namenode0": "10.555.xx.xxx:xxx2",
        "dfs.namenode.rpc-address.dataMaster.namenode1": "10.555.xx.xxx:xxx3",
        "dfs.namenode.http-address.dataMaster.namenode1": "10.555.xx.xxx:xxx4"
      }
    }
  }
}
```

### About Data Compute Clusters

You can separate the data and compute nodes in a Hadoop cluster, and you can control how nodes are placed on the vSphere ESXi hosts in your environment.

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.

Ambari and Cloudera Manager application managers do not support data-compute separation and compute-only clusters.
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 virtual machine automatic migration of the cluster. 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 placement policy of the cluster, 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**
- Start the Serengeti vApp.
- Ensure that you have adequate resources allocated to run the Hadoop cluster.
- To use any Hadoop distribution other than the default distribution, 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 characteristics of the cluster, including the node groups, topology, and placement constraints.

   - **Note** To create an Apache Bigtop, Cloudera CDH4 and CDH5, Hortonworks HDP 2.x, or Pivotal PHD 1.1 or later cluster, you must configure a valid DNS and FQDN for the cluster’s HDFS and MapReduce network traffic. If the DNS server cannot provide valid forward and reverse FQDN/IP resolution, 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.

```json
{
    "nodeGroups": [
    {
        "name": "master",
        "roles": [
            "hadoop_namenode",
            "hadoop_jobtracker"
        ],
        "instanceNum": 1,
```


```json
{
  "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
  }
}
```
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 default distribution, 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 characteristics of the cluster.

   ```json
   { "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,   }]   }
   ```

   **Note** To create an Apache Bigtop, Cloudera CDH4 and CDH5, Hortonworks HDP 2.x, or Pivotal PHD 1.1 or later cluster, you must configure a valid DNS and FQDN for the cluster's HDFS and MapReduce network traffic. If the DNS server cannot provide valid forward and reverse FQDN/IP resolution, 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.
"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 virtual machine automatic migration of the cluster. 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 placement policy of the cluster, 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 default distribution, 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 characteristics of the cluster, including the node groups and placement policy constraints.

**Note**: To create an Apache Bigtop, Cloudera CDH4 and CDH5, Hortonworks HDP 2.x, or Pivotal PHD 1.1 or later cluster, you must configure a valid DNS and FQDN for the cluster's HDFS and MapReduce network traffic. If the DNS server cannot provide valid forward and reverse FQDN/IP resolution, 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.

```json
{
    "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 Default Application Manager

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

**Note** To create an Apache Bigtop, Cloudera CDH4 and CDH5, Hortonworks HDP 2.x, or Pivotal PHD 1.1 or later cluster, you must configure a valid DNS and FQDN for the cluster’s HDFS and MapReduce network traffic. If the DNS server cannot provide valid forward and reverse FQDN/IP resolution, the cluster creation process might fail or the cluster is created but does not function.

**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 default distribution, 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 content to a new cluster specification file.

   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.

   ```json
   {
     "externalHDFS": "hdfs://hostname-of-namenode:8020",
     "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 cluster_name --distro distro_name --specFile path/spec_file_name
```

Create a Compute-Only Cluster with the Cloudera Manager Application Manager

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

You can use a Cloudera Manager application manager with any external HDFS.

If you use EMC Isilon OneFS as the external HDFS cluster to the HBase only cluster, you must create and configure users and user groups, and prepare your Isilon OneFS environment. See "Prepare the EMC Isilon OneFS as the External HDFS Cluster," on page 42

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 default distribution, 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 `yarn_compute_only_cluster.json` sample cluster specification file found in the directory `/opt/serengeti/samples/cloudera-manager` on the Serengeti server.

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

In this cluster specification file, the `default_fs_name` field points to an HDFS Namenode URI and the `webhdfs_url` field points to an HDFS web URL.

```json
{
    "nodeGroups": [
    {
        "name": "master",
        "roles": [
            "YARN_RESOURCE_MANAGER",
            "YARN_JOB_HISTORY"
        ],
        "instanceNum": 1,
        "cpuNum": 2,
        "memCapacityMB": 7500,
        "storage": {
            "type": "SHARED",
            "sizeGB": 50
        },
        "haFlag": "on",
        "configuration": { }
    },
    {
        "name": "worker",
        "roles": [
```
"YARN_NODE_MANAGER",
   "GATEWAY"
],
"instanceNum": 3,
"cpuNum": 2,
"memCapacityMB": 7500,
"storage": {
   "type": "LOCAL",
   "sizeGB": 50
},
"haFlag": "off",
"configuration": {
}
],
"configuration": {
   "ISILON": {
      // service level configurations
      // check for all settings by running "appmanager list --name <name> --configurations"
      "default_fs_name": "hdfs://FQDN:8020",
      "webhdfs_url": "hdfs://FQDN:8020/webhdfs/v1"
   },
   "YARN": {
      // service level configurations
   },
   "YARNRESOURCEMANAGER": {
   },
   "YARN_NODE_MANAGER": {
      "yarn_nodemanager_local_dirs": "/yarn/nm"
   }
}
}

3 Access the Serengeti CLI.
4 Run the `cluster create` command and include the cluster specification file parameter and associated filename.

   `cluster create --name computeOnlyCluster_name -- appManager appManager_name --distro distro_name --specFile path/spec_file_name`

---

**Create a Compute-Only Cluster with the Ambari Application Manager and Isilon**

You can create a compute-only cluster with the Ambari application manager by using Isilon OneFS. To create a compute-only cluster with Isilon OneFS you must enable Isilon SmartConnect (network load balancing).

To use EMC Isilon OneFS as the external HDFS cluster to the HBase-only cluster, you must create and configure users and user groups, and prepare your Isilon OneFS environment. See “Prepare the EMC Isilon OneFS as the External HDFS Cluster,” on page 42

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

Verify that the Hadoop distribution you wish to use is compatible with Isilon OneFS. Visit the EMC Web site and see Supported Hadoop Distributions in OneFS.

Procedure

1. Create a cluster specification file modeled on one of the following the sample cluster specification files, hdp_v2_1_yarn_compute_only_cluster.json or hdp_v2_2_yarn_compute_only_cluster.json which you can find in the directory /opt/serengeti/samples/ambari/ on the Serengeti server.

2. Enable Isilon SmartConnect.

   isi networks modify subnet --sc-service-addr=SmartConnect_IP --name=subnet_name
   isi networks modify pool --name=subnet_name:pool_name --sc-subnet=subnet_name --zone=zone_name

3. Specify the Ambari server and name node FQDN on your Isilon environment.

   isi zone zones modify System --hdfs-ambari-namenode=smart_connect_FQDN
   isi zone zones modify System --hdfs-ambari-server=ambari_server_FQDN

4. Edit the cluster specification file, /opt/serengeti/samples/ambari/hdp_v2_*_yarn_compute_only_cluster.json, and set externalNamenode to the Isilon SmartConnect FQDN. If the externalSecondaryNamenode attribute in the cluster specification file is set to the same value as the externalNamenode, remove the entry for the externalSecondaryNamenode.

5. Access the Serengeti CLI.

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

   cluster create --name computeOnlyCluster_name -- appManager appManager_name
   --distro distro_name --specFile path/spec_file_name

What to do next

Verify that your Ambari managed, compute-only cluster creates successfully, using the configuration necessary for your environment and usage.

Create a Compute Workers Only Cluster With Non-Namenode HA HDFS Cluster

If you already have a physical Hadoop cluster and want to do more CPU or memory intensive operations, you can increase the compute capacity by provisioning a worker only cluster. The worker only cluster is a part of the physical Hadoop cluster and can be scaled out elastically.

With the compute workers only clusters, you can “burst out to virtual.” It is a temporary operation that involves borrowing resources when you need them and then returning the resources when you no longer need them. With “burst out to virtual,” you spin up compute only workers nodes and add them to either an existing physical or virtual Hadoop cluster.

Restrictions

- Worker only clusters are not supported on Ambari and Cloudera Manager application managers.
- These options are not supported on compute workers only clusters.
  - --appmanager appmanager_name
  - --type cluster_type
Prerequisites

- Start the Big Data Extensions vApp.
- Ensure that you have an existing Hadoop cluster.
- Verify that you have the IP addresses of the NameNode and ResourceManager node.
- Ensure that you have adequate resources allocated to run the Hadoop cluster.
- To use any Hadoop distribution other than the default distribution, add one or more Hadoop distributions. See the VMware vSphere Big Data Extensions Administrator’s and User’s Guide.

Procedure

1. To define the characteristics of the new cluster, make a copy of the following cluster specification file:
   `/opt/serengeti/samples/compute_workers_only_mr1.json`
2. Replace `hdfs://hostname-of-namenode:8020` in the specification file with the namenode uniform resource identifier (URI) of the external HDFS cluster.
3. Replace the `hostname-of-jobtracker` in the specification file with the FQDN or IP address of the JobTracker in the external cluster.
4. Change the configuration section of the MapReduce Worker only cluster specification file. All the values can be found in `hdfs-site.xml` of the external cluster.

About Customized Clusters

You can use an existing cluster specification file to create clusters by using the same configuration as your previously created clusters. You can also edit the cluster specification file to customize the cluster configuration.

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 default distribution, 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.
   - 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 Bigtop, add one or more Hadoop distributions. See the VMware vSphere Big Data Extensions Administrator’s and User’s Guide.

```
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 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 and use the virtual machines to install Big Data applications.

The basic cluster does not install the Big Data application packages used when creating a cluster. Instead, you can install and manage Big Data applications with third party application management tools such as Ambari or Cloudera Manager within your Big Data Extensions environment, and integrate it with your Hadoop software. The basic cluster does not deploy a cluster. You must deploy software into the 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 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.
   
   ```json
   {
     "nodeGroups": [
       {
         "name": "master",
         "roles": [...
   ```
Access the Serengeti CLI.

2. 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.

Create a Cluster with an Application Manager by Using the Serengeti Command-Line Interface

You can use the Serengeti CLI to add a cluster with an application manager other than the default application manager. Then you can manage your cluster with the new application manager.

**Note** If you want to create a local yum repository, you must create the repository before you create the cluster.

**Prerequisites**

- Connect to an application manager.
- Ensure that you have adequate resources allocated to run the cluster. For information about resource requirements, see the documentation for your application manager.
- Verify that you have more than one distribution if you want to use a distribution other than the default distribution. See the *VMware vSphere Big Data Extensions Administrator’s and User’s Guide*. 
Procedure

1. Access the Serengeti CLI.
2. Run the `cluster` command.

   ```
   cluster create --name cluster_name --appManager appmanager_name
   --[localrepoURL local_repository_url]
   ```

   If you do not use the `appManager` parameter, the default application manager is used.

Create a Compute Workers Only Cluster by Using the vSphere Web Client

If you already have a physical Hadoop cluster and want to do more CPU or memory intensive operations, you can increase the compute capacity by provisioning a workers only cluster. The workers only cluster is a part of the physical Hadoop cluster and can be scaled out elastically.

With the compute workers only clusters, you can "burst out to virtual." It is a temporary operation that involves borrowing resources when you need them and then returning the resources when you no longer need them. With "burst out to virtual," you spin up compute only workers nodes and add them to either an existing physical or virtual Hadoop cluster.

Worker only clusters are not supported on Ambari and Cloudera Manager application managers.

Prerequisites

- Ensure that you have an existing Hadoop cluster.
- Verify that you have the IP addresses of the NameNode and ResourceManager node.

Procedure

1. Click **Create Big Data Cluster** on the objects pane.
2. In the Create Big Data Cluster wizard, choose the same distribution as the Hadoop cluster.
3. Set the DataMaster URL HDFS: `namenode ip or fqdn:8020`.
4. Set the ComputeMaster URL `nodeManager ip or fqdn`.
5. Follow the steps in the wizard and add the other resources.

   There will be three node managers in the cluster. The three new node managers are registered to the resource manager.

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 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 default distribution, 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.

```
ccluster create --name cluster_name --password
```

3. Enter your custom password, and enter it again.

Passwords must be from 8 to 20 characters, use only visible lowerASCII characters (no spaces), and must contain at least one uppercase alphabetic character (A - Z), at least one lowercase alphabetic character (a - z), at least one digit (0 - 9), and at least one of 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 is created with the default distribution, Apache Bigtop.

**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 default distribution, 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, Hortonworks HDP 2.x, or Pivotal PHD 1.1 or later cluster, you must configure a valid DNS and FQDN for the cluster's HDFS and MapReduce network traffic. If the DNS server cannot provide valid forward and reverse FQDN/IP resolution, 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:

```
ccluster create --name clusterName --distro cdh
```

This example creates a customized cluster named `mycdh` that uses the CDH5 Hadoop distribution, and is configured according to the `/opt/serengeti/samples/default_cdh5_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.

```
ccluster create --name mycdh --distro cdh5 --specFile /opt/serengeti/samples/default_cdh5_ha_hadoop_cluster.json
```
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 default distribution, 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, Hortonworks HDP 2.x, or Pivotal PHD 1.1 or later cluster, you must configure a valid DNS and FQDN for the cluster’s HDFS and MapReduce network traffic. If the DNS server cannot provide valid forward and reverse FQDN/IP resolution, 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 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 default distribution, 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 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 default distribution, 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, Hortonworks HDP 2.x, or Pivotal PHD 1.1 or later cluster, you must configure a valid DNS and FQDN for the cluster’s HDFS and MapReduce network traffic. If the DNS server cannot provide valid forward and reverse FQDN/IP resolution, 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": [
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 default distribution, 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.

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

   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, Hortonworks HDP 2.x, or Pivotal PHD 1.1 or later cluster, you must configure a valid DNS and FQDN for the cluster’s HDFS and MapReduce network traffic. If the DNS server cannot provide valid forward and reverse FQDN/IP resolution, the cluster creation process might fail or the cluster is created but does not function.
Managing Hadoop and HBase Clusters

You can use the vSphere Web Client to start and stop your big data cluster and modify the 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.

This chapter includes the following topics:

- “Stop and Start a Cluster with the Serengeti Command-Line Interface,” on page 71
- “Scale Out a Cluster with the Serengeti Command-Line Interface,” on page 72
- “Scale CPU and RAM with the Serengeti Command-Line Interface,” on page 72
- “Reconfigure a Cluster with the Serengeti Command-Line Interface,” on page 73
- “Delete a Cluster by Using the Serengeti Command-Line Interface,” on page 75
- “About vSphere High Availability and vSphere Fault Tolerance,” on page 75
- “Reconfigure a Node Group with the Serengeti Command-Line Interface,” on page 75
- “Expanding a Cluster with the Command-Line Interface,” on page 76
- “Recover from Disk Failure with the Serengeti Command-Line Interface Client,” on page 77
- “Recover a Cluster Node Virtual Machine,” on page 77
- “Enter Maintenance Mode to Perform Backup and Restore with the Serengeti Command-Line Interface Client,” on page 78
- “Perform Backup and Restore with the Serengeti Command-Line Interface Client,” on page 79

**Stop and Start a Cluster with the Serengeti Command-Line Interface**

You can stop a currently running cluster and start a stopped cluster from the Serengeti CLI. When you start or stop a cluster through Cloudera Manager or Ambari, only the services are started or stopped. However, when you start or stop a cluster through Big Data Extensions, not only the services, but also the virtual machines are started or stopped.

**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 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 nodes of a cluster, the changed password is not used for the new nodes that are created when you scale out a cluster. If you set the initial administrator password for the cluster when you created the cluster, that initial administrator password is used for the new nodes. If you did not set the initial administrator password for the cluster when you created the cluster, new random passwords are used for the new nodes.

**Prerequisites**
Ensure that the cluster is started.

Procedure
1. Access the Serengeti CLI.
2. Run the `cluster resize` command.
   ```
   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 the compute capacity and RAM of a cluster 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 CPU of a cluster 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 RAM of a cluster 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 the master node of a cluster because Serengeti powers down the virtual machine during the scaling process.

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

**Table 6-1. Maximum CPU and RAM Settings**

<table>
<thead>
<tr>
<th>Virtual Machine Version</th>
<th>Maximum Number of CPUs</th>
<th>Maximum RAM, in GB</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>8</td>
<td>255</td>
</tr>
<tr>
<td>8</td>
<td>32</td>
<td>1011</td>
</tr>
<tr>
<td>9</td>
<td>64</td>
<td>1011</td>
</tr>
<tr>
<td>10</td>
<td>64</td>
<td>1011</td>
</tr>
</tbody>
</table>
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`.

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

Reconfigure a Cluster with the Serengeti Command-Line Interface

You can reconfigure any big data 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`.

For details about the Serengeti JSON-formatted configuration file and associated attributes in Hadoop distribution files see the VMware vSphere Big Data Extensions Command-Line Interface Guide.

For details about the Serengeti JSON-formatted configuration file and associated attributes in Hadoop distribution files see Chapter 8, “Cluster Specification Reference,” on page 85.

**Note** Always use the `cluster config` command to change the parameters specified by the 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.

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cluster_name</code></td>
<td>Name of the cluster that you want to reconfigure.</td>
</tr>
<tr>
<td><code>file_path</code></td>
<td>The file system path to which to export the specification file.</td>
</tr>
<tr>
<td><code>cluster_spec_file_name</code></td>
<td>The name with which to label the exported cluster specification file.</td>
</tr>
</tbody>
</table>

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.

   ```json
   "configuration": {
     "hadoop": {
       "core-site.xml": {
         // note: any value (int, float, boolean, string) must be enclosed in double quotes
         and here is a sample:
   ```
3 (Optional) If the JAR files of your Hadoop distribution 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.

```json
"configuration": {
  "hadoop": {
    "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": {
        // "text": "the full content of fair-scheduler.xml in one line"
    },
    "capacity-scheduler.xml": {
    }
  }
}
```
"fair-scheduler.xml": { 
   
   
}

4 Access the Serengeti CLI.

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 configuration section of the cluster configuration file or comment out the attribute using double back slashes (//).
   
   b Re-run the `cluster config` command.

Delete a Cluster by Using the Serengeti Command-Line Interface

You can delete a cluster that you no longer need, regardless of whether it is running. When a 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`

**Expanding a Cluster with the Command-Line Interface**

You can expand an existing Big Data cluster by adding additional node groups.

**Procedure**

1. Access the Serengeti CLI.

2. Edit the cluster specification file to include the new node groups you want to add to the cluster.

When editing the cluster specification file to expand the cluster, keep the following items in mind:

- The new, expanded node groups must not use the same names as the existing node groups in the cluster.
- Ensure you use the correct syntax when editing the cluster specification file. Each element and its configuration value must be correct, or the expansion operation will fail.

This example illustrates an updated `nodeGroups` configuration from the larger cluster specification file.

```json
{
  "nodeGroups": [
    {
      "name": "master1",
      "roles": [
        "basic"
      ],
      "instanceNum": 1,
      "cpuNum": 2,
      "memCapacityMB": 3768,
      "storage": {
        "type": "SHARED",
        "sizeGB": 10
      },
      "haFlag": "on"
    },
    {
      "name": "worker1",
      "roles": [
        "basic"
      ],
      "instanceNum": 1,
      "cpuNum": 2,
      "memCapacityMB": 3768,
      "storage": {
        "type": "LOCAL",
        "sizeGB": 10
      },
      "haFlag": "off"
    }
  ]
}
```
3 Run the `cluster expand` command to apply the new cluster configuration with the expanded node groups.

```bash
cluster expand --name cluster_name --specFile path_name/spec_file_name
```

If the `cluster expand` operation fails, the cluster's status changes to `PROVISION_ERROR`. To recover from this condition, verify that the cluster specification file uses the correct syntax, and run the `cluster expand` command once again to recover from the failure.

**What to do next**

You can verify that the node groups were added to the cluster using the `cluster list` command. See “View Provisioned Clusters with the Serengeti Command-Line Interface,” on page 83.

---

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

**Client**

If there is a disk failure in a 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 JBOD). If several disks fail, the Hadoop data node might shutdown. Big Data Extensions enables 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 storage type and placement policies of the corresponding failed disk.

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 nodes of the cluster, the changed password is not used for the new nodes that are created by the disk recovery operation. If you set the initial administrator password of the cluster when you created the cluster, that initial administrator password is used for the new nodes. If you did not set the initial administrator password of the cluster 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.

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

---

### Recover a Cluster Node Virtual Machine

You can recover cluster node virtual machines that have become disassociated from their managed object identifier (MOID), or resource pool and virtual machine name.

In rare instances, the managed object identifier (MOID) of a cluster node virtual machine can change. This may occur when a host crashes and re-registers with vCenter Server. When BDE can not locate a node virtual machine in vCenter Server by its MOID, it first tries to locate the node by it resource pool and virtual machine name. If this it not possible, you can recover the cluster node virtual machine with the `cluster recover` command.

**Procedure**

1 Access the Serengeti CLI.
2 Run the cluster recover command to update the cluster, and recover the cluster node virtual machine.

cluster recover

What to do next
You can verify that the cluster node virtual machine was successfully recovered.

Enter Maintenance Mode to Perform Backup and Restore with the Serengeti Command-Line Interface Client

Before performing backup and restore operations, or other maintenance tasks, you must place Big Data Extensions into maintenance mode.

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 default distribution, add one or more Hadoop distributions. See the VMware vSphere Big Data Extensions Administrator's and User's Guide.

Procedure
1 Log into the Serengeti Management Server.
2 Run the script /opt/serengeti/sbin/serengeti-maintenance.sh to place Big Data Extensions into maintenance mode, or check maintenance status.

serengeti-maintenance.sh on | off | status

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>on</td>
<td>Turns on maintenance mode. Upon entering maintenance mode, Big Data Extensions continues executing jobs that have already been started, but will not respond to any new requests.</td>
</tr>
<tr>
<td>off</td>
<td>Turn off maintenance mode, and returns Big Data Extensions to its normal operating state.</td>
</tr>
<tr>
<td>status</td>
<td>Displays the maintenance status of Big Data Extensions.</td>
</tr>
<tr>
<td></td>
<td>A status of safe means it is safe to backup or perform other maintenance tasks on your Big Data Extensions deployment.</td>
</tr>
<tr>
<td></td>
<td>A status of off means maintenance mode has been turned off, and it is not safe to perform maintenance tasks such as backup and restore.</td>
</tr>
<tr>
<td></td>
<td>A status of on means Big Data Extensions has entered maintenance mode, but it is not yet safe to perform back and restore operations. You must wait until the system returns the safe status message.</td>
</tr>
</tbody>
</table>

To place your Big Data Extensions deployment into maintenance mode, run the serengeti-maintenance.sh script with the on option.

serengeti-maintenance.sh on

3 Verify that Big Data Extensions is in maintenance mode.
When Big Data Extensions completes all jobs that have been submitted, the maintenance status will enter safe mode. Run the serengeti-maintenance.sh with the status parameter repeatedly until it returns the safe system status message.

serengeti-maintenance.sh status
safe

4 Perform the necessary system maintenance tasks.
Perform Backup and Restore with the Serengeti Command-Line Interface Client

You can perform a backup and restore operation to recover your data from failures, or to otherwise restore a Big Data Extensions system that is in an inconsistent state. Additionally, backups of your Big Data Extensions environment is useful for routine administrative purposes, such as copying a database from one server to another or archiving.

**Prerequisites**

Before performing a backup and restore operation, you must enter maintenance mode. See “Enter Maintenance Mode to Perform Backup and Restore with the Serengeti Command-Line Interface Client,” on page 78.

**Procedure**

1. Log into the Serengeti Management Server.
2. Run the script /opt/serengeti/sbin/backup.sh to backup the data from your Big Data Extensions environment to a file that you specify.
   
   `/opt/serengeti/sbin/backup.sh backup_filename.tar.gz`
3. Copy the backup file to another Big Data Extensions environment.
4. With the target Big Data Extensions server running, run the script /opt/serengeti/sbin/restore.sh to restore the backup to the new system.
   
   `/opt/serengeti/sbin/restore.sh backup_filename.tar.gz`

When the restore procedure completes, the Big Data Extensions environment is ready for use with the newly restored data.
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 available in the vSphere Web Client.

This chapter includes the following topics:

- “View List of Application Managers by using the Serengeti Command-Line Interface,” on page 81
- “View Available Hadoop Distributions with the Serengeti Command-Line Interface,” on page 82
- “View Supported Distributions for All Application Managers by Using the Serengeti Command-Line Interface,” on page 82
- “View Configurations or Roles for Application Manager and Distribution by Using the Serengeti Command-Line Interface,” on page 82
- “View Provisioned Clusters with the Serengeti Command-Line Interface,” on page 83
- “View Datastores with the Serengeti Command-Line Interface,” on page 83
- “View Networks with the Serengeti Command-Line Interface,” on page 83
- “View Resource Pools with the Serengeti Command-Line Interface,” on page 84

View List of Application Managers by using the Serengeti Command-Line Interface

You can use the `appManager list` command to list the application managers that are installed on the Big Data Extensions environment.

**Prerequisites**

Verify that you are connected to an application manager.

**Procedure**

1. Access the Serengeti CLI.
2. Run the `appmanager list` command.

```
appmanager list
```

The command returns a list of all application managers that are installed on the Big Data Extensions environment.
View Available Hadoop Distributions with the Serengeti Command-Line Interface

Supported distributions are those distributions that are supported by Big Data Extensions. Available distributions are those distributions that have been added into your Big Data Extensions environment. You use the distro list command to 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 Supported Distributions for All Application Managers by Using the Serengeti Command-Line Interface

Supported distributions are those distributions that are supported by Big Data Extensions. Available distributions are those distributions that have been added into your Big Data Extensions environment. You can view a list of the Hadoop distributions that are supported in the Big Data Extensions environment to determine if a particular distribution is available for a particular application manager.

Prerequisites

Verify that you are connected to an application manager.

Procedure

1. Access the Serengeti CLI.
2. Run the appmanager list command.

   appmanager list --name application_manager_name [--distros]

   If you do not include the --name parameter, the command returns a list of all the Hadoop distributions that are supported on each of the application managers in the Big Data Extensions environment.

The command returns a list of all distributions that are supported for the application manager of the name that you specify.

View Configurations or Roles for Application Manager and Distribution by Using the Serengeti Command-Line Interface

You can use the appManager list command to list the Hadoop configurations or roles for a specific application manager and distribution.

The configuration list includes those configurations that you can use to configure the cluster in the cluster specifications.

The role list contains the roles that you can use to create a cluster. You should not use unsupported roles to create clusters in the application manager.
Prerequisites

Verify that you are connected to an application manager.

Procedure

1. Access the Serengeti CLI.
2. Run the `appmanager list` command.

   `appmanager list --name application_manager_name [--distro distro_name (--configurations | --roles)]`

   The command returns a list of the Hadoop configurations or roles for a specific application manager and distribution.

View Provisioned Clusters with the Serengeti Command-Line Interface

From the Serengeti CLI, you can list the provisioned 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.

   `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
   ```
To customize your clusters, you must know how to use Serengeti cluster specification files and define the cluster requirements with the various attributes and objects. After you create your configuration files you can convert them to JSON file format.

This chapter includes the following topics:

- “Cluster Specification File Requirements,” on page 85
- “Cluster Definition Requirements,” on page 85
- “Annotated Cluster Specification File,” on page 86
- “Cluster Specification Attribute Definitions,” on page 89
- “White Listed and Black Listed Hadoop Attributes,” on page 92
- “Convert Hadoop XML Files to Serengeti JSON Files,” on page 94

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.
- You can include only single-line comments using the pound sign (#) to identify the comment.

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 different Hadoop and HBase nodes and their resources for use by your Big Data cluster. You can use this annotated cluster specification file, and the sample files in /opt/serengeti/samples, as models to emulate when you create your Big Data clusters.

The following code is a typical cluster specification file. For code annotations, see Table 8-1.

```json
1 {
2   "nodeGroups" : [
3     {
4       "name": "master",
5       "roles": [
6         "hadoop_namenode",
7         "hadoop_resourcemanager"
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": ["rp1"
19      ],
20     },
21     {
22       "name": "data",
23       "roles": [
24         "hadoop_datanode"
25       ],
26       "instanceNum": 3,
27       "instanceType": "MEDIUM",
28       "cpuNum": 2,
29       "memCapacityMB": 2048,
30       "storage": {
31         "type": "LOCAL",
32         "sizeGB": 50,
33         "dsNames4Data": ["DSLOCALSSD"],
34         "dsNames4System": ["DSNDFS"
35      ]
36     },
37     "placementPolicies": {
38         "instancePerHost": 1,
39         "groupRacks": {
40           "type": "ROUNDROBIN",
41           "racks": ["rack1", "rack2", "rack3"
42      ]
43     },
44   },
45 }
46```
The cluster definition elements are defined in the table.

Table 8-1. Example Cluster Specification Annotation

<table>
<thead>
<tr>
<th>Line(s)</th>
<th>Attribute</th>
<th>Example Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>name</td>
<td>master</td>
<td>Node group name.</td>
</tr>
<tr>
<td>5-8</td>
<td>role</td>
<td>hadoop_namenode, hadoop_jobtracker</td>
<td>Node group role. hadoop_namenode and hadoop_jobtracker are deployed to the node group's virtual machine.</td>
</tr>
</tbody>
</table>
Table 8-1. Example Cluster Specification Annotation (Continued)

<table>
<thead>
<tr>
<th>Line(s)</th>
<th>Attribute</th>
<th>Example Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>instanceNum</td>
<td>1</td>
<td>Number of instances in the node group. Only one virtual machine is created for the group. 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.</td>
</tr>
<tr>
<td>10</td>
<td>instanceType</td>
<td>LARGE</td>
<td>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.</td>
</tr>
<tr>
<td>11</td>
<td>cpuNum</td>
<td>2</td>
<td>Number of CPUs per virtual machine. This attribute overrides the number of vCPUs in the predefined virtual machine specification.</td>
</tr>
<tr>
<td>12</td>
<td>memCapacityMB</td>
<td>4096</td>
<td>RAM size, in MB, per virtual machine. This attribute overrides the RAM size in the predefined virtual machine specification.</td>
</tr>
<tr>
<td>13-16</td>
<td>storage</td>
<td>See lines 14-15 for one group’s storage attributes</td>
<td>Node group storage requirements.</td>
</tr>
<tr>
<td>14</td>
<td>type</td>
<td>SHARED</td>
<td>Storage type. The node group is deployed using only shared storage.</td>
</tr>
<tr>
<td>15</td>
<td>sizeGB</td>
<td>20</td>
<td>Storage size. Each node in the node group is deployed with 20GB available disk space.</td>
</tr>
<tr>
<td>17</td>
<td>haFlag</td>
<td>on</td>
<td>HA protection for the node group. The node group is deployed with vSphere HA protection.</td>
</tr>
<tr>
<td>18-20</td>
<td>rpNames</td>
<td>rp1</td>
<td>Resource pools under which the node group virtual machines are deployed. These pools can be an array of values.</td>
</tr>
<tr>
<td>22-36</td>
<td>Node group definition for the data node</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>37-44</td>
<td>placementPolicies</td>
<td>See code sample</td>
<td>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.</td>
</tr>
</tbody>
</table>

VMware vSphere Big Data Extensions Command-Line Interface Guide 88 VMware, Inc.
Table 8-1. Example Cluster Specification Annotation (Continued)

<table>
<thead>
<tr>
<th>Line(s)</th>
<th>Attribute</th>
<th>Example Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>45-57</td>
<td>Node group definition for the compute node</td>
<td></td>
<td>See lines 4-16, which define the same attributes for the master node.</td>
</tr>
<tr>
<td>58-65</td>
<td>placementPolicies</td>
<td>See code sample</td>
<td>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.</td>
</tr>
<tr>
<td>66-82</td>
<td>Node group definition for the client node</td>
<td></td>
<td>See previous node group definitions.</td>
</tr>
<tr>
<td>83-86</td>
<td>configuration</td>
<td>Empty in the code sample</td>
<td>Hadoop configuration customization.</td>
</tr>
</tbody>
</table>

Cluster Specification Attribute Definitions

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

Cluster Specification Outer Attributes

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

Table 8-2. Cluster Specification Outer Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Mandatory/ Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nodeGroups</td>
<td>object</td>
<td>Mandatory</td>
<td>One or more group specifications. See Table 8-3.</td>
</tr>
<tr>
<td>configuration</td>
<td>object</td>
<td>Optional</td>
<td>Customizable Hadoop configuration key/value pairs.</td>
</tr>
<tr>
<td>externalHDFS</td>
<td>string</td>
<td>Optional</td>
<td>Valid only for compute-only clusters. URI of external HDFS.</td>
</tr>
</tbody>
</table>

Cluster Specification Node Group Objects and Attributes

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

Table 8-3. Cluster Specification’s Node Group Objects and Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Mandatory/ Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>string</td>
<td>Mandatory</td>
<td>User defined node group name.</td>
</tr>
<tr>
<td>roles</td>
<td>list of string</td>
<td>Mandatory</td>
<td>List of software packages or services to install on the virtual machine. Values must match the roles displayed by the distro list command.</td>
</tr>
<tr>
<td>Attribute</td>
<td>Type</td>
<td>Mandatory/Optional</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------</td>
<td>--------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>instanceNum</td>
<td>integer</td>
<td>Mandatory</td>
<td>Number of virtual machines in the node group:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Positive integer.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Generally, you can have multiple instances for hadoop_tasktracker, hadoop_datanode, hadoop_client, pig, and hive.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- For HDFS1 clusters, you can have only one instance of hadoop_namenode and hadoop_jobtracker.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- For HDFS2 clusters, you can have two hadoop_namenode instances.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- With a MapR distribution, you can configure multiple instances of hadoop_jobtracker.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>n Positive integer.</td>
</tr>
<tr>
<td>instanceType</td>
<td>string</td>
<td>Optional</td>
<td>Size of virtual machines in the node group, expressed as the name of a predefined virtual machine template. See Table 8-4.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- SMALL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- MEDIUM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- LARGE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- EXTRA_LARGE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If you specify cpuNum, memCapacityMB, or sizeGB attributes, they override the corresponding value of your selected virtual machine template for the applicable node group.</td>
</tr>
<tr>
<td>cpuNum</td>
<td>integer</td>
<td>Optional</td>
<td>Number of CPUs per virtual machine. If the haFlag value is FT, the cpuNum value must be 1.</td>
</tr>
<tr>
<td>memCapacityMB</td>
<td>integer</td>
<td>Optional</td>
<td>RAM size, in MB, per virtual machine.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note.</strong> When using MapR 3.1, you must specify a minimum of 5120 MBs of memory capacity for the zookeeper, worker, and client nodes.</td>
</tr>
<tr>
<td>swapRatio</td>
<td>float</td>
<td>Optional</td>
<td>Defines the ratio of OS Swap Disk Size and Memory Size. For example, if memory is 4GB (4096MB) and the swapRatio is 1. The swap disk will be 4GB. If you specify a swapRatio of 2, the swap disk will be 8GB. You can also specify a float for the swapRatio attribute. Specifying a value of 0.5 with if memory is 4GB of memory creates a swap disk of 2GB.</td>
</tr>
</tbody>
</table>
### Table 8-3. Cluster Specification’s Node Group Objects and Attributes (Continued)

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>latencySensitivity</td>
<td>string</td>
<td>Optional</td>
<td>You can specify LOW, NORMAL, MEDIUM, or HIGH, which defines the virtual machine's latency sensitivity setting within vCenter Server to optimize cluster performance. When deploying an HBase cluster you can optimize HBase performance by setting latencySensitivity to HIGH. You must then also set the reservedMemRatio parameter (see below) to 1.</td>
</tr>
<tr>
<td>reservedMemRatio</td>
<td>integer</td>
<td>Optional</td>
<td>You can specify 0 or 1 to define the ratio of reserved memory. When deploying an HBase cluster you can optimize HBase performance by setting parameter to 1. You must also set the latencySensitivity parameter (see above) to HIGH.</td>
</tr>
<tr>
<td>reservedCpuRatio</td>
<td>integer</td>
<td>Optional</td>
<td>You can specify 0 or 1 to define the ratio of reserved CPU.</td>
</tr>
<tr>
<td>Storage</td>
<td>object</td>
<td>Optional</td>
<td>Storage settings.</td>
</tr>
</tbody>
</table>
| type                  | string        | Optional           | Storage type:  
  - LOCAL. For local storage  
  - SHARED. For shared storage.                                                                                                                |
| sizeGB                | integer       | Optional           | Data storage size. Must be a positive integer.                                                                                               |
| diskNum               | integer       | Optional           | Specifies the number of disks to use for each node group.                                                                                     |
| 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 8-4. Serengeti Predefined Virtual Machine Sizes

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

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 8-5. Configuration Attribute White List

<table>
<thead>
<tr>
<th>File</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>core-site.xml</td>
<td>All core-default configuration attributes listed on the Apache Hadoop 2.x documentation Web page. For example, <a href="http://hadoop.apache.org/docs/branch_name/core-default.html">http://hadoop.apache.org/docs/branch_name/core-default.html</a>. Exclude the attributes defined in the black list.</td>
</tr>
<tr>
<td>hdfs-site.xml</td>
<td>All hdfs-default configuration attributes listed on the Apache Hadoop 2.x documentation Web page. For example, <a href="http://hadoop.apache.org/docs/branch_name/hdfs-default.html">http://hadoop.apache.org/docs/branch_name/hdfs-default.html</a>. Exclude the attributes defined in the black list.</td>
</tr>
<tr>
<td>mapred-site.xml</td>
<td>All mapred-default configuration attributes listed on the Apache Hadoop 2.x documentation Web page. For example, <a href="http://hadoop.apache.org/docs/branch_name/mapred-default.html">http://hadoop.apache.org/docs/branch_name/mapred-default.html</a>. Exclude the attributes defined in the black list.</td>
</tr>
<tr>
<td>hadoop-env.sh</td>
<td>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</td>
</tr>
</tbody>
</table>
### Table 8-5. Configuration Attribute White List (Continued)

<table>
<thead>
<tr>
<th>File</th>
<th>Attributes</th>
</tr>
</thead>
</table>
| 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 8-6. Configuration Attribute Black List

<table>
<thead>
<tr>
<th>File</th>
<th>Attributes</th>
</tr>
</thead>
</table>
| 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 |
| 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.
This section provides descriptions and syntax requirements for every Serengeti CLI command. This chapter includes the following topics:

- "appmanager Commands," on page 95
- "cluster Commands," on page 97
- "connect Command," on page 104
- "datastore Commands," on page 104
- "disconnect Command," on page 105
- "distro list Command," on page 105
- "mgmtvncfg Commands," on page 105
- "network Commands," on page 106
- "resourcepool Commands," on page 108
- "template Commands," on page 109
- "topology Commands," on page 109
- "usermgmt Commands," on page 109

**appmanager Commands**

The `appmanager` {*\} commands let you add, delete, and manage your application managers.

**appmanager add Command**

The `appmanager add` command lets you add an application manager other than the default to your environment. You can specify either Cloudera Manager or Ambari application manager. The `appmanager add` command reads the user name and password in interactive mode. If https is specified, the command prompts for the file path of the certificate.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--name application_manager_name</td>
<td>Mandatory</td>
<td>Application manager name</td>
</tr>
<tr>
<td>--description description</td>
<td>Optional</td>
<td></td>
</tr>
</tbody>
</table>
### appmanager delete Command

You can use the Serengeti CLI to delete an application manager when you no longer need it.

The application manager to delete must not contain clusters or the process fails.

```
appmanager delete --name application_manager_name
```

### appmanager modify Command

With the `appmanager modify` command, you can modify the information for an application manager, for example, you can change the manager server IP address if it is not a static IP, or you could upgrade the administrator account.

**Important** Making an error when you modify an application manager can have serious consequences. For example, you change a Cloudera Manager URL to the URL for a new application manager. If you create Big Data Extensions clusters with the old Cloudera Manager instance, the previous Cloudera Manager cluster cannot be managed again. In addition, the Cloudera Manager cluster is not available to the new application manager instance.

```
appmanager modify --name application_manager_name
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--type</code></td>
<td>Mandatory</td>
<td>Name of the type of application manager to use, either Cloudera Manager or Ambari</td>
</tr>
<tr>
<td><code>--url &lt;http[s]://server:port&gt;</code></td>
<td>Mandatory</td>
<td>Application manager service URL, formatted as http[s]://application_manager_server_ip_or_hostname:port, prompts for a login, username, and password.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory or Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--name application_manager_name</code></td>
<td>Mandatory</td>
<td>Application manager name</td>
</tr>
<tr>
<td><code>--url http[s]://server:port&gt;</code></td>
<td>Optional</td>
<td>Application manager service URL, formatted as http[s]://application_manager_server_ip_or_hostname:port, prompts for a login, username, and password. You can use either http or https.</td>
</tr>
<tr>
<td><code>--changeAccount</code></td>
<td>Optional</td>
<td>Changes the login account and password for the application manager.</td>
</tr>
<tr>
<td><code>--changeCertificate</code></td>
<td>Optional</td>
<td>Changes the SSL certificate of the application manager. This parameter only applies to application managers with a URL that starts with https.</td>
</tr>
</tbody>
</table>
**appmanager list Command**

The `appmanager list` command returns a list of all available application managers including the default application manager.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--name application_manager_name</code></td>
<td>Optional</td>
<td>The application manager name.</td>
</tr>
<tr>
<td><code>--distro distribution_name</code></td>
<td>Optional</td>
<td>The name of a specific distribution. If you do not include the <code>distribution_name</code> variable, the command returns all Hadoop distributions that are supported by the application manager.</td>
</tr>
<tr>
<td>`--configurations</td>
<td>--roles`</td>
<td>Optional</td>
</tr>
</tbody>
</table>

**cluster Commands**

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

**cluster config Command**

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

*Note* The `cluster config` command can only be used with clusters that were created with the default application manager. For those clusters that were created with either Ambari or Cloudera Manager, any cluster configuration changes should be made from the application manager. Also, new services and configurations changed in the external application manager cannot be synced from Big Data Extensions.

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 nodes in a cluster.

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

If the external HDFS cluster was created by Big Data Extensions, the user should use the `clusterconfig` command to add the HBase cluster topology to the HDFS cluster.

The following example depicts the specification file to add the topology:

```
"configuration" : { 
  "hadoop" : { 
    "topology.data": { "text": "10.1.1.1 /rack4,10.2.2.2 /rack4" } } 
}
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--name cluster_name_in_Serengeti</code></td>
<td>Mandatory</td>
<td>Name of Hadoop cluster to configure.</td>
</tr>
<tr>
<td><code>--specFile spec_file_path</code></td>
<td>Optional</td>
<td>File name of Hadoop cluster specification</td>
</tr>
<tr>
<td><code>--yes</code></td>
<td>Optional</td>
<td>Answer Y to Y/N confirmation. If not specified, manually type y or n.</td>
</tr>
<tr>
<td><code>--skipConfigValidation</code></td>
<td>Optional</td>
<td>Skip cluster configuration validation.</td>
</tr>
</tbody>
</table>
cluster create Command

You use the `cluster create` command to create a Hadoop or HBase cluster.

If the cluster specification does not include the required nodes, for example a master node, the Serengeti Management Server creates the cluster according to the default cluster configuration that Serengeti Management Server deploys.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory or Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--name cluster_name_in_Serengeti</code></td>
<td>Mandatory</td>
<td>Cluster name.</td>
</tr>
<tr>
<td><code>--networkName management_network_name</code></td>
<td>Mandatory</td>
<td>Network to use for management traffic in Hadoop clusters. If you omit any of the optional network parameters, the traffic associated with that parameter is routed on the management network that you specify with the <code>--networkName</code> parameter.</td>
</tr>
<tr>
<td><code>--adminGroupName admin_group_name</code></td>
<td>Optional</td>
<td>Administrative group to use for this cluster as defined in Active Directory or LDAP.</td>
</tr>
<tr>
<td><code>--userGroupName user_group_name</code></td>
<td>Optional</td>
<td>User group to use for this cluster as defined in Active Directory or LDAP.</td>
</tr>
<tr>
<td><code>--appmanager appmanager_name</code></td>
<td>Optional</td>
<td>Name of an application manager other than the default to manage your clusters.</td>
</tr>
</tbody>
</table>
| `--type cluster_type` | Optional              | Cluster type:  
  - Hadoop (Default)  
  - HBase |
| `--password` | Optional              | Custom password for all the nodes in the cluster.  
  Do not use if you use the `--resume` parameter.  
  Passwords must be from 8 to 20 characters, use only visible lowerASCII characters (no spaces), and must contain at least one uppercase alphabetic character (A - Z), at least one lowercase alphabetic character (a - z), at least one digit (0 - 9), and at least one of the following special characters: _ @ # $ % ^ & * |
| `--specFile spec_file_path` | Optional              | Cluster specification filename. For compute-only clusters, you must revise the spec file to point to an external HDFS. |
| `--distro Hadoop_distro_name` | Optional              | Hadoop distribution for the cluster. |
| `--dsNames datastore_names` | Optional              | Datastore to use to deploy Hadoop cluster in Serengeti. Multiple datastores can be used, separated by comma.  
  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. |
| `--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 comma. |
### cluster delete Command

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--name</code> cluster_name</td>
<td>Mandatory</td>
<td>Name of cluster to delete</td>
</tr>
<tr>
<td><code>--template</code> template_name</td>
<td>Optional</td>
<td>The template to use for clusters. If there is more than one template virtual machine, you must specify this parameter.</td>
</tr>
</tbody>
</table>

### cluster expand Command

The `cluster expand` command lets you expand and update the nodes in Big Data cluster.

You can expand an existing Big Data cluster with the `cluster expand` command. Edit the cluster’s specification file to include additional nodes and other available resources, and use the `cluster expand` command to apply the configuration to the existing cluster.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--name</code> cluster_name</td>
<td>Mandatory</td>
<td>Name of cluster to expand.</td>
</tr>
<tr>
<td><code>--specFile</code> spec_file_path</td>
<td>Mandatory</td>
<td>Cluster specification filename.</td>
</tr>
</tbody>
</table>
**cluster export Command**

The cluster export command lets you export cluster data. Depending on the options and parameters that you specify, you can export the cluster data to a specific location, format the delimiter of the export file, specify the type of data to export, and indicate the value for the topology.

You can use either of the following commands to export the cluster specification file.

- `cluster export --name cluster_name --specFile path_to_file`
  
  The use of the specfile parameter with the cluster export command is deprecated in Big Data Extensions 2.1.

- `cluster export --name cluster_name --type SPEC --output path_to_file`

You can use the cluster export command to print the IP to RACK mapping table. The format of the command is `ip rack`. The external HDFS cluster can use the cluster export command to implement the data location of the HBase and MapReduce cluster.

You can use the cluster export command to print the IP for the management network of all nodes in a cluster.

You can use cluster export command to print the IP to FQDN mapping table for all nodes in a cluster. You can chose to display the mapping table to the terminal, or export it to a file.

```
cluster export --name cluster_name --type IP2FQDN
cluster export --name cluster_name --type IP2FQDN --output path_to_file
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory or Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--name cluster_name</td>
<td>Mandatory</td>
<td>Name of cluster to export</td>
</tr>
<tr>
<td>--type SPEC</td>
<td>RACK</td>
<td>IP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SPEC, the default, to export a spec file.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RACK to export the rack topology of all nodes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IP to export the IP of all nodes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FQDN to export a cluster FQDN IP mapping of all nodes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IP2FQDN to export the IP to FQDN mapping table for all nodes in a cluster</td>
</tr>
<tr>
<td>--output path_to_output_file</td>
<td>Optional</td>
<td>Output file in which to save the exported data</td>
</tr>
<tr>
<td>--specfile path_to_spec_file</td>
<td>Optional</td>
<td>Output file in which to save the cluster specification.</td>
</tr>
<tr>
<td>--topology HOST_AS_RACK</td>
<td>RACK_AS_RACK</td>
<td>HVE</td>
</tr>
</tbody>
</table>
| --delimiter            | Optional             | Symbol or string to separate each line in the result. The default value is \
|                        |                      | n, line by line. |
cluster fix Command

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

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--name cluster_name</td>
<td>Mandatory</td>
<td>Name of cluster that has a failed disk.</td>
</tr>
<tr>
<td>--disk</td>
<td>Required</td>
<td>Recover node disks.</td>
</tr>
<tr>
<td>--nodeGroup nodegroup_name</td>
<td>Optional</td>
<td>Perform scan and recovery only on the specified node group, not on all the management nodes in the cluster.</td>
</tr>
</tbody>
</table>

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 information about each node group. The node group information consists of the instance count, CPU, memory, type, and size.

The application managers monitor the services and functions of your Big Data Extensions environment. Big Data Extensions syncs up the status from the application managers periodically. You can use the cluster list command to get the latest status of your environment. If there are any warnings displayed, you can check the details from the application manager console.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--name cluster_name_in_Serengeti</td>
<td>Optional</td>
<td>Name of cluster to list.</td>
</tr>
<tr>
<td>--detail</td>
<td>Optional</td>
<td>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.</td>
</tr>
</tbody>
</table>

cluster resetParam Command

The cluster resetParam command lets you reset the ioShares level for a cluster to default values.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--name cluster_name</td>
<td>Mandatory</td>
<td>Name of cluster for which to reset scaling parameters.</td>
</tr>
<tr>
<td>--ioShares</td>
<td>Optional</td>
<td>Reset to NORMAL.</td>
</tr>
</tbody>
</table>
**cluster resize Command**

The cluster resize command lets you change the number of nodes in a node group or scale the size of the up/down virtual machine’s CPU or RAM in a node group. When creating new nodes, the new created nodes will have the same services and configurations as the original nodes. When deleting nodes, Serengeti Management Server only allows tasktracker and nodemanager roles to be deleted. 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 nodes, the changed password is not used for the new nodes that are created by the cluster resize operation. If you set the initial administrator password when you created the cluster, that initial administrator password is used for the new nodes. If you did not set the initial administrator password when you created the cluster, new random passwords are used for the new nodes.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--name cluster_name</code></td>
<td>Mandatory</td>
<td>Target the Hadoop cluster that was deployed by Serengeti Management Server.</td>
</tr>
<tr>
<td><code>--nodeGroup name_of_the_node_group</code></td>
<td>Mandatory</td>
<td>Target node group to scale out in the cluster that was deployed by Serengeti Management Server.</td>
</tr>
<tr>
<td><code>--instanceNum instance_number</code></td>
<td>Optional</td>
<td>New instance number to scale to. If it is greater than the original count, Serengeti Management Server creates new nodes in the target node group. If it is less than the original count, Serengeti Management Server deletes nodes in the target node group. If the cluster resize operation fails, you can use the target instance number again to retry the cluster resize operation.</td>
</tr>
<tr>
<td><code>--cpuNumPerNode num_of_vCPUs</code></td>
<td>Optional</td>
<td>Number of vCPUs in a virtual machine in a target node group.</td>
</tr>
<tr>
<td><code>--force</code></td>
<td>Optional</td>
<td>When scaling out a cluster, you can overcome hardware or software failures using the <code>--force</code> parameter. Applying this parameter allows the cluster resize operation to proceed without being blocked by limited virtual machine failures.</td>
</tr>
<tr>
<td><code>--memCapacityMbPerNode size_in_MB</code></td>
<td>Optional</td>
<td>Memory size, in MB, of each virtual machine in a target node group.</td>
</tr>
<tr>
<td><code>--skipVcRefresh true</code></td>
<td>Optional</td>
<td>When performing cluster operations in a large vCenter Server environment, refreshing the inventory list may take considerable time. You can improve cluster resize performance using this parameter. <strong>Note</strong> If Serengeti Management Server shares the vCenter Server environment with other workloads, do not use this parameter. Serengeti Management Server cannot track the resource usage of other product’s workloads, and must refresh the inventory list in such circumstances.</td>
</tr>
</tbody>
</table>
cluster setParam Command

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--name cluster_name</td>
<td>Mandatory</td>
<td>Name of cluster for which to set elasticity parameters.</td>
</tr>
<tr>
<td>--ioShares level</td>
<td>Optional</td>
<td>Priority access level: LOW, NORMAL, or HIGH.</td>
</tr>
</tbody>
</table>

cluster start Command

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

Table 9-4.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--name cluster_name</td>
<td>Mandatory</td>
<td>Name of cluster to start.</td>
</tr>
<tr>
<td>--force</td>
<td>Optional</td>
<td>When starting a cluster, you can overcome hardware or software failures using the <code>--force</code> parameter. Applying this parameter allows the cluster start operation to proceed without being blocked by limited virtual machine failures.</td>
</tr>
</tbody>
</table>

cluster stop Command

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

Table 9-5.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--name cluster_name</td>
<td>Mandatory</td>
<td>Name of cluster to stop.</td>
</tr>
</tbody>
</table>

cluster update Command

The `cluster update` command lets you expand and update the resource pools and datastores available to your Big Data Extensions environment.

Table 9-6.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--name cluster_name</td>
<td>Mandatory</td>
<td>Name of cluster to update.</td>
</tr>
<tr>
<td>--rpNames resource_pool_name</td>
<td>Optional</td>
<td>Resource pools to use with your Hadoop clusters. You can specify multiple resource pools by separating each resource pool name with a comma. The resource pools you specify must include those resource pools already in use by the cluster.</td>
</tr>
<tr>
<td>--dsName datastore_names</td>
<td>Optional</td>
<td>Datastores to use with your Hadoop clusters. You can specify multiple datastores by separating each resource pool name with a comma. The datastores you specify must include those datastores already in use by the cluster.</td>
</tr>
</tbody>
</table>
Table 9-6. (Continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--append</td>
<td>Optional</td>
<td>This parameter appends the new resource pools and datastores you want to add to your environment to those already existing resources. This will not override your existing resources.</td>
</tr>
<tr>
<td>--yes</td>
<td>Optional</td>
<td>You can confirm whether or not to proceed with the specified action following an error message. If the responses are not specified, you can type Y or N. If you specify Y, cluster creation continues. If you do not specify Y, the CLI presents the following prompt after displaying the warning message: Are you sure you want to continue (Y/N) ?</td>
</tr>
</tbody>
</table>

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 9-7.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--host</td>
<td>Mandatory</td>
<td>Serengeti Web service URL, formatted as serengeti_management_server_ip_or_host:port. By default, the Serengeti web service is started at port 8443.</td>
</tr>
</tbody>
</table>

datastore Commands

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

datastore add Command

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

Table 9-8.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--name datastore_name_in_Serengeti</td>
<td>Mandatory</td>
<td>Datstore name in Serengeti.</td>
</tr>
<tr>
<td>--spec datastore_name_in_vCenter_Server</td>
<td>Mandatory</td>
<td>Datstore name in vSphere. You can use a wildcard to specify multiple vmfs stores. Supported wildcards are * and ?.</td>
</tr>
<tr>
<td>--type {LOCAL</td>
<td>SHARED}</td>
<td>Optional</td>
</tr>
</tbody>
</table>

datastore delete Command

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

Table 9-9.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--name datastore_name_in_Serengeti</td>
<td>Mandatory</td>
<td>Name of datastore to delete.</td>
</tr>
</tbody>
</table>
**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>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--name</code></td>
<td>Optional</td>
<td>Name of datastore to list.</td>
</tr>
<tr>
<td><code>--detail</code></td>
<td>Optional</td>
<td>List the datastore details, including the datastore path in vSphere.</td>
</tr>
</tbody>
</table>

**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>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--name</code></td>
<td>Optional</td>
<td>Name of distribution to show.</td>
</tr>
</tbody>
</table>

**mgmtvmcfg Commands**

The `mgmtvmcfg` commands let you view your current LDAP configuration settings, and change the user account management mode.

**mgmtvmcfg get Command**

You can list the current LDAP configuration using the `mgmtvmcfg get` command.

The `mgmtvmcfg get` command lists the current LDAP configuration settings in use by your Big Data Extensions environment.

`mgmtvmcfg get`

<table>
<thead>
<tr>
<th>type</th>
<th>The external user authentication service to use, which is either AD_AS_LDAP or LDAP.</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseUserDn</td>
<td>The base user DN.</td>
</tr>
<tr>
<td>baseGroupDn</td>
<td>The base group DN.</td>
</tr>
<tr>
<td>primaryUrl</td>
<td>The primary server URL of your Active Directory or LDAP server.</td>
</tr>
<tr>
<td>mgmtVMUserGroupDn</td>
<td>(Optional) The base DN for searching groups to access the Serengeti Management Server.</td>
</tr>
<tr>
<td>userName</td>
<td>The username of the Active Directory or LDAP server administrator account.</td>
</tr>
<tr>
<td>password</td>
<td>The password of the Active Directory or LDAP server administrator account.</td>
</tr>
</tbody>
</table>
**mgmtvmcfg modify Command**

The `mgmtvmcfg modify` command let you specify different user account management modes. You can choose to use local user management, LDAP, or a combination of the two.

You can use one of three user authentication modes: LOCAL, LDAP or MIXED.

- Specify MIXED to use a combination of both local users and users stored in an external identity source. If you choose mixed mode you must configure Big Data Extensions to use an LDAP or Active Directory service.

  ```
  mgmtvmcfg modify --usermgmtmode MIXED
  ```

- Specify LDAP to create and manage users and groups that are stored in your organization’s identity source, such as Active Directory as LDAP or LDAP. If you use LDAP you must configure Big Data Extensions to use an LDAP or Active Directory service.

  ```
  mgmtvmcfg modify --usermgmtmode LDAP
  ```

- Specify LOCAL to create and manage users and groups that are stored locally in your Big Data Extensions environment. LOCAL is the default user management solution when no Active Directory or LDAP service is available.

  ```
  mgmtvmcfg modify --usermgmtmode LOCAL
  ```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory or Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--usermgmtmode LOCAL</td>
<td>Optional</td>
<td>Specify LOCAL, MIXED, or LDAP user management modes.</td>
</tr>
<tr>
<td>--usermgmtmode MIXED</td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>--usermgmtmode LDAP</td>
<td>Optional</td>
<td></td>
</tr>
</tbody>
</table>

**network Commands**

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

**network add Command**

The `network add` command lets you add a network to Serengeti so that the IP addresses of the network 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
```

This example adds a dynamic network with DHCP assigned IP addresses and generates meaningful host names for the nodes in a Hadoop cluster.

```
network add --name ddnsNetwork --dhcp --portGroup pg1 --dnsType DYNAMIC
```

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 9-13.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--name network_name_in_Serengeti</td>
<td>Mandatory</td>
<td>Name of network resource to add.</td>
</tr>
<tr>
<td>--portGroup port_group_name_in_vSphere</td>
<td>Mandatory</td>
<td>Name of port group in vSphere to add.</td>
</tr>
<tr>
<td>--dhcp</td>
<td>Mandatory for dynamic addresses. Do not use for static addresses.</td>
<td>Assign dynamic DHCP IP addresses.</td>
</tr>
<tr>
<td>--ip IP_range</td>
<td>Optional</td>
<td>Assign static IP addresses.</td>
</tr>
<tr>
<td>--dns dns_server_ip_addr</td>
<td>Optional</td>
<td>Express the IP_range in the format xx.xx.xx.xx-xx[,]*.</td>
</tr>
<tr>
<td>--secondDNS dns_server_ip_addr</td>
<td>Optional</td>
<td>Express IP addresses in the format xx.xx.xx.xx.</td>
</tr>
<tr>
<td>--gateway gateway_IP_addr</td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>--mask network_IP_addr_mask</td>
<td>Optional</td>
<td>Possible values include NORMAL, DYNAMIC, and OTHERS. The default value is NORMAL.</td>
</tr>
</tbody>
</table>

network delete Command

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--name network_name_in_Serengeti</td>
<td>Mandatory</td>
<td>Delete the specified network in Serengeti.</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--name network_name_in_Serengeti</td>
<td>Optional</td>
<td>Name of network to display.</td>
</tr>
<tr>
<td>--detail</td>
<td>Optional</td>
<td>List network details.</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--name network_name_in_Serengeti</td>
<td>Mandatory</td>
<td>Modify the specified static IP network in Serengeti.</td>
</tr>
<tr>
<td>--addIP IP_range</td>
<td>Optional</td>
<td>IP address segments, in the format xx.xx.xx.xx-xx[,]*.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Mandatory/Optional</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--dnsType</td>
<td>Optional</td>
<td>Possible values include Normal, Dynamic, and Other. The default is Normal.</td>
</tr>
<tr>
<td>--generateHostname</td>
<td>Optional</td>
<td>The default is False.</td>
</tr>
</tbody>
</table>

**resourcepool Commands**

The `resourcepool` (*) commands let you manage resource pools.

**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.

**resourcepool delete Command**

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

**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.
**template Commands**

The template commands let you manage node templates in your Serengeti environment.

**template list Command**

The template list command lists the available node templates in your Serengeti environment.

There are no command parameters.

**topology Commands**

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

**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 9-17.**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--fileName</td>
<td>Mandatory</td>
<td>Name of topology file.</td>
</tr>
<tr>
<td>--yes</td>
<td>Optional</td>
<td>Answer Y to Y/N confirmation. If not specified, manually type y or n.</td>
</tr>
</tbody>
</table>

**usermgmt Commands**

The usermgmtserver commands let you manage users and groups in an Active Directory or LDAP server for user authentication, allowing you to manage users from a central point.

**usermgmtserver add Command**

You activate centralized account management for use with LDAP user authentication using the usermgmtserver add command.

To activate centralized account management for use with Big Data Extensions using LDAP (using either Active Directory as LDAP or OpenLDAP), run the usermgmtserver add command. This example activates centralized account management using the LDAP configuration settings contained in the file /opt/serengeti/sbin/LDAPConfigFile.cfg

usermgmtserver add --cfgfile /opt/serengeti/sbin/LDAPConfigFile.cfg
**Parameter** | **Mandatory or Optional** | **Description**
--- | --- | ---
`add --cfgfile config_file_path` | Optional | Activates centralized account management. The configuration file you specify contains the LDAP server configuration settings. You must create this file manually. See “Create an LDAP Service Configuration File Using the Serengeti Command-Line Interface,” on page 29

**usermgmtserver modify Command**

You can modify your LDAP configuration and make those changes available to your Big Data Extensions environment.

You can populate changes you make to your LDAP configuration settings to Big Data Extensions. This lets you update your LDAP service information.

To modify the LDAP settings for Big Data Extensions environment, update the LDAP configuration file as needed, and run the `usermgmtserver modify` command. This example updates your environment's LDAP configuration using the settings contained in the file `/opt/serengeti/sbin/LDAPConfigFile.cfg`

`usermgmtserver modify --cfgfile /opt/serengeti/sbin/LDAPConfigFile.cfg`

**Parameter** | **Mandatory or Optional** | **Description**
--- | --- | ---
`modify --cfgfile config_file` | Optional | If you update the LDAP configuration file to use new settings, you must run the `modify` parameter to update the settings within Big Data Extensions. You must update the configuration file manually. See “Create an LDAP Service Configuration File Using the Serengeti Command-Line Interface,” on page 29
Index

A
accessing, Command-Line Interface 11
activate 31
Active Directory
configuration file 29
managing users 29
adding
datastores 23, 104
networks 24
resource pools 22
topology 40
adding clusters, with an application manager 64
adding a software management server, with the Serengeti CLI 16, 20
Ambari 15, 16, 19, 20
Ambari distribution, DNS and FQDN for cluster traffic 60
application manager, modify 17, 20
application managers
adding 95
adding clusters 64
deleting 18, 22
listing 97
managing 15
monitoring 16, 22, 81
appmanager, commands 95, 96
appmanager add, command 95, 97
appmanager commands 95
appmanager delete command, appmanager delete 96
appmanager list, command 95, 97
appmanager add command 15, 19, 95
appmanager list command 16, 22, 81, 97
appmanager modify command 96

B
backup and restore 79
backup.sh 79
backup.sh script 79
balancing workloads 40
basic cluster 63
basic Hadoop clusters 37
black listed Hadoop attributes 92
burst out to virtual 61, 65

C
CLI, accessing 11
CLI command reference 95
client nodes for Hadoop 37
cloning performance 26
Cloudera distribution
administrative commands with the Serengeti CLI 11
DNS and FQDN for cluster traffic 40, 52, 54, 55, 59, 66–69
Cloudera Manager 15, 16, 19, 20
cluster create command 62, 98
cluster delete command 75, 99
cluster node virtual machine 77
cluster resize command 72, 102
cluster commands 97
cluster config command 73, 75, 97
cluster expand 99
cluster expand Command 76, 99
cluster export command 73, 100
cluster fix command 77, 101
cluster list command 83, 101
cluster names 62
cluster resetParam command 101
cluster setParam command 103
cluster specification files
annotated example 86
cluster definition requirements 85
compute-only cluster 57, 59, 60
data-compute separated clusters 54
defining attributes 89
file requirements 85
Hadoop distribution JARs 73
node group overrides 75
node placement 55
nodes 68
placement policies 39
reconfiguring clusters 73
reference 85
resource pool symbolic link 22
topology 39
topology constraints 52
cluster start command 71, 103
cluster stop command 71, 103
cluster with external namenode HA HDFS 47, 49, 50

clusters
adding with an application manager 64
assigning networks 67
assigning resources 67
attributes in definitions 89
basic Hadoop 37
compute workers only 61, 65
compute-only 37, 57, 59, 60
configuring 68, 97
creating, See creating clusters
custom administrator passwords 65
customized 37
data-compute separated 37, 54, 55
default Hadoop configuration 37
default HBase configuration 38
defining nodes 89
definition requirements in cluster specification files 85
definitions, exporting 100
deleting 75, 99
deploying under different resource pools 23
elastic scaling 101
failover 75
Hadoop default 37
HBase 37, 44
HBase only 43
managing 71
manual scaling 101
naming 62
node administrator passwords 65
node group roles 69
reconfiguring 73, 75
scaling 102
scaling out 72
starting 71, 103
stopping 71, 103
topology 38, 40, 52
viewing provisioned 83, 101
Command-Line Interface, accessing 11
commands, appmanager 95
compute capacity, scaling 72
compute only clusters 51
compute workers only cluster 61, 65
compute-only clusters 57, 59, 60
configuration file 29
configuration files, converting Hadoop XML to Serengeti JSON 73
configurations or roles, listing 17, 21, 82
configuring, clusters 68, 97
connect command 104

connecting
Serengeti services 11
to Serengeti servers 104
convert-hadoop-conf.rb conversion tool 73, 94
converting Hadoop XML to Serengeti JSON 73
create cluster command 47, 49, 50
create external namenode HA HDFS cluster 47, 49, 50
creating clusters
compute-only 57, 59, 60
custom administrator password 65
customized 69
data-compute separated 52, 54, 55
default HBase 41
default Hadoop 62
MapReduce v2 48
placement policies 52
placement policy constraints 55
specifying master, worker, and client nodes 68
topology-aware 40, 52
vSphere HA-protected 44
with an application manager 64
with assigned networks 67
with assigned resources 67
with available distributions 66
creating HBase only clusters, with the CLI 43
customized clusters, creating 69

D
data compute clusters 51
data-compute separated clusters 37, 54, 55
datastore add command 23, 104
datastore commands 104
datastore delete command 23, 104
datastore list command 23, 83, 105
datastores
adding 23, 104
deleting 104
removing 23
viewing 83, 105
defining, node attributes 89
deleting
clusters 75, 99
datastores 104
See also removing
deleting application managers 18, 22
description parameter 16, 20
disconnect command 105
disconnecting from Serengeti servers 105
disk failure, recovering from 77
distributions
listing 17, 21, 82
supported 15, 19
See also Hadoop distributions
distro list command 82, 105
DNS type 25

E
elastic scaling, cluster configuration 101
expand cluster nodes 99
expand cluster 76
exporting, cluster definitions 100
external HDFS cluster, preparing 42
external namenode HA HDFS cluster 47, 49, 50

F
failed disk, recovering 101
federation 66

H
Hadoop clusters
default configuration 37
See also clusters
Hadoop configuration, converting XML to JSON 94
Hadoop nodes
logging in 12
passwords 12
Hadoop Virtualization Extensions (HVE) 38, 40
Hadoop attributes
black listed 92
white listed 92
Hadoop distributions
configuration files 92
JAR files 73
viewing available 82
viewing list of 105
Hadoop distributions supporting MapReduce clusters 37
HBase clusters
configuring 44
creating default 41
default configuration 38
See also clusters
HBase only clusters, creating with the CLI 43
HBase only cluster, using OneFS as external HDFS cluster 42
HBase only cluster, prerequisites 42
HDFS, avoiding node role conflicts 57, 59
HOST_AS_RACK 38
hostname, generate for cluster nodes 25

I
I/O shares 101, 103
IP address segments 25
IP addresses 24

J
Java Runtime Environment (JRE) 11

L
LDAP, managing users 29
LDAP settings, modify 33, 110
LDAP settings, modify 33
LDAP, configuration file 29
list of, application managers 16, 22, 81
list of application managers 16, 22, 81
listing configurations or roles, with the Serengeti CLI 17, 21, 82
listing distributions, with the Serengeti CLI 17, 21, 82
listing supported distributions, with the Serengeti CLI 17, 21, 82
log4j.properties file 73
logging in to Hadoop nodes 12

M
maintenance mode 78
maintenance mode, enter 78
managing, clusters 71
managing resources 19
managing users and user accounts 29
manual scaling, cluster configuration 101
mapping files, rack-hosts 39
MapR distribution, administrative commands with the Serengeti CLI 11
MapReduce clusters, creating 48
MapReduce jobs, HBase clusters 44
MapReduce v1 clusters 37
MapReduce v1 Worker only cluster 49
MapReduce v2 (YARN) clusters 37
MapReduce v2 worker only cluster 50
master nodes for Hadoop 37
memory, scaling 72
mgmtvmcfg Commands 105
mgmtvmcfg modify 106
mgmtvmcfg get Command 105
mgmtvmcfg get 105
mgmtvmcfg modify Command 106
modify an application manager 17, 20
monitoring, Big Data Extensions environment 81
monitoring application managers 16, 22, 81

N
network add command 24, 106
network commands 106
network delete command 24, 106
network list command 24, 107
network modify command 107
resources contention, addressing 72
resource pools
adding 22
removing 23
viewing 84
resourcepool add command 22, 108
resourcepool commands 108
resourcepool delete command 23, 108
resourcepool list command 23, 84, 108
restore.sh script 79
roles, listing 17, 21, 82
S
scaling
clusters 72, 102
CPU 72
parameters, configuring 103
parameters, resetting 101
RAM 72
Serengeti servers
connecting to 104
disconnecting from 105
Serengeti services, connecting 11
serengeti-maintenance.sh 78
serengeti-maintenance.sh script 78
server information, adding 16, 20
Single Sign-On (SSO) 11
software management server, adding 16, 20
starting clusters 71, 103
stopping, clusters 103
stopping clusters 71
supported distributions, listing 17, 21, 82
T
template list command 109
topology
adding 40
cluster 38
commands 109
constraints 52
placement policies 39
topology rack-hosts mapping files 39
topology upload command 40, 109
topology list command 39, 40, 109
U
updated information 9
uploading, topology 40
user modes, change 32
user authentication
Active Directory 106
change mode 106
LDAP 106
local 106
mixed 106

See also user management

usermgmtserver 109
usermgmtserver Command 109
usermgmtserver modify 110
usermgmtserver modify Command 110

users and user accounts, managing 29

V

viewing
  clusters 101
datastores 83, 105
Hadoop distributions, available 82, 105
networks 83
  provisioned clusters 83
resource pools 84
virtual machines, predefined sizes for
  Serengeti 89
vSphere Fault Tolerance (FT) 75
vSphere resources
  assigning to clusters 67
  resource pools 22
vSphere High Availability (HA) 44, 75

W

white listed Hadoop attributes 92
with the CLI 64
worker nodes for Hadoop 37
workloads, balancing 40

X

XML Hadoop configuration, converting to
  JSON 94

Y

Yarn cluster 50
Yarn Worker only cluster 50