

Hortonworks Data Platform

Installing HDP Manually

(Jan 20, 2014)

Hortonworks Data Platform : Installing HDP Manually

Copyright © 2012, 2013, 2014 Hortonworks, Inc. Some rights reserved.

The Hortonworks Data Platform, powered by Apache Hadoop, is a massively scalable and 100% open source platform for storing, processing and analyzing large volumes of data. It is designed to deal with data from many sources and formats in a very quick, easy and cost-effective manner. The Hortonworks Data Platform consists of the essential set of Apache Hadoop projects including MapReduce, Hadoop Distributed File System (HDFS), HCatalog, Pig, Hive, HBase, Zookeeper and Ambari. Hortonworks is the major contributor of code and patches to many of these projects. These projects have been integrated and tested as part of the Hortonworks Data Platform release process and installation and configuration tools have also been included.

Unlike other providers of platforms built using Apache Hadoop, Hortonworks contributes 100% of our code back to the Apache Software Foundation. The Hortonworks Data Platform is Apache-licensed and completely open source. We sell only expert technical support, [training](#) and partner-enablement services. All of our technology is, and will remain free and open source.

Please visit the [Hortonworks Data Platform](#) page for more information on Hortonworks technology. For more information on Hortonworks services, please visit either the [Support](#) or [Training](#) page. Feel free to [Contact Us](#) directly to discuss your specific needs.



Except where otherwise noted, this document is licensed under
Creative Commons Attribution ShareAlike 3.0 License.
<http://creativecommons.org/licenses/by-sa/3.0/legalcode>

Table of Contents

1. Getting Ready to Install	1
1.1. Understand the Basics	1
1.2. Meet Minimum System Requirements	2
1.2.1. Hardware Recommendations	2
1.2.2. Operating Systems Requirements	2
1.2.3. Software Requirements	3
1.2.4. Database Requirements	6
1.3. Decide on Deployment Type	8
1.4. Collect Information	8
1.5. Prepare the Cluster Environment	8
1.5.1. Configure common time for all cluster nodes	9
1.5.2. Ensure that Windows networking uses IPv4 addresses	9
1.5.3. Optional - Create Hadoop user	9
1.5.4. Configure ports	9
1.5.5. Enable networking configurations for Workgroups	13
1.5.6. Enable networking configurations for Active Directory Domains	14
1.6. Define Cluster Configuration	24
1.6.1. HDP Setup GUI Interface	24
1.6.2. Using the CLI for Cluster Configuration	28
1.6.3. Configure NameNode High Availability	30
2. Quick Start Guide for Single Node HDP Installation	32
3. Deploying HDP	38
3.1. Option I - Central Push Install Using Corporate Standard Procedures	38
3.2. Option II - Central Push Install Using Provided Script	39
3.3. Option III - Manual Install One Node At A Time	41
3.4. Optional - Install Client Host	43
4. Upgrade HDP Manually	45
4.1. Getting Ready to Upgrade	45
4.2. Backing up critical HDFS metadata	46
4.2.1. Save the HDFS namespace	46
4.3. Backing Up Your Configuration Files	47
4.4. Stopping Running HDP 1.3 Services	47
4.5. Uninstalling HDP 1.3 on All Nodes	47
4.6. Prepare the Metastore Databases	47
4.7. Preparing the HDP 2.0 Cluster Layout	48
4.8. Installing HDP 2.0 and Maintaining Your Prior Data	48
4.9. Upgrading HDFS Metadata	48
4.10. Upgrading HBase	49
4.11. Upgrading Oozie	49
4.12. Starting HDP 2.0 Services	50
4.13. Validating Your Data	50
4.14. Verifying that HDP 2.0 Services are Working	50
4.15. Finalize Upgrade	51
4.16. Troubleshooting	51
4.16.1. Troubleshooting HBase Services not Starting	51
4.16.2. Troubleshooting Flume Services not Starting	51
5. Managing HDP on Windows	52
5.1. Starting the HDP Services	52

5.2. Enabling NameNode High Availability	53
5.3. Validating HA Configuration	54
5.4. Stopping the HDP Services	55
6. Troubleshoot Deployment	56
6.1. Collect Troubleshooting Information	56
6.2. File locations, Ports, and Common HDFS Commands	61
6.2.1. File Locations	62
6.2.2. Ports	64
6.2.3. Common HDFS Commands	65
7. Uninstalling HDP	68
7.1. Option I - Use Windows GUI	68
7.2. Option II - Use Command Line Utility	68
8. Appendix A: Create a Smoke Test User	69

List of Tables

1.1. HDFS Ports	10
1.2. YARN Ports	11
1.3. Hive Ports	11
1.4. WebHCat Port	12
1.5. HBase Ports	12
6.1. HDFS HTTP Ports	65
6.2. HDFS IPC Ports	65

1. Getting Ready to Install

This section describes the information and materials you need to get ready to install the Hortonworks Data Platform (HDP) on Windows.

Use the following instructions before you start deploying Hadoop using HDP installer:

- [Understand the Basics](#)
- [Meet Minimum System Requirements](#)
- [Decide on Deployment Type](#)
- [Collect Information](#)
- [Prepare the Cluster Environment](#)
- [Define Cluster Configuration](#)

1.1. Understand the Basics

The Hortonworks Data Platform consists of three layers.

- **Core Hadoop 1:** The basic components of Apache Hadoop version 1.x.
 - **Hadoop Distributed File System (HDFS)** : A special purpose file system designed to provide high-throughput access to data in a highly distributed environment.
 - **MapReduce:** A framework for performing high volume distributed data processing using the MapReduce programming paradigm.
- **Core Hadoop 2:** The basic components of Apache Hadoop version 2.x.
 - **Hadoop Distributed File System (HDFS)**: A special purpose file system designed to provide high-throughput access to data in a highly distributed environment.
 - **YARN:** A resource negotiator for managing high volume distributed data processing. Previously part of the first version of MapReduce.
 - **MapReduce 2 (MR2):** A set of client libraries for computation using the MapReduce programming paradigm and a History Server for logging job and task information. Previously part of the first version of MapReduce.
- **Essential Hadoop:** A set of Apache components designed to ease working with Core Hadoop.
 - **Apache Pig:** A platform for creating higher level data flow programs that can be compiled into sequences of MapReduce programs, using Pig Latin, the platform's native language.
 - **Apache Hive:** A tool for creating higher level SQL-like queries using HiveQL, the tool's native language, that can be compiled into sequences of MapReduce programs.

- **Apache HCatalog:** A metadata abstraction layer that insulates users and scripts from how and where data is physically stored.
- **WebHCat (Templeton):** A component that provides a set of REST-like APIs for HCatalog and related Hadoop components.
- **Apache HBase:** A distributed, column-oriented database that provides the ability to access and manipulate data randomly in the context of the large blocks that make up HDFS.
- **Apache ZooKeeper:** A centralized tool for providing services to highly distributed systems. ZooKeeper is necessary for HBase installations.
- **Supporting Components:** A set of components that allow you to monitor your Hadoop installation and to connect Hadoop with your larger compute environment.
 - **Apache Oozie:** A server based workflow engine optimized for running workflows that execute Hadoop jobs.
 - **Apache Sqoop:** A component that provides a mechanism for moving data between HDFS and external structured datastores. Can be integrated with Oozie workflows.
 - **Apache Flume:** A log aggregator. This component must be installed manually.
 - **Apache Mahout:** A scalable machine learning library that implements several different approaches to machine learning.

For more information on the structure of the HDP, see [Understanding the Hadoop Ecosystem](#).

1.2. Meet Minimum System Requirements

To run the Hortonworks Data Platform, your system must meet minimum requirements.

- [Hardware Recommendations](#)
- [Operating System Requirements](#)
- [Software Requirements](#)
- [Database Requirements](#)

1.2.1. Hardware Recommendations

Although there is no single hardware requirement for installing HDP, there are some basic guidelines.

You can see sample setups here: [Hardware Recommendations for Apache Hadoop](#).

1.2.2. Operating Systems Requirements

The following operating systems are supported:

- Windows Server 2008 R2 (64-bit)
- Windows Server 2012 (64-bit)

1.2.3. Software Requirements

This section provides download locations and installation instructions for each software prerequisite.

Install the following software on each machine using any one of the following options:

- **Option I - Use CLI:** Download all prerequisites to a single directory and use command line interface (CLI) to install these prerequisites on a machine. For multi-node installations, you can add these CLI commands to a reusable script.
- **Option II - Install manually:** Download each prerequisite and follow the step by step GUI driven manual instructions provided after download.

1.2.3.1. Option I - Use CLI

Identify a workspace directory that will have all the following files and dependencies. In the instructions below, %WORKSPACE% will refer to the full path of the workspace directory. Ensure that you install the following on every host machine in your cluster:

- **Python 2.7.5**

Use the following instructions to manually install Python in your local environment:

1. Download Python from [here](#) to the workspace directory.
2. Update the PATH environment variable. Using Administrator privileges. From the Powershell window, execute the following commands as Administrator user:

```
msiexec /qn /norestart /log %WORKSPACE%\python-2.7.5.log /i %WORKSPACE%\python-2.7.5.msi  
setx PATH "$env:path;C:\Python27" /m
```

where

- %WORKSPACE% is the full workspace directory path.
- \$env is the Environment setting for your cluster.



Important

Ensure the downloaded Python MSI name matches `python-2.7.5.msi`. If not, change the above command to match the MSI file name.

- **Microsoft Visual C++ 2010 Redistributable Package (64-bit)**

1. Use the instructions provided [here](#) to download Microsoft Visual C++ 2010 Redistributable Package (64-bit) to the workspace directory.
2. Execute the following command from Powershell with Administrator privileges:

```
%WORKSPACE%\vcredist_x64.exe /q /norestart
```

For example:

```
C:\prereqs\vcredist_x64.exe /q /norestart
```

- **Microsoft.NET framework 4.0**

1. Use the instructions provided [here](#) to download Microsoft.NET framework 4.0 to the workspace directory.
2. Execute the following command from Powershell with Administrator privileges:

```
%WORKSPACE%\slavesetup\dotNetFx40_Full_setup.exe /q /norestart /log  
%WORKSPACE%\dotNetFx40_Full_setup.exe
```

- **JDK 6u31 or higher**

Use the instructions provided below to manually install JDK to the workspace directory:

1. Check the version. From a command shell or Powershell window, type:

```
java -version
```

2. (Optional): Uninstall the Java package if the JDK version is less than v1.6 update 31.
3. Go to [Oracle Java SE 6 Downloads](#) page and accept the license.

Download the JDK installer to the workspace directory.



Important

Ensure that no whitespace characters are present in the installation directory's path.

For example, `C:\Program Files` is **not** allowed.

4. From Powershell with Administrator privileges, execute the following commands:

```
%WORKSPACE%\jdk-6u31-windows-x64.exe /qn /norestart /log %WORKSPACE%\  
jdk-6u31-windows-x64.log INSTALLDIR=C:\java\jdk1.6.0_31  
setx JAVA_HOME "C:\java\jdk1.6.0_31" /m
```

where `%WORKSPACE%` is the full workspace directory path.



Important

Ensure the downloaded JDK `.exe` file's name matches with `jdk-6u31-windows-x64.exe`. If not, change the above command to match the EXE file name.

For example:

```
C:\prereqs\jdk-6u31-windows-x64.exe /qn /norestart/log C:\  
prereqs\jdk-6u31-windows-x64.log  
INSTALLDIR=C:\java\jdk1.6.0_31
```

1.2.3.2. Option II - Install manually

1. Install **Microsoft Visual C++ 2010 Redistributable Package (64 bit)**

Use the instructions provided [here](#) to download and install Microsoft Visual C++ 2010 Redistributable Package (64 bit).

2. Install **Microsoft.NET framework 4.0**

Use the instructions provided [here](#) to download and install Microsoft.NET framework 4.0.

3. Install **Java JDK**

Your system must have the correct JDK installed on all the nodes of the cluster. HDP supports the following JDKs.

- Oracle JDK 1.6 update 31 64-bit
- Oracle JDK 7 64-bit

Use the following instructions to manually install JDK 1.6 update 31:

- a. Download the [Oracle JDK](#) and install to a directory path that has no whitespace characters in its path. For example, "C:\Program Files\Java\" is not a valid path. "C:\Software\Java\" is a valid path.
- b. Create a system variable named `JAVA_HOME`. The value of this variable will be the full path to installation directory for JDK.
 - i. Open the **Control Panel** -> **System** pane and click on **Advanced system settings**.
 - ii. Click on the **Advanced** tab.
 - iii. Click the **Environment Variables** button.
 - iv. Under **System variables**, click **New**.
 - v. Enter the **Variable Name** as `JAVA_HOME`.
 - vi. Enter the **Variable Value**, as the installation path for the Java Development Kit.

For example, if your JDK is installed at `C:\Software\Java\jdk1.6.0_31`, then you must provide this path to the **Variable Value**.
 - vii. Click **OK**.
 - viii. Click **OK** to close the Environment Variables dialog box.

Use the following instructions to manually install JDK 7:

- a. Download the [Oracle JDK](#) and install to a directory path that has no whitespace characters in its path. For example, "C:\Program Files\Java\" is not a valid path. "C:\Software\Java\" is a valid path. 5

- b. Create a system variable named `JAVA_HOME`. The value of this variable will be the full path to installation directory for JDK.
 - i. Open the **Control Panel -> System** pane and click on **Advanced system settings**.
 - ii. Click on the **Advanced** tab.
 - iii. Click the **Environment Variables** button.
 - iv. Under **System variables**, click **New**.
 - v. Enter the **Variable Name** as `JAVA_HOME`.
 - vi. Enter the **Variable Value**, as the installation path for the Java Development Kit.

For example, if your JDK is installed at `C:\Software\Java\jdk1.7.0_45`, then you must provide this path to the **Variable Value**.
 - vii. Click **OK**.
 - viii. Click **OK** to close the Environment Variables dialog box.

4. Install Python 2.7

- a. Download Python from [here](#).
- b. Update the `PATH` environment variable. Using Administrator privileges:
 - i. Open the **Control Panel -> System** pane and click on the **Advanced system settings** link.
 - ii. Click on the **Advanced** tab.
 - iii. Click the **Environment Variables** button.
 - iv. Under **System Variables**, find `PATH` and click **Edit**.
 - v. In the Edit windows, modify `PATH` by appending the installation path for your Python directory to the value of `PATH`.

For example, if Python executable is installed at `C:\Python27\` then you must append this value to `PATH`.
 - vi. To validate your settings, from a command shell or Powershell window, type:

```
python
```

1.2.4. Database Requirements

- By default, Hive and Oozie use an embedded Derby database for its metastore.

To use an external database for Hive and Oozie metastores, ensure that Microsoft SQL Server database is deployed and available in your environment.



Important

Before using SQL server for Hive metastore, ensure that you set up Microsoft SQL Server JDBC Driver using the instructions provided [here](#).

- Ensure that your database administrator creates the following databases and users.

It is recommended that you note down the database name and user account credentials. You will need these details while configuring the HDP Installer.

- For Hive:

1. hive_dbname



Note

Create Hive database in SQL.

2. hive_dbuser



Note

Create Hive users on SQL and add them to the `sysadmin` role within SQL.

3. hive_dbpasswd

- For Oozie:

1. oozie_dbname



Note

Create Oozie database in SQL.

2. oozie_dbuser



Note

Create Oozie users on SQL and add them to the `sysadmin` role within SQL.

3. oozie_dbpasswd



Important

Ensure that you set the security policy of Microsoft SQL server to use both SQL and Windows authentication. (By default, the security policy uses Windows authentication.)

1.2.4.1. (Optional) Install Microsoft SQL Server JDBC Driver

If you are using MS SQL Server for Hive and Oozie metastores, you must install the MS SQL Server JDBC driver.

1. Download the SQL JDBC JAR file [sqljdbc_3.0.1301.101_enu.exe](#).
2. Run the downloaded file.

(By default, the SQL JDBC driver file will be extracted at `C:\Users\Administrator\Downloads\Microsoft SQL Server JDBC Driver 3.0.`)

3. Copy and paste the `C:\Users\Administrator\Downloads\Microsoft SQL Server JDBC Driver 3.0\sqljdbc_3.0\enu\sqljdbc4.jar` file to `$HIVE_HOME/lib` (where `$HIVE_HOME` can be set to `D:\hadoop\hive-0.9.0`).

1.3. Decide on Deployment Type

While it is possible to deploy all of HDP on a single host, this is appropriate only for initial evaluation.

In general you should use at least three hosts: one master host and two slaves.

1.4. Collect Information

To deploy your HDP installation, you need to collect the following information:

- The network resolvable name for each host in your cluster. This can be the IP address or the hostname.

To determine the hostname for a particular cluster host, open the command shell on that cluster host and execute **hostname**.



Important

The installer will fail if it cannot resolve the hostname of each cluster node.

- (Optional): To be able to use an external database for Hive or Oozie metastore, ensure that you have the hostname (for an existing instance), database name, and user account credentials for the SQL Server instance.



Note

If you are using an existing instance, the database user you create for HDP's use must be granted `ALL PRIVILEGES` on that instance.

1.5. Prepare the Cluster Environment

To deploy HDP across a cluster, you need to prepare your multi-node cluster deploy environment. Follow these steps to ensure each cluster node is prepared to be an HDP cluster node:

- [Ensure that all cluster nodes use a common time](#)
- [Ensure that Windows networking uses IPv4 addresses](#)
- [Optional - Create Hadoop user](#)
- [Configure ports](#)

To use the remote push install and remote services management scripts in a Workgroup cluster, use the following section to set up networking configurations:

- [Enable networking configurations for Workgroups](#)

To use the remote push install and remote services management scripts in an Active Directory Domain cluster, use the following section to set up networking configurations:

- [Enable networking configurations for Active Directory Domains](#)

1.5.1. Configure common time for all cluster nodes

The clocks of all the nodes in your cluster must be able to synchronize with each other. To configure this for Windows Server, use the instructions provided [here](#).

1.5.2. Ensure that Windows networking uses IPv4 addresses

Configure all the Windows Server nodes in your cluster to use IPv4 addresses only. You can either disable IPv6 or set the preference to IPv4.

Use the following Microsoft KB article to disable IPv6: [How to disable IP version 6 or its specific components in Windows](#).

1.5.3. Optional - Create Hadoop user

HDP installer takes the following actions to create hadoop user for your environment:

- If user `hadoop` does not exist, HDP installer automatically creates a local user with random password.
- If the user `hadoop` already exists, HDP installer will change the current password to a new random password. The random password is passed on the command line throughout the install process, then discarded. Administrator can change the password later, but it must be done both in the user configuration and in the service objects installed on each machine via Service Manager.

1.5.4. Configure ports

HDP uses multiple ports for communication with clients and between service components. To enable this communication, you will need to either open all ports or the specific ports that HDP uses.

To open specific ports only, you can set the access rules in Windows.

For example, the following command will open up port 80 in the active Windows Firewall:

```
netsh advfirewall firewall add rule name=AllowRPCCommunication dir=in action=allow protocol=TCP localport=135
```

For example, the following command will open up ports 49152-65535 in the active Windows Firewall:

```
netsh advfirewall firewall add rule name=AllowRPCCommunication dir=in action=allow protocol=TCP localport=49152-65535
```

The tables below specify which ports must be opened for which ecosystem components to communicate with each other.

Make sure that appropriate ports are opened before you install HDP.

HDFS Ports: The following table lists the default ports used by the various HDFS services.

Table 1.1. HDFS Ports

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
NameNode WebUI	Master Nodes (NameNode and any back-up NameNodes)	50070	http	Web UI to look at current status of HDFS, explore file system	Yes (Typically admins, Dev/Support teams)	dfs.http.address
		50470	https	Secure http service		dfs.https.address
NameNode metadata service		8020/9000	IPC	File system metadata operations	Yes (All clients who directly need to interact with the HDFS)	Embedded in URI specified by fs.default.name
DataNode	All Slave Nodes	50075	http	DataNode WebUI to access the status, logs etc.	Yes (Typically admins, Dev/Support teams)	dfs.datanode.http.address
		50475	https	Secure http service		dfs.datanode.https.address
		50010		Data transfer		dfs.datanode.address
		50020	IPC	Metadata operations	No	dfs.datanode.ipc.address
Secondary NameNode	Secondary NameNode and any backup	50090	http	Checkpoint for NameNode metadata	No	dfs.secondary.http.address

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
	Secondary NameNode					

YARN Ports: The following table lists the default ports used by the various YARN services.

Table 1.2. YARN Ports

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
Resource Manager WebUI	Master Nodes (Resource Manager and any back-up Resource Manager node)	8088	http	Web UI for Resource Manager	Yes	yarn.resourcemanager.webapp
Resource Manager	Master Nodes (Resource Manager Node)	8032	IPC	For application submissions	Yes (All clients who need to submit the YARN applications including Hive, Hive server, Pig)	Embedded in URI specified by yarn.resourcemanager.address
NodeManager Web UI	All Slave Nodes	50060	http		Yes (Typically admins, Dev/ Support teams)	yarn.nodemanager.webapp.address

Hive Ports: The following table lists the default ports used by the Hive services.

Table 1.3. Hive Ports

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
HiveServer2	HiveServer2 machine (Usually a utility machine)	10001	thrift	Service for programmatically connecting to Hive	Yes	ENV Variable HIVE_PORT
Hive Server	Hive Server machine (Usually a utility machine)	10000	thrift	Service for clients connecting to Hive	Yes (Clients who need to connect to Hive either programmatically or through UI SQL tools that use JDBC)	ENV Variable HIVE_PORT
Hive Metastore		9083	thrift	Service for programmatically connecting	Yes (Clients that run Hive, Pig and potentially)	hive.metastore.uris

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
				to Hive metadata	M/R jobs that use HCatalog)	

WebHcat Port: The following table lists the default port used by the WebHCat service.

Table 1.4. WebHCat Port

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
WebHCat Server	Any utility machine	50111	http	Web API on top of HCatalog and other Hadoop services	Yes	templeton.port

Table 1.5. HBase Ports

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
HMaster	Master Nodes (HBase Master Node and any back-up HBase Master node)	60000			Yes	hbase.master.port
HMaster Info Web UI	Master Nodes (HBase master Node and back up HBase Master node if any)	60010	http	The port for the HBase-Master web UI. Set to -1 if you do not want the info server to run.	Yes	hbase.master.info.port
Region Server	All Slave Nodes	60020			Yes (Typically admins, dev/ support teams)	hbase.regionserver.port
Region Server	All Slave Nodes	60030	http		Yes (Typically admins, dev/ support teams)	hbase.regionserver.info.port
ZooKeeper	All ZooKeeper Nodes	2888		Port used by ZooKeeper peers	No	hbase.zookeeper.peerport

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
				to talk to each other. See here for more information.		
ZooKeeper	All ZooKeeper Nodes	3888		Port used by ZooKeeper peers to talk to each other. See here for more information.		<code>hbase.zookeeper.leaderport</code>
		2181		Property from ZooKeeper's config <code>zoo.cfg</code> . The port at which the clients will connect.		<code>hbase.zookeeper.property.clientPort</code>

1.5.5. Enable networking configurations for Workgroups

The MSI installation scripts and many utility scripts within HDP require Powershell scripts to be enabled, on every host machine in your Hadoop cluster. Furthermore, the utility scripts (for starting and stopping the whole cluster with a single command) provided with HDP, requires remote scripting and trust to be enabled. Therefore, we strongly recommend that you complete the following three settings on every host in your cluster.

You can set these in Active Directory via Group Policies (for a Group including all hosts in your Hadoop cluster), or you can execute the given Powershell commands on every host in your cluster.



Important

Ensure that the Administrator account on the Windows Server node has a password. The remote scripting below will not work if the Administrator account has an empty password.

Enable remote scripting using Powershell commands

1. On each host in the cluster, execute the following commands in a Powershell window with "Run as Administrator" elevation:

```
Set-ExecutionPolicy "AllSigned"
```

```
Enable-PSRemoting
```

```
Set-item wsman:localhost\client\trustedhosts -value "Host1,Host2"
```

The last argument is a list of comma-separated hostnames in your cluster (for example, "HadoopHost1, HadoopHost2, HadoopHost3").

2. On each host in the cluster, execute the following commands in a Powershell window with "Run as Administrator" elevation:

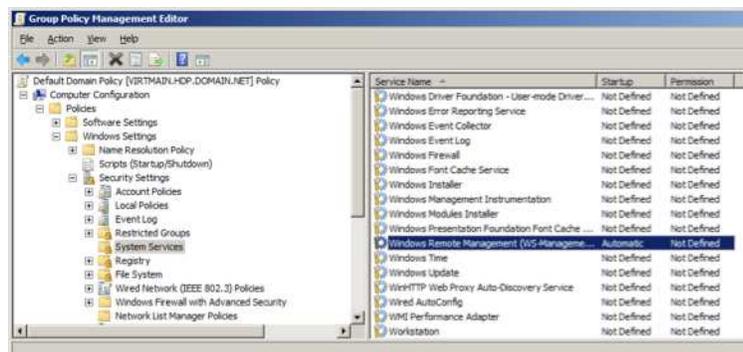
```
winrm quickconfig
winrm set winrm/config/client '@{TrustedHosts="host1, host2, host3"}'
```

The last argument is a list of comma-separated hostnames in your cluster (for example, "HadoopHost1, HadoopHost2, HadoopHost3").

1.5.6. Enable networking configurations for Active Directory Domains

To enable remote scripting and to configure right domain policies for Windows Remote Management complete the following instructions on a domain controller machine (all actions are performed via **Group Policy Management\Default Domain Policy/Edit**):

1. Set the WinRM service to auto start.
 - Go to **Computer Configuration -> Policies -> Windows Settings -> Security Settings -> System Services -> Windows Remote Management (WS-Management)**.

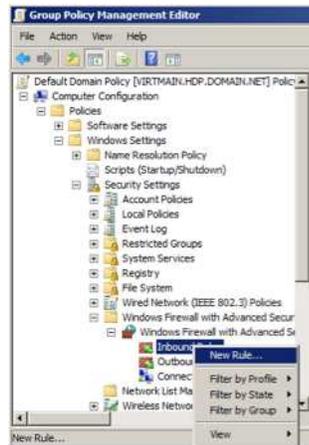


- Set **Startup Mode to Automatic**.

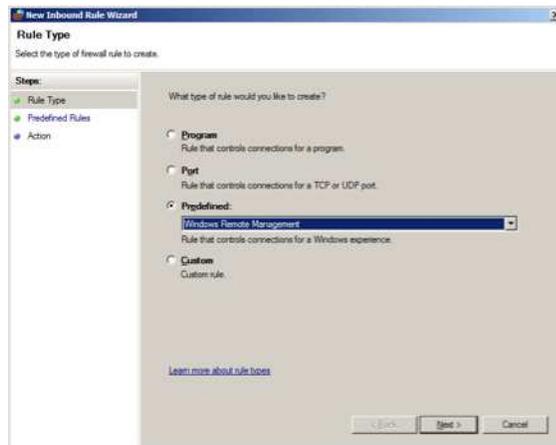


2. Add firewall exceptions to allow the service to communicate.

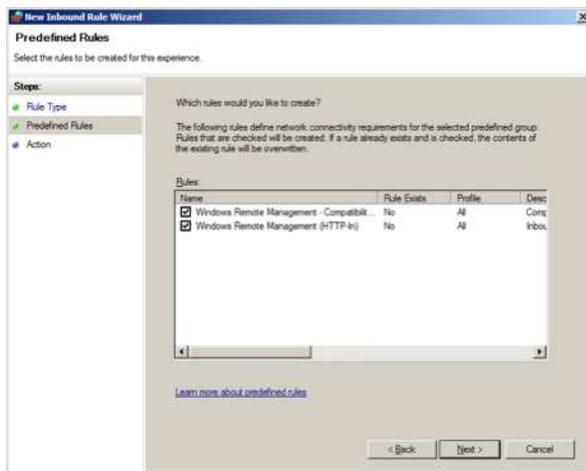
- Go to **Computer Configuration -> Policies -> Windows Settings -> Security Settings -> Windows Firewall with Advanced Security** .
- Right click on **Windows Firewall with Advanced Security** to create a new Inbound Rule.



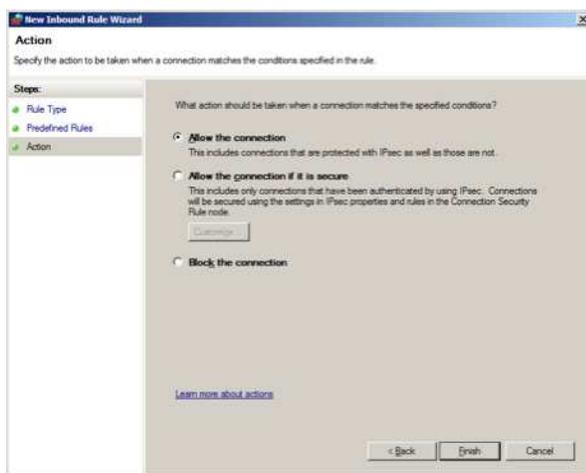
- Select the type of rule as **Predefined as Windows Remote Management** .



The Predefined rule will automatically create two rules as shown below:

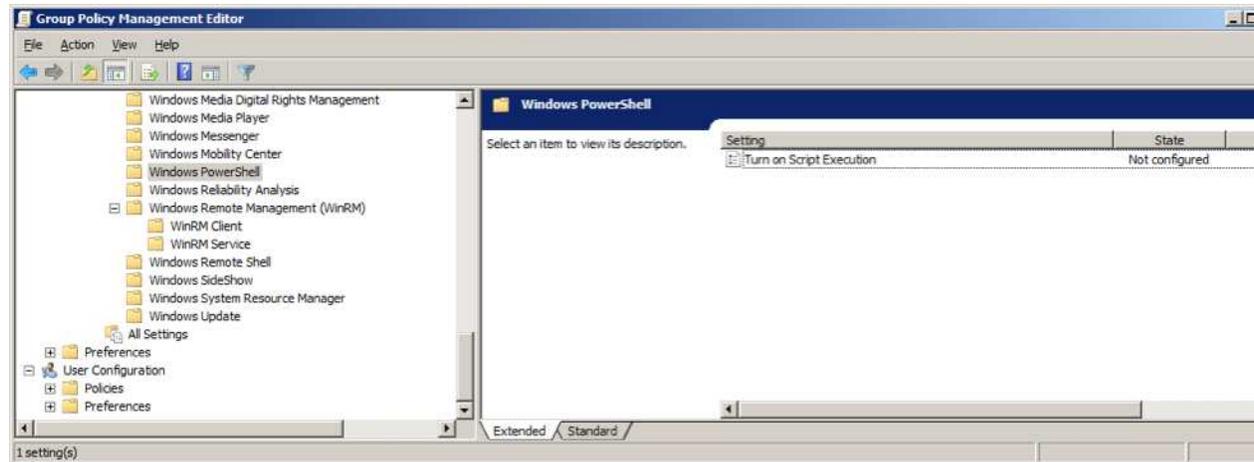


- Configure the **Action** as **Allow the connection** and click **Finish**.

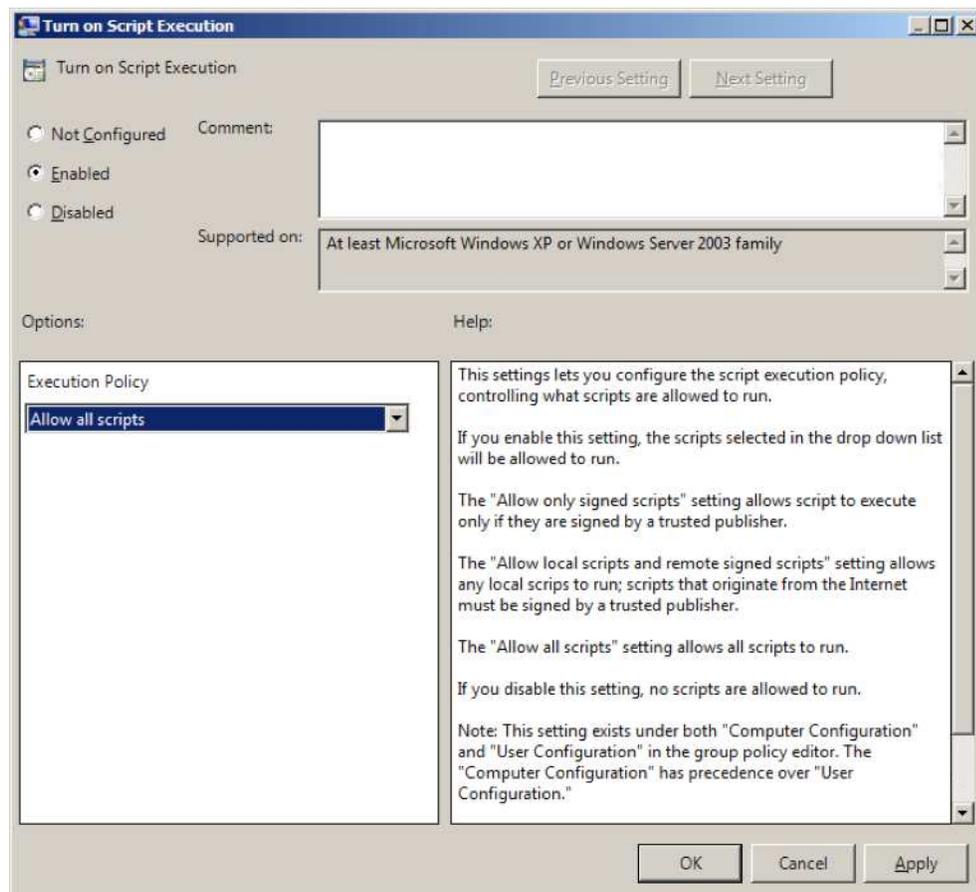


3. Set script execution policy.

- Go to **Computer Configuration -> Policies -> Administrative Templates -> Windows Components -> Windows PowerShell** .
- Enable **Script Execution** .

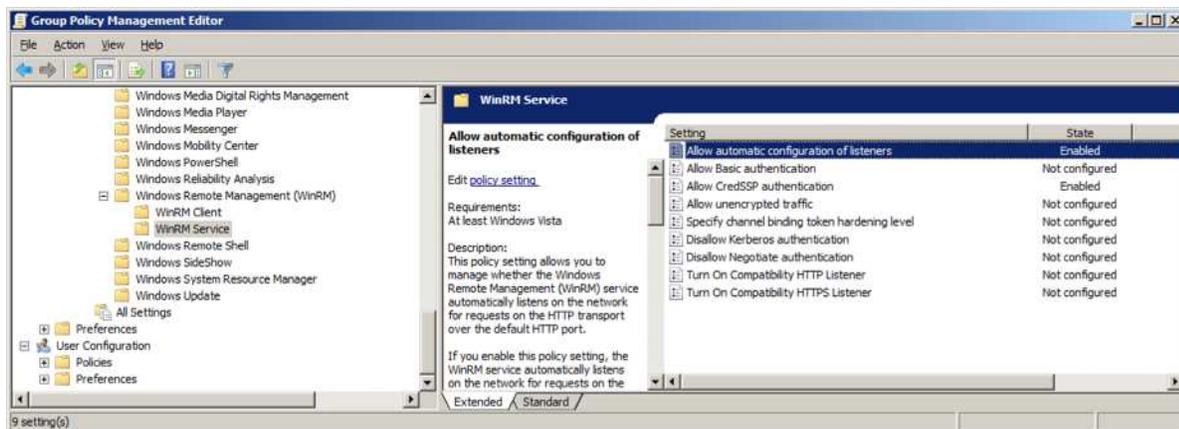


- Set Execution Policy to **Allow all scripts**.

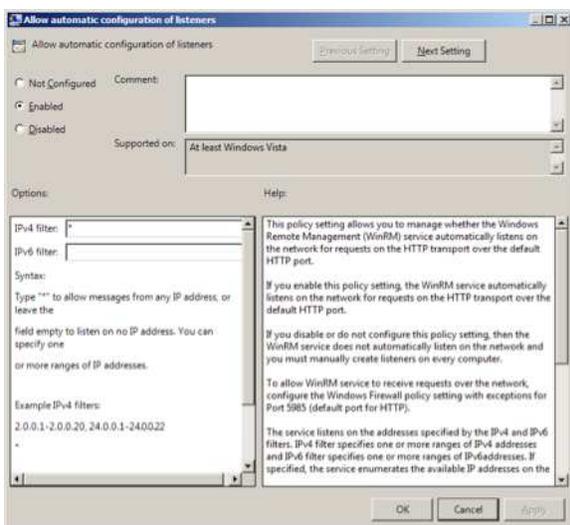


4. Setup WinRM service.

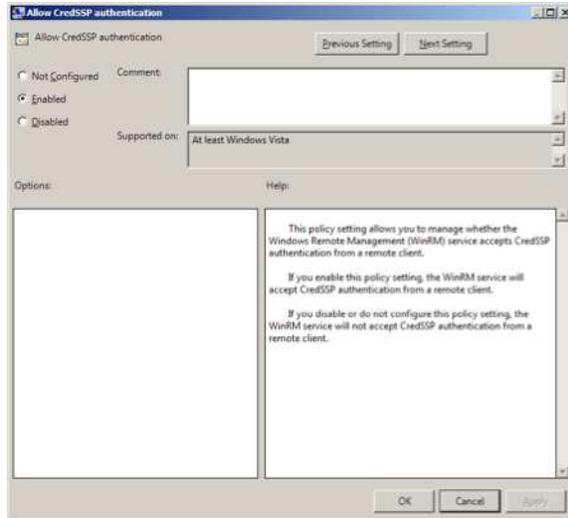
- Go to **Computer Configuration -> Policies -> Administrative Templates -> Windows Components -> Windows Remote Management (WinRM) -> WinRM Service**.



- Create a WinRM listener.
 - a. To allow automatic configuration of listeners, select **Enabled**.
 - b. Set **IPv4 filter** to * (all addresses or specify range)

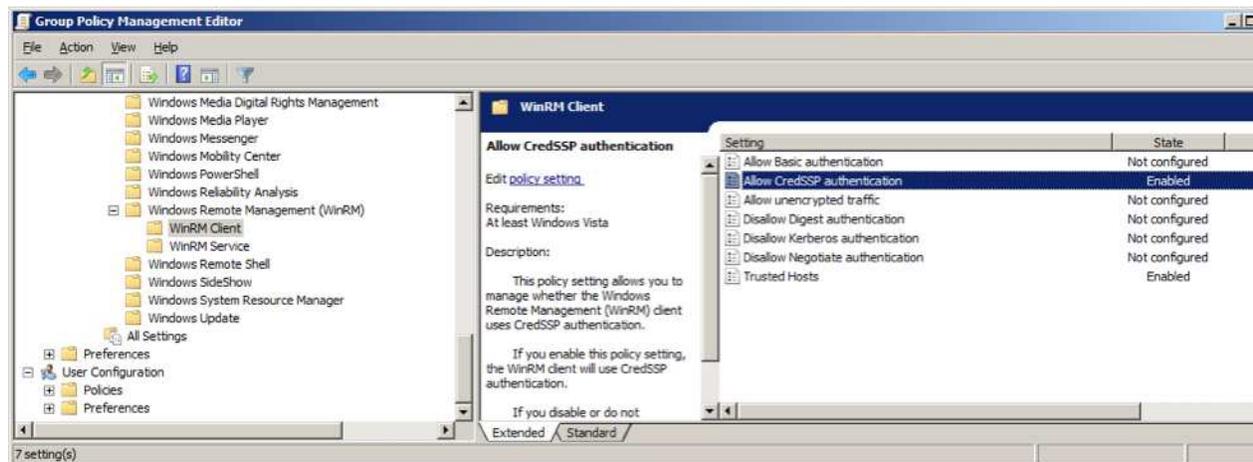


- c. Allow CredSSP authentication and click OK.



5. Setup WinRM client.

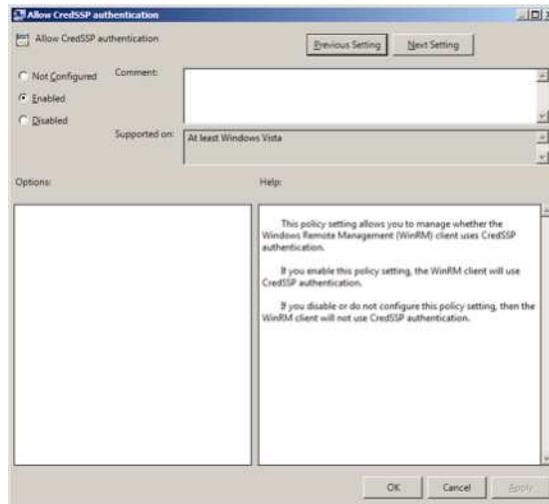
- Go to **Computer Configuration -> Policies -> Administrative Templates -> Windows Components -> Windows Remote Management (WinRM) -> WinRM Client**.



- Configure the trusted host list (the IP addresses of the computers that can initiate connections to the WinRM service). To do this, set **TrustedHostsList** to * (all addresses or specify range).

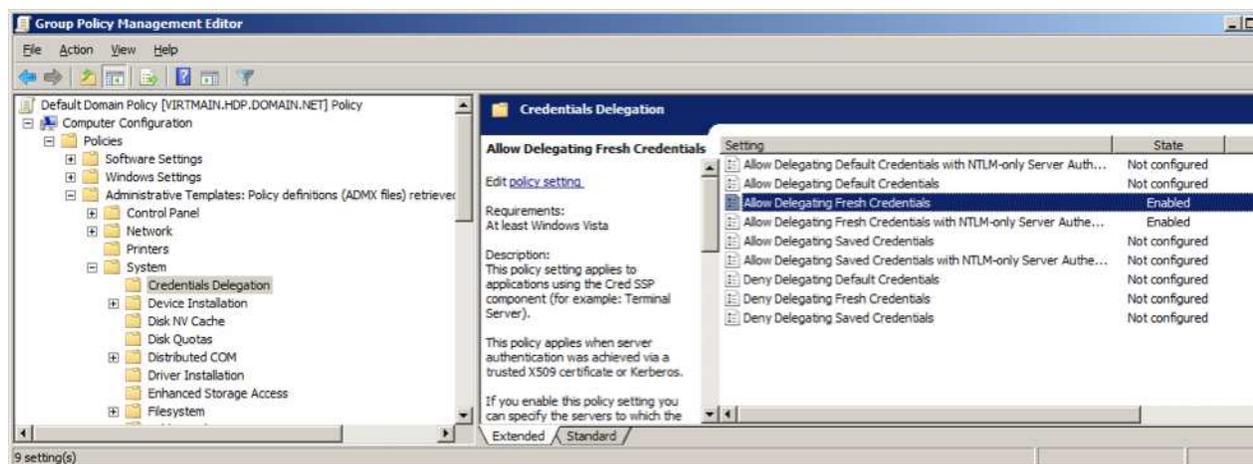


- Allow CredSSP authentication and click OK.

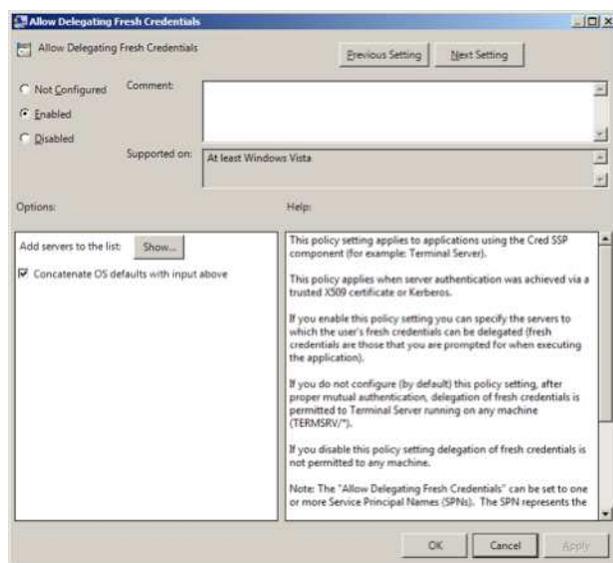


6. Enable credentials delegation.

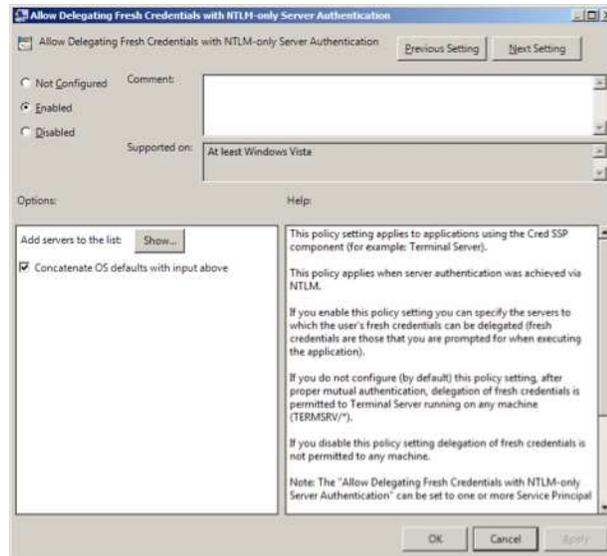
- Go to **Computer Configuration -> Policies -> Administrative Templates -> System -> Credentials Delegation**.



- Select **Enabled** to allow delegation fresh credentials.
- Under **Options** click on **Show**. Set **WSMAN** to * (all addresses or specify range). Click on **Next Setting**.

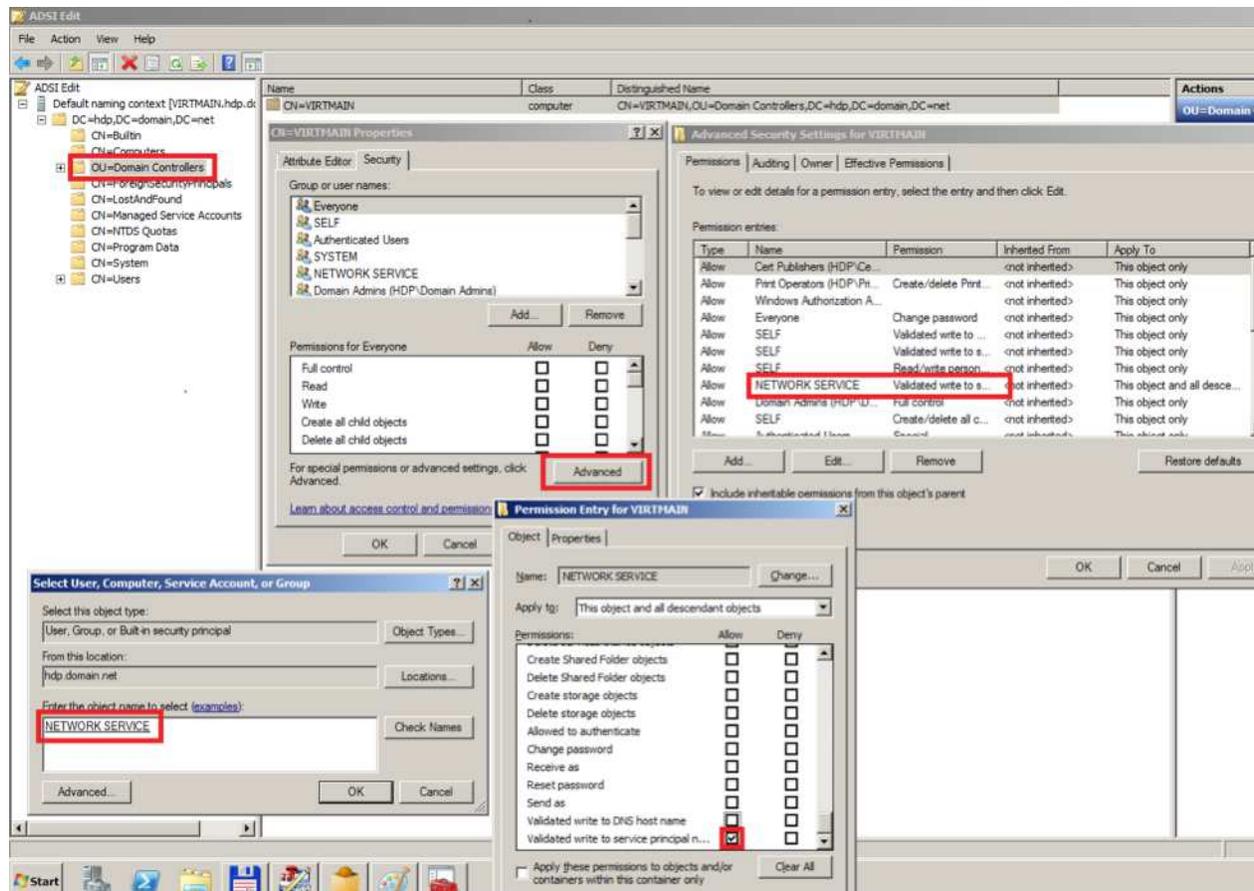


- Select **Enabled** to allow delegation fresh credentials with NTLM-only server authentication.
- Under **Options** click on **Show**. Set **WSMAN** to * (all addresses or specify range). Click on **Finish**.



7. Enable creating WSMAN SPN.

- Go to **Start-> Run**. In the dialog box, type `ADSIEdit .msc` and click **Enter**.
- Expand **OU=Domain Controllers** menu item and select **CN=domain controller hostname**. Go to **Properties -> Security -> Advanced -> Add**.
- Enter **NETWORK SERVICE**, click **Check Names**, then **Ok**. In the Permission Entry select **Validated write to service principal name**. Click **Allow** and **OK** to save your changes.



8. Restart WinRM service and update policies.

- On the domain controller machine, execute the following commands in PowerShell:

```
Restart-Service WinRM
```

- On other hosts in domain, execute the following commands:

```
gpupdate /force
```

- Ensure that SPN-s WSMAN is created for your environment. Execute the following command on your domain controller machine:

```
setspn -l $Domain_Controller_Hostname
```

You should see output similar to the following:

```

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> Restart-Service WinRM
PS C:\Users\Administrator> setspn -l UIRTMAIN
Registered Service SIDs:
  WSMAN/UIRTMAIN
  WSMAN/UIRTMAIN.hdp.domain.net
  E31B6C55EB04/UIRTMAIN.hdp.domain.net
  ldap/UIRTMAIN.hdp.domain.net/ForestDnsZones.hdp.domain.net
  ldap/UIRTMAIN.hdp.domain.net/DomainDnsZones.hdp.domain.net
  DNS/UIRTMAIN.hdp.domain.net
  GC/UIRTMAIN.hdp.domain.net/hdp.domain.net
  RestrictedKrbHost/UIRTMAIN.hdp.domain.net
  RestrictedKrbHost/UIRTMAIN
  HOST/UIRTMAIN/HDP
  HOST/UIRTMAIN.hdp.domain.net/HDP
  HOST/UIRTMAIN
  HOST/UIRTMAIN.hdp.domain.net
  HOST/UIRTMAIN.hdp.domain.net/hdp.domain.net
  E314235-4B06-11D1-AB04-00C04FC2DCD2/e665522-1123-472f-b422-bd2d496c734e/hdp.domain.net
  ldap/UIRTMAIN/HDP
  ldap/e665522-1123-472f-b422-bd2d496c734e._msdcs.hdp.domain.net
  ldap/UIRTMAIN.hdp.domain.net/HDP
  ldap/UIRTMAIN
  ldap/UIRTMAIN.hdp.domain.net
  ldap/UIRTMAIN.hdp.domain.net/hdp.domain.net
PS C:\Users\Administrator>

```

9. Check the WSMAN SPN on other host in domain. Execute the following command on any one of your host machines:

```
setspn -l $Domain_Controller_Hostname
```

You should see output similar to the following:

```

Administrator: Windows PowerShell
  ldap/UIRTMAL01.hdp.domain.net/hdp.domain.net
PS C:\Users\Administrator> setspn -l UIRTMAL01
Registered Service SIDs:
  WSMAN/UIRTMAL01
  WSMAN/UIRTMAL01.hdp.domain.net
  RESTRICTEDKRBHOST/UIRTMAL01
  HOST/UIRTMAL01
  RestrictedKrbHost/UIRTMAL01.hdp.domain.net
  HOST/UIRTMAL01.hdp.domain.net
PS C:\Users\Administrator>

```

1.6. Define Cluster Configuration

The Hortonworks Data Platform consists of multiple components that are installed across the cluster. The cluster properties file specifies the directory locations and node host locations for each of the components.

Use one of the following methods to modify the cluster properties file:

- **Option I - Using the HDP Setup GUI interface** to generate a cluster properties file for GUI use or export a generated `clusterproperties.txt` for a CLI installation. Recommended for first-time users and single-node installations.
- **Option II - Using the CLI** to manually create a `clusterproperties.txt` file, if you are familiar with your systems and HDP requirements.

1.6.1. HDP Setup GUI Interface

The HDP setup interface is a GUI interface where you can set up your cluster properties. After you set the cluster property fields, you can then either export this configuration to be used in a CLI/script driven deployment, or you can kick off the deployment from this GUI itself.

1. Start the MSI GUI.
 - a. Open a Powershell or Command prompt window in Administrative mode.

- b. From here, run the msixec command to start the install:

```
> msixec /i $InstallerMSIpath
```

The MSI displays the HDP Setup GUI which provides options for both single-node and multi-node configuration.

The screenshot displays the 'HDP Setup' window with the following configuration details:

- HDP directory:** c:\hdp
- Log directory:** c:\hadoop\logs
- Data directory:** c:\hdpdata
- Configuration Mode:** Configure Single Node, Configure Multi Node
- Delete existing HDP data:**
- "Hadoop" user password:** [Masked]
- Hosts:**
 - Namenode Host: WIN-Q40CMBVNQ46
 - Secondary Namenode Host: WIN-Q40CMBVNQ46
 - ResourceManager Host: WIN-Q40CMBVNQ46
 - Hive Server Host: WIN-Q40CMBVNQ46
 - Oozie Server Host: WIN-Q40CMBVNQ46
 - WebHcat Host: WIN-Q40CMBVNQ46
 - Slave hosts: WIN-Q40CMBVNQ46
 - Client Hosts: WIN-Q40CMBVNQ46
- Install HDP additional components:**
- Zookeeper hosts:** [Empty]
- Flume hosts:** [Empty]
- HBase Master host:** [Empty]
- Hbase Region Server hosts:** [Empty]
- Database Configuration:**
 - Hive DB Name: hive
 - Hive DB Username: hive
 - Hive DB Password: [Masked]
 - Oozie DB Name: oozie
 - Oozie DB Username: oozie
 - Oozie DB Password: [Masked]
 - DB Flavor: DERBY
 - Database hostname: WIN-Q40CMBVNQ46
 - Database port: 1527
- Advanced Options:**
 - Enable Namenode HA
 - Journal Node Hosts: [Empty]
 - HA Cluster Name: [Empty]
 - Journal Node Edits Directory: [Empty]
 - Standby Namenode Host: [Empty]

2. To install on a single-node system, select Single Node Configuration. The GUI pre-populates the fields with the hostname of the server instance that you are installing on and you can complete your HDP installation following the steps provided in the [Quick Start Guide for Single Node Installation](#).
3. To install on a multi-node system, select Multi Node Configuration and fill out the fields manually. Use the table below:

Configuration Values for MSI Installer

Configuration Property Name	Description	Example
Log directory	HDP's operational logs are written to this directory on each cluster host. Ensure that you have sufficient disk space for storing these log files.	d:\had
Data Directory	HDP data will be stored in this directory on each cluster node. You can add multiple comma-separated data locations for multiple data directories.	d:\hdp
NameNode Host	The FQDN for the cluster node that will run the NameNode master service.	NAMENO
Secondary NameNode Host	The FQDN for the cluster node that will run the Secondary NameNode master service.	SECONDO
ResourceManager Host	The FQDN for the cluster node that will run the YARN Resource Manager master service.	RESOUR
Hive Server Host	The FQDN for the cluster node that will run the Hive Server master service.	HIVE_S
Oozie Server Host	The FQDN for the cluster node that will run the Oozie Server master service.	OOZIE_
WebHcat Host	The FQDN for the cluster node that will run the WebHcat master service.	WEBHCA
Flume hosts	A comma-separated list of FQDN for those cluster nodes that will run the Flume service.	FLUME_ FLUME_
HBase Master host	The FQDN for the cluster node that will run the HBase master.	HBASE_
HBase Region Server hosts	A comma-separated list of FQDN for those cluster nodes that will run the HBase Region Server services.	slave1
Slave hosts	A comma-separated list of FQDN for those cluster nodes that will run the DataNode and TaskTracker services.	slave1
Zookeeper hosts	A comma-separated list of FQDN for those cluster nodes that will run the Zookeeper hosts.	ZOOKEE
Hive DB Name	Database for Hive metastore. If using SQL Server, ensure that you create the database on the SQL Server instance.	hivedb
Hive DB Username	User account credentials for Hive metastore database instance. Ensure that this user account has appropriate permissions.	hive_u
Hive DB Password		hive_p
Oozie DB Name	Database for Oozie metastore. If using SQL Server, ensure that you create the database on the SQL Server instance.	ooziedb
Oozie DB Username	User account credentials for Oozie metastore database instance. Ensure that this user account has appropriate permissions.	oozie_u
Oozie DB Password		oozie_p
DB Flavor	Database type for Hive and Oozie metastores (allowed databases are SQL Server and Derby). To use default embedded Derby instance, set the value of this property to <code>derby</code> . To use an existing SQL Server instance as the metastore DB, set the value as <code>mssql</code> .	mssql
Database Hostname	FQDN for the node where the metastore database service is installed. If using SQL Server, set the value to your SQL Server hostname. If using Derby for Hive metastore, set the value to <code>HIVE_SERVER_HOST</code> .	sqlser

Configuration Property Name	Description	Example
Database port	This is an optional property required only if you are using SQL Server for Hive and Oozie metastores. By default, the database port is set to 1433.	1433

To ensure that a multi-node cluster remains available, you should configure and enable NameNode High Availability. Configuring NameNode High Availability includes defining the locations and names of hosts in a cluster that are available to act as JournalNodes, along with specifying a standby NameNode to fall back on in the event that the primary NameNode fails.

To configure NameNode High Availability, select the **Enable Namenode HA** check box, then enter values in the following fields:

High Availability Configuration Values for MSI Installer

Property	Description	Example Value
Enable Namenode HA	Whether to deploy a highly available NameNode or not.	Selected
Journal Node Hosts	A comma-separated list of FQDN for those cluster nodes that will run the JournalNode processes.	journalnode1.acme.com, journalnode2.acme.com, journalnode3.acme.com
HA Cluster Name	This name is used for both configuration and authority component of absolute HDFS paths in the cluster.	hdp2-ha
Journal Node Edits Directory	This is the absolute path on the JournalNode machines where the edits and other local state used by the JournalNodes (JNs) are stored. You can only use a single path for this configuration.	d:\hadoop\journal
Standby Namenode Host	The host for the standby NameNode.	STANDBY_NAMENODE.acme.com



Note

To [Enable NameNode High Availability](#), you must run several commands while starting cluster services.

- To continue with the GUI installation process, select **Install**.



Note

If you make a configuration mistake and want to clear fields, select **Reset** to clear all fields and begin again.

- To export your HDP Setup configuration as a cluster properties text file and switch to the CLI installation process, select **Export**.



Note

Selecting Export stops the GUI installation process and produces the `clusterproperties.txt` file based on your GUI fields. Verify that all information in the fields are accurate before proceeding.

1.6.2. Using the CLI for Cluster Configuration

Use the following instructions to manually configure the cluster properties for the HDP installer:

1. Create a `clusterproperties.txt` file.
2. Add the properties to the `clusterproperties.txt` file as described in the table below:



Important

- All properties in the `clusterproperties.txt` file must be separated by a newline character.
- Directory paths cannot contain whitespace characters.

For example, `C:\Program Files\Hadoop` is an invalid directory path for HDP.

- Use Fully Qualified Domain Names (FQDN) for specifying the network host name for each cluster host. The FQDN is a DNS name that uniquely identifies the computer on the network. By default, it is a concatenation of the host name, the primary DNS suffix, and a period.
- When specifying the host lists in the `clusterproperties.txt` file, if the hosts are multi-homed or have multiple NIC cards, make sure that each name or IP address by which you specify the hosts is the preferred name or IP address by which the hosts can communicate among themselves. In other words, these should be the addresses used internal to the cluster, not those used for addressing cluster nodes from outside the cluster.
- To Enable NameNode HA, you must include the HA properties at the bottom of the following table.

Configuration Values for MSI Installer

Configuration Property Name	Description	Example
HDP_LOG_DIR	HDP's operational logs are written to this directory on each cluster host. Ensure that you have sufficient disk space for storing these log files.	d:\had
HDP_DATA_DIR	HDP data will be stored in this directory on each cluster node. You can add multiple comma-separated data locations for multiple data directories.	d:\hdp
NAMENODE_HOST	The FQDN for the cluster node that will run the NameNode master service.	NAMENO
SECONDARY_NAMENODE_HOST	The FQDN for the cluster node that will run the Secondary NameNode master service.	SECON
RESOURCEMANAGER_HOST	The FQDN for the cluster node that will run the YARN Resource Manager master service.	RESOUR
HIVE_SERVER_HOST	The FQDN for the cluster node that will run the Hive Server master service.	HIVE_S

Configuration Property Name	Description	Example
OOZIE_SERVER_HOST	The FQDN for the cluster node that will run the Oozie Server master service.	OOZIE_
WEBHCAT_HOST	The FQDN for the cluster node that will run the WebHCat master service.	WEBHCA
FLUME_HOSTS	A comma-separated list of FQDN for those cluster nodes that will run the Flume service.	FLUME_ FLUME_
HBASE_MASTER	The FQDN for the cluster node that will run the HBase master.	HBASE_
HBASE_REGIONSERVERS	A comma-separated list of FQDN for those cluster nodes that will run the HBase Region Server services.	slave1
SLAVE_HOSTS	A comma-separated list of FQDN for those cluster nodes that will run the DataNode and TaskTracker services.	slave1
ZOOKEEPER_HOSTS	A comma-separated list of FQDN for those cluster nodes that will run the Zookeeper hosts.	ZOOKEE
DB_FLAVOR	Database type for Hive and Oozie metastores (allowed databases are SQL Server and Derby). To use default embedded Derby instance, set the value of this property to <code>derby</code> . To use an existing SQL Server instance as the metastore DB, set the value as <code>mssql</code> .	mssql
DB_HOSTNAME	FQDN for the node where the metastore database service is installed. If using SQL Server, set the value to your SQL Server hostname. If using Derby for Hive metastore, set the value to <code>HIVE_SERVER_HOST</code> .	sqlser
DB_PORT	This is an optional property required only if you are using SQL Server for Hive and Oozie metastores. By default, the database port is set to 1433.	1433
HIVE_DB_NAME	Database for Hive metastore. If using SQL Server, ensure that you create the database on the SQL Server instance.	hivedb
HIVE_DB_USERNAME	User account credentials for Hive metastore database instance. Ensure that this user account has appropriate permissions.	hive_u
HIVE_DB_PASSWORD		hive_p
OOZIE_DB_NAME	Database for Oozie metastore. If using SQL Server, ensure that you create the database on the SQL Server instance.	oozied
OOZIE_DB_USERNAME	User account credentials for Oozie metastore database instance.	oozie_
OOZIE_DB_PASSWORD	Ensure that this user account has appropriate permissions.	oozie_
HA	Whether or not to deploy a highly available NameNode.	yes or
HA_JOURNALNODE_HOSTS	A comma-separated list of FQDN for those cluster nodes that will run the JournalNode processes.	journa journa
HA_CLUSTER_NAME	This name will be used for both configuration and authority component of absolute HDFS paths in the cluster.	hdp2-h
HA_JOURNALNODE_EDITS_DIR	This is the absolute path on the JournalNode machines where the edits and other local state used by the JournalNodes (JNs) are stored. You can only use a single path for this configuration.	d:\had
HA_NAMENODE_HOST	The host for the standby NameNode.	STANDE

The following snapshot illustrates a sample `clusterproperties.txt` file:

```
#Log directory
HDP_LOG_DIR=d:\hadoop\logs

#Data directory
HDP_DATA_DIR=d:\hdp\data

#Hosts
NAMENODE_HOST=NAMENODE_MASTER.acme.com
SECONDARY_NAMENODE_HOST=SECONDARY_NAMENODE_MASTER.acme.com
JOBTRACKER_HOST=JOBTRACKER_MASTER.acme.com
```

```

HIVE_SERVER_HOST=HIVE_SERVER_MASTER.acme.com
OOZIE_SERVER_HOST=OOZIE_SERVER_MASTER.acme.com
WEBHCAT_HOST=WEBHCAT_MASTER.acme.com
FLUME_HOSTS=FLUME_SERVICE1.acme.com,FLUME_SERVICE2.acme.com,FLUME_SERVICE3.
acme.com
HBASE_MASTER=HBASE_MASTER.acme.com
HBASE_REGIONSERVERS=slave1.acme.com, slave2.acme.com, slave3.acme.com
ZOOKEEPER_HOSTS=slave1.acme.com, slave2.acme.com, slave3.acme.com
SLAVE_HOSTS=slave1.acme.com, slave2.acme.com, slave3.acme.com

#Database host
DB_FLAVOR=derby
DB_HOSTNAME=DB_myHostName

#Hive properties
HIVE_DB_NAME=hive
HIVE_DB_USERNAME=hive
HIVE_DB_PASSWORD=hive

#Oozie properties
OOZIE_DB_NAME=oozie
OOZIE_DB_USERNAME=oozie
OOZIE_DB_PASSWORD=oozie

```

1.6.3. Configure NameNode High Availability

To ensure that a multi-node cluster remains available, configure and enable NameNode High Availability. Configuring NameNode High Availability includes defining locations and names of hosts in a cluster that are available to act as journal nodes and a standby name node in the event that the primary namenode fails. To configure NameNode High Availability, add the following properties to `clusterproperties.txt` and set values as follows:

Configuring NameNode High Availability in a Windows-based Cluster

Property	Description	Example Value	Mandatory/Optional
HA	Whether to deploy a highly available NameNode or not.	yes or no	Optional
HA_JOURNALNODE_HOSTS	A comma-separated list of FQDN for those cluster nodes that will run the JournalNode processes.	journalnode1.acme.com, journalnode2.acme.com, journalnode3.acme.com	Optional
HA_CLUSTER_NAME	This name is used for both configuration and authority component of absolute HDFS paths in the cluster.	hdp2-ha	Optional
HA_JOURNALNODE_EDITS_DIR	This is the absolute path on the JournalNode machines where the edits and other local state used by the JournalNodes (JNs) are stored. You can only use a single path for this configuration.	d:\hadoop\journal	Optional

Property	Description	Example Value	Mandatory/Optional
HA_NAMENODE_HOST	The host for the standby NameNode.	STANDBY_NAMENODE.acme.com	Optional

To [Enable NameNode High Availability](#), you must run several commands while starting cluster services.

2. Quick Start Guide for Single Node HDP Installation

Use the following instructions to deploy HDP on a single node Windows Server machine:

1. Install the necessary prerequisites using one of the following options:

- **Option I - Use CLI:** Download all prerequisites to a single directory and use command line interface (CLI) to install these prerequisites.
- **Option II - Install manually:** Download each prerequisite and follow the step by step GUI driven manual instructions provided after download.

2. Prepare the single node machine.

a. Collect Information.

Get the hostname of the server where you plan to install HDP. Open the command shell on that cluster host and execute the following command:

```
> hostname  
WIN-RT345SERVER
```

Use the output of this command to identify the cluster machine.

b. Configure firewall.

HDP uses multiple ports for communication with clients and between service components.

If your corporate policies require maintaining per server firewall, you must enable the ports listed [here](#). Use the following command to open these ports:

```
netsh advfirewall firewall add rule name=AllowRPCCommunication dir=in  
action=allow protocol=TCP localport=$PORT_NUMBER
```

- For example, the following command will open up port 80 in the active Windows Firewall:

```
netsh advfirewall firewall add rule name=AllowRPCCommunication dir=in  
action=allow protocol=TCP localport=80
```

- For example, the following command will open ports all ports from 49152 to 65535. in the active Windows Firewall:

```
netsh advfirewall firewall add rule name=AllowRPCCommunication dir=in  
action=allow protocol=TCP localport=49152-65535
```

If your networks security policies allow you open all the ports, use the following instructions to disable Windows Firewall: [http://technet.microsoft.com/en-us/library/cc766337\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/cc766337(v=ws.10).aspx)

- a. Download the HDP for Windows MSI file from: <http://private-repo-1.hortonworks.com/HDP-Win/2.0/2.0.6.0-GA/hdp-2.0.6.0.winpkg.msi>.
- b. Double-click the Windows MSI file to open the Windows Installer.

-or-

Open a command prompt with Administrator privileges and execute the MSI installer command. If you are installing on Windows Server 2012, use this method to open the installer:

```
msiexec /i "<PATH_to_MSI_file>"
```

The following example illustrates the command to launch the installer:

```
msiexec /i "C:\Users\Administrator\Downloads\hdp-2.0.6.0.winpkg.msi"
```

As shown in the example above, the `PATH_to_MSI_file` parameter should be modified to match the location of the downloaded MSI file.

- c. The HDP Setup window appears pre-populated with the host name of the server, as well as default installation parameters.

You must specify the following parameters:

- **Hadoop User Password** –Type in a password for the Hadoop super user (the administrative user). This password enables you to log in as the administrative user and perform administrative actions. Password requirements are controlled by Windows, and typically require that the password include a combination of uppercase and lowercase letters, digits, and special characters.
- **Hive and Oozie DB Names, Usernames, and Passwords** – Set the DB (database) name, user name, and password for the Hive and Oozie metastores. You can use the boxes at the lower left of the HDP Setup window ("Hive DB Name", "Hive DB Username", etc.) to specify these parameters.
- **DB Flavor** –Select DERBY to use an embedded database for the single-node HDP installation.

You can optionally configure the following parameters:

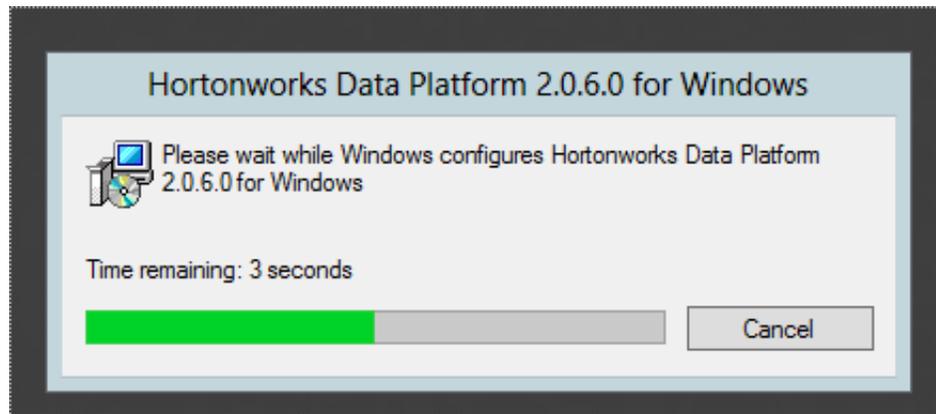
- **HDP Directory** –The directory in which HDP will be installed. The default installation directory is `c:\hdp`.
- **Log Directory** –The directory for the HDP service logs. The default location is `c:\hadoop\logs`.
- **Data Directory** –The directory for user data for each HDP service. The default location is `c:\hdpdata`.
- **Delete Existing HDP Data** –Selecting this check box removes any existing data from prior HDP installs. This will ensure that HDFS will automatically start with a

formatted file system. For a single node installation, it is recommended that you select this option to start with a freshly formatted HDFS.

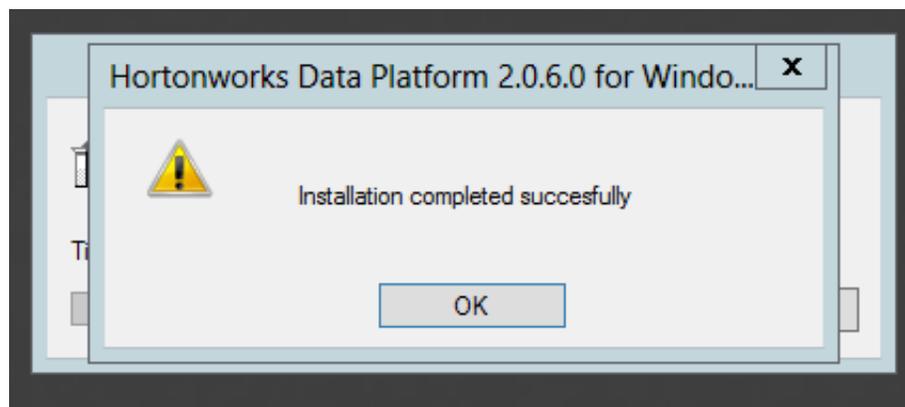
- **Install HDP Additional Components** –Select this check box to install Zookeeper, Flume, and HBase as HDP services deployed to the single node server.

- d. When you have finished setting the installation parameters, click **Install** to install HDP.

The HDP Setup window will close, and a progress indicator will be displayed while the installer is running. The installation may take several minutes. Also, the time remaining estimate may not be accurate.



A confirmation message displays when the installation is complete.



Note: If you did not select the "Delete existing HDP data" check box, the HDFS file system must be formatted. To format the HDFS file system, open the Hadoop Command Line shortcut on the Windows desktop, then run the following command:

```
bin\hadoop namenode -format
```

- e. Start all HDP services on the single machine.

In a command prompt, navigate to the HDP install directory. This is the "HDP directory" setting you specified in the HDP Setup window.

Run the following command from the HDP install directory:

```
start_local_hdp_services
```

- f. Validate the install by running the full suite of smoke tests.

```
Run-SmokeTests
```

Note: You can use the **Export** button on the HDP Setup window to export the configuration information for use in a CLI/script-driven deployment. Clicking **Export** stops the installation and creates a clusterproperties.txt file that contains the configuration information specified in the fields on the HDP Setup window.

The following table provides descriptions and example values for all of the parameters available in the HDP Setup window, and also indicates whether or not each value is required.

Configuration Values for HDP Setup

Configuration Field	Description	Example v
Log directory	HDP's operational logs will be written to this directory on each cluster host. Ensure that you have sufficient disk space for storing these log files.	d:\hadoop
Data directory	HDP data will be stored in this directory on each cluster node. You can add multiple comma-separated data locations for multiple data directories.	d:\hdp\data
NameNode Host	The FQDN for the cluster node that will run the NameNode master service.	NAMENOD
ResourceManager Host	The FQDN for the cluster node that will run the ResourceManager master service.	RESOURCE
Hive Server Host	The FQDN for the cluster node that will run the Hive Server master service.	HIVE_SERV
Oozie Server Host	The FQDN for the cluster node that will run the Oozie Server master service.	OOZIE_SER
WebHcat Host	The FQDN for the cluster node that will run the WebHcat master service.	WEBHCAT_
Flume Hosts	A comma-separated list of FQDN for those cluster nodes that will run the Flume service.	FLUME_SE FLUME_SE
HBase Master Host	The FQDN for the cluster node that will run the HBase master.	HBASE_MA
HBase Region Server hosts	A comma-separated list of FQDN for those cluster nodes that will run the HBase Region Server services.	slave1.acm
Slave hosts	A comma-separated list of FQDN for those cluster nodes that will run the HDFS DataNode and YARN Nodemanager services.	slave1.acm
Zookeeper Hosts	A comma-separated list of FQDN for those cluster nodes that will run the Zookeeper hosts.	ZOOKEEP
DB Flavor	Database type for Hive and Oozie metastores (allowed databases are SQL Server and Derby). To use default embedded Derby instance, set the value of this property to derby. To use an existing SQL Server instance as the metastore DB, set the value as mssql.	mssql or de
Database hostname	FQDN for the node where the metastore database service is installed. If you are using SQL Server, set the value to your SQL Server hostname. If you are using Derby for the Hive metastore, set the value to HIVE_SERVER_HOST.	sqlserver1.
Database port	This is an optional property required only if you are using SQL Server for Hive and Oozie metastores. The database port is set to 1433 by default.	1433
Hive DB Name	Database for Hive metastore. If you are using SQL Server, ensure that you create the database on the SQL Server instance.	hivedb
Hive DB Username	User account credentials for the Hive metastore database instance. Ensure that this user account has appropriate permissions.	hive_user
Hive DB Password	Password for Hive database.	hive_passw

Oozie DB Name	Database for the Oozie metastore. If you are using SQL Server, ensure that you create the database on the SQL Server instance.	ooziedb
Oozie DB Username	User account credentials for the Oozie metastore database instance. Ensure that this user account has appropriate permissions.	oozie_user
Oozie DB Password	Password for Oozie database.	oozie_pass

3. Deploying HDP

Use any one of the following options to deploy Hadoop cluster in your environment:

- [Option I: Central push install using corporate standard procedures](#)
- [Option II: Central push install using provided script](#)
- [Option III: Manual Install one node at a time](#)

3.1. Option I - Central Push Install Using Corporate Standard Procedures

Many Windows Data Centers have standard corporate procedures for performing centralized push-install of software packages to hundreds or thousands of computers at the same time. In general, these same procedures will work for doing a centralized push-install of HDP to a Hadoop cluster.

If your Data Center already has such procedures in place, then follow this simple checklist:

1. Identify the hosts that will constitute the Hadoop cluster nodes, and configure them for centralized push-install, according to your standard procedures.
2. Complete all the prerequisites provided in the section [Prepare the Environment](#).



Note

In many cases, your standard procedures will cover all the suggestions in this section. In some cases, you may need to make additional configuration changes for Hadoop to run correctly. In particular, you will want to enable Remote Scripting from your administrative Master Node to the other nodes in the cluster, since many Powershell scripts are used for Hadoop cluster management and administration. To enable Remote Scripting, see the instructions provided [here](#).

3. Extract the HDP-Win download from [here](#).

Identify the MSI and sample `clusterproperties.txt` file.



Note

Downloaded MSI includes full source, binary-only MSI, and documentation for all components. Hortonworks recommends using the binary-only MSI for faster downloads.

4. Create your own custom `clusterproperties.txt` file, following the instructions in the section [Define Cluster Configuration](#).



Important

When specifying the host lists in the `clusterproperties.txt` file, if the hosts are multi-homed or have multiple NIC cards, make sure that each name or IP address by which you specify the hosts are the preferred name or IP address by which the hosts can communicate among themselves. In other words, these should be the addresses used internal to the cluster, not those used for addressing cluster nodes from outside the cluster.

5. Using your standard procedures, push both the MSI and the custom `clusterproperties.txt` file to each node in the cluster. Alternatively, you can place the MSI and the `clusterproperties.txt` file in a network share location accessible via CIFS file path from each node in the cluster.



Note

You must place the two files together in the same target directory on each node.

6. Continuing to use your standard procedures, remotely execute on each node the `msiexec` command documented in section [Install from the MSI \[42\]](#). This will cause the MSI to install itself, using the parameters in the `clusterproperties.txt` file.
7. Examine the return results and/or logs from your standard procedures to ensure that all nodes were successfully installed.
8. Smoke test your installation using the instructions provided in the [Validate the Install \[43\]](#) section.

3.2. Option II - Central Push Install Using Provided Script

If your Data Center does not have established procedures for doing centralized push-install, then you may either follow the below suggested procedure, or you may avoid the issue of centralized installation by manually installing HDP on each individual node in the cluster, documented in section [Installing HDP manually, one node at a time](#).

You can choose to use the helper install script `push_install_hdp.ps1`. The `push_install_hdp.ps1` script only installs one machine at a time, and does all the machines in the cluster in sequence. This is sufficient for a small test cluster, and it does not require any shared storage. It pushes the files to each node using that node's "Administrative Shares", so make sure that the Admin Shares feature is turned on for all hosts in the cluster, and the Administrator user id you will use to run the install script has privileges to write to the Admin Shares on the other hosts.

1. Identify the hosts that will constitute the Hadoop cluster nodes and follow the instructions in section [Prepare the Environment](#) to configure them all for centralized push-install.
2. Extract the HDP-Win zip folder from [here](#) and identify the MSI and example `clusterproperties.txt` file.



Note

Downloaded MSI includes full source, binary-only MSI, and documentation for all components. Hortonworks recommends using the binary-only MSI for faster downloads.

3. Create your own custom `clusterproperties.txt` file using the instructions in section [Define Cluster Configuration](#).
4. Place the MSI and custom `clusterproperties.txt` file in a convenient local subdirectory on the host from which you are running the push install.
 - **Advanced:** You may also place in this subdirectory other files that you want pushed to all the cluster hosts.
5. On the master install node, run a command prompt in Administrator mode, and execute the install scripts with the following parameters:
 - `source_path`: Local or network share directory path where all the installable files reside, including the MSI, the `clusterproperties.txt` file, and any other files you want pushed to the cluster nodes.
 - `target_path`: A single absolute directory path, the same for all target machines, for the install directory (`$HADOOP_NODE`). The path should be specified as it would be locally on a target host. For example, `D:\hadoop\`.
 - `clusterpropfile`: Simple file name of the custom `clusterproperties.txt` file.

The `clusterproperties.txt` resides in the `$source_path` directory. So, ensure that you provide the filename (without any path) as the value for the `clusterpropfile` parameter.

- `files_list`: A single string containing a comma-delimited list of all simple file names to be pushed to all nodes.

Typically these will include the MSI file, the custom `clusterproperties.txt` file, etc.

Ensure that all files are in the `$source_path` directory. These files will be copied to `$target_path` directory.

You may not input paths for these files, only simple names. Ensure that you include the custom `clusterproperties.txt` file.

- `msiexec` command line token sequence: Complete `msiexec` command must be provided just as though you were entering it on a PowerShell command line of one of the target servers, after the files have been pushed.

Start with the word `msiexec` and provide the entire command line, each argument in its own string object.

The `msiexec` command line must be constructed by referring to section [Install from the MSI \[42\]](#).

- Ensure that `<$MSI_PATH>` and `<$PATH_to_clusterproperties.txt_File>` arguments are specified as simple filenames (assume that the `msiexec` command runs from the context of the `target_path`).
- The `<$PATH_to_Installer_Log_File>` argument may be an absolute path or will be interpreted relative to the `target_path`.
- The `HDP_DIR` argument is different from the `target_path`:
 - `target_path` is the location on each host where the MSI and `clusterproperties.txt` files will be copied to while preparing for the installation.
 - `HDP_DIR` is the location on each host where the HDP components will actually be installed by `msiexec`.

The syntax for `msiexec` command line token sequence parameter is given below:

```
msiexec /qn /i "<$MSI_PATH>" /lv "<$PATH_to_Installer_Log_File>"
HDP_LAYOUT="<$PATH_to_clusterproperties.txt_File>" HDP_DIR="<
$PATH_to_HDP_Install_Dir>" HDP_USER_PASSWORD="<password for hadoop account>"
DESTROY_DATA="<Yes_OR_No>"
```



Note

The `push_install_hdp.ps1` script, will deploy the MSI and other files to each host in the cluster by writing (pushing) them from the install master host to the administrative share corresponding to the `target_path` argument on each host. (e.g., a target path of `D:\hadoop\` will cause the admin share `\\hostname\D:\hadoop\` to be written to on each host.)

6. The installer script will return error messages or successful completion results to the Install Master host. These messages will be printed out at the end of the script execution. Examine these return results to ensure that all nodes were successfully installed.
7. Smoke test your installation using the instructions provided in the [Validate the Install \[43\]](#) section.

3.3. Option III - Manual Install One Node At A Time

Use the following instructions to deploy Hadoop using HDP:

1. Complete all the prerequisites provided [here](#).
2. Download the HDP for Windows MSI from [here](#).



Note

Downloaded MSI includes full source, binary-only MSI, and documentation for all components. Hortonworks recommends using the binary-only MSI for faster downloads.

3. Use the instructions provided [here](#) to complete the configuration for HDP installer.
4. Install from MSI.
 - a. Launch the MSI installer with the `clusterproperties.txt` file created previously.



Important

This MSI must be executed on each and every cluster node and must use the same `clusterproperties.txt` file.

- b. On each node, run a command prompt in Administrator mode (to use Administrator privileges), and execute the following command:

```
msiexec /qn /i "<MSI_PATH>" /lv "<PATH_to_Installer_Log_File>"
HDP_LAYOUT="<PATH_to_clusterproperties.txt_File>"
HDP_DIR="<PATH_to_HDP_Install_Dir>" HDP_USER_PASSWORD=
"<Password_for_hadoop_account>" DESTROY_DATA="<Yes_OR_No>"
```

You must provide appropriate values for the following mandatory command line options:

- **HDP_USER_PASSWORD:** Mandatory parameter. Provide the password for the hadoop user account.
- **HDP_LAYOUT:** Mandatory parameter. Provide location of the `clusterproperties.txt` file (For example, `d:\config\clusterproperties.txt`).



Important

The path to the `clusterproperties.txt` file must be absolute. Relative paths will not work.

Optionally, you can also use the following command line options:

- **HDP_DIR:** Install directory for HDP (For example, `d:\hdp`). Default value is `<Default_Drive>/hdp`.
- **DESTROY_DATA:** Specifies whether to preserve or delete existing data in target data directories (allowed values are `undefined`(default), `yes`, and `no`).

The `DESTROY_DATA` parameter takes care of the following conditions:

- During installation, when `HDP_DATA_DIR` has data from previous installation if `DESTROY_DATA` is set to `undefined`, installation will fail.
- During installation, if `DESTROY_DATA` is set to `no`, the installer reuses the existing data and would not format the NameNode.



Note

Installer does not check for the data correctness.

- During installation, if `DESTROY_DATA` is set to `yes`, installation removes previous data and formats the NameNode.
- During installation, if no data exists in `$HDP_DATA_DIR` then the `HDP_DATA_DIR` is created irrespective of the value of `DESTROY_DATA` and NameNode is formatted.

The following example illustrates the command to launch the MSI installer:

```
msiexec /qn /i "hdp-win-1.1.msi" /lv "hdp.log" HDP_LAYOUT="D:\config\clusterproperties.txt" HDP_DIR="D:\hdp\hadoop" DESTROY_DATA="no"
```

- c. **(Optional):** Configure compression for HDFS. Download the `zlib1.dll` from [here](#). Copy the downloaded file to either `$HADOOP_HOME\lib\native` or to `C:\Windows\System32`. To use `GzipCodec`, ensure that you copy the downloaded file to `C:\Windows\System32`.

5. Validate the install.

- Use the instructions provided [here](#) to start the HDP Services.
- On a cluster node, open a command shell and execute the smoke test command script as shown below:

```
cd %HADOOP_NODE_INSTALL_ROOT%
Run-SmokeTests
```

The smoke tests validate the installed functionality by executing a set of tests for each HDP component. You must run these tests as the `hadoop` user or [Create a Smoke Test User](#)



Note

It is recommended to re-install HDP, if you see installation failures for any HDP component.

3.4. Optional - Install Client Host

A client host has all the HDP JAR files on it for communicating with Hive, HDFS, etc. Note that you will not find any HDP service running on the client host machine.

Use the following instructions to install a client host:

- Copy existing `clusterproperties.txt` file from any host machine in your cluster.
- Run the HDP installer from the client host. Execute the following command on your client host machine:

```
msiexec /i "<$MSI_PATH>" /lv "<$PATH_to_Installer_Log_File>" HDP_LAYOUT="<$PATH_to_clusterproperties.txt_File>" HDP_DIR="<$PATH_to_HDP_Install_Dir>" DESTROY_DATA="<Yes_OR_No>"
```

Provide the appropriate values for the following mandatory command line option:

- **HDP_LAYOUT:** Mandatory parameter. Provide location of the copied `clusterproperties.txt` file on your client host machine (For example, `d:\config\clusterproperties.txt`).



Important

The path to the `clusterproperties.txt` file must be absolute. Relative paths will not work.

Optionally, you can also use the following command line options:

- **HDP_DIR:** Install directory for HDP (For example, `d:\hdp`). Default value is `<$Default_Drive>/hdp`.
- **DESTROY_DATA:** Specifies whether to preserve or delete existing data in target data directories (allowed values are `undefined`(default), `yes`, and `no`).

4. Upgrade HDP Manually

This document provides instructions on upgrading an HDP Windows cluster from HDP 1.3 to HDP 2.0. This is an “in-place” upgrade, where your user data and metadata does not need to be moved during the upgrade process but services must be stopped and re-installed.

- [Getting Ready to Upgrade](#)
- [Backing Up Critical HDFS Metadata](#)
- [Backing Up Your Configuration Files](#)
- [Stopping Running HDP 1.3 Services](#)
- [Uninstalling HDP 1.3 on All Nodes](#)
- [Preparing the HDP 2.0 Cluster Layout](#)
- [Prepare the Metastore Databases](#)
- [Installing HDP 2.0 and Maintaining Your Prior Data](#)
- [Upgrading HDFS Metadata](#)
- [Upgrading HBase](#)
- [Upgrading Oozie](#)
- [Starting HDP 2.0 Services](#)
- [Validating Your Data](#)
- [Verifying that HDP 2.0 Services are Working](#)
- [Finalize Upgrade](#)
- [Troubleshooting](#)

4.1. Getting Ready to Upgrade

To prepare for upgrade, you must:

- Identify your existing cluster layout. Copy the `clusterproperties.txt` file from one of the nodes in the cluster. The cluster properties file is located at `$HDP-install-directory/cluster.properties`.
- Confirm that the Uninstallation packages are available on each node in the cluster. The uninstallation package is in `C:\HadoopInstallFiles`. Without these packages, the Uninstaller cannot remove the HDP 1.3 packages on each node.
- Identify where user data and metadata is being stored by HDFS and MapReduce. These directories are retained during the upgrade process.

- Identify the location of new service components to be installed as part of HDP 2.0. The new service components offered in HDP 2.0 are HBase, ZooKeeper, Mahout, and Flume. Note where these new service components will be installed in the set of nodes available in this cluster. You will use this information to define the `clusterproperties.txt` for the HDP 2.0 install.



Note

During upgrade all HDP services are offline. No new jobs or data read/writes are serviced.

4.2. Backing up critical HDFS metadata

Back up the following critical data before attempting an upgrade. On the node that hosts the NameNode, open the **Hadoop Command line** shortcut that opens a command window in the Hadoop directory. Run the following commands:

1. Run the `fsck` command as the HDFS Service user and fix any errors. (The resulting file contains a complete block map of the file system):

```
hadoop fsck / -files -blocks -locations > dfs-old-fsck-1.log
```

2. Capture the complete namespace directory tree of the file system:

```
hadoop fs -lsr / > dfs-old-lsr-1.log
```

3. Create a list of DataNodes in the cluster:

```
hadoop dfsadmin -report > dfs-old-report-1.log
```

4. Capture output from `fsck` command:

```
hadoop fsck / -block -locations -files > fsck-old-report-1.log
```



Note

Verify there are no missing or corrupted files/replicas in the `fsck` command output.

4.2.1. Save the HDFS namespace

To save the HDFS namespace:

1. Place the NameNode in safe mode, to keep HDFS from accepting any new writes:

```
hadoop dfsadmin -safemode enter
```

2. Save the namespace.

```
hadoop dfsadmin -saveNamespace
```



Note

Do NOT leave safe mode from this point onwards. HDFS should not accept any new writes.

3. On the machine that hosts the NameNode, copy the following checkpoint directories into a backup directory:

```
<HDP data directory>/hdfs/nn/edits/current  
<HDP data directory>/hdfs/nn/edits/image  
<HDP data directory>/hdfs/nn/edits/previous.checkpoint
```

4.3. Backing Up Your Configuration Files

Copy the configuration files and write down all important configurations, such as:

- Custom configurations in the cluster for HDP.
- Locations of Hadoop Core configuration files, for example:

```
C:\hdp\hadoop-1.2.0.1.3.0-107\etc\hadoop  
C:\hdp\flume-1.4.0.1.3.0-107\conf  
C:\hdp\hbase-0.94.6.1.3.0-107-hadoop2\conf  
C:\hdp\hcatalog-0.11.0.1.3.0-107\etc
```

4.4. Stopping Running HDP 1.3 Services

Stop all running services. Go to the HDP install directory and run on each cluster node:

```
stop_local_hdp_services.cmd
```

4.5. Uninstalling HDP 1.3 on All Nodes

On each cluster node, uninstall the HDP 1.3 for Windows packages.

From the command-line:

```
> msexec /lv D:\MSI_Download\hdp_uninstall.log /qn /x D:\MSI_Download\hdp-1.3.0.0.winpkg.msi HDP_LAYOUT=D:\MSI_Download\cluster.properties HDP_DIR=D:\hdp HDP=yes DESTROY_DATA=no
```

Or, if you prefer to use the Control Panel:

Go to **Programs and Features-> Uninstall HDP 1.3 for Windows**.

The uninstall option keeps existing data in place, maintaining the data directories for HDFS and MapReduce.

4.6. Prepare the Metastore Databases

Hive and Oozie use a relational database to store metadata. This section assumes that you used SQL Server to store Hive and Oozie metadata for your HDP 1.3 cluster.



Note

If you used the Derby database option, you can skip this section.

If you use a new database name and set up new users, then you must add this new information into the `clusterproperties.txt` file used to upgrade to HDP 2.0.

4.7. Preparing the HDP 2.0 Cluster Layout

To keep the same metadata and user data from HDP 1.3 in HDP 2.0, use the directory settings and database settings in your HDP 1.3 cluster properties configuration file as the base for the HDP 2.0 cluster properties configuration file.

Review [HDP 2.0 installation instructions](#).

Make the following changes to the HDP 1.3 cluster properties file:

1. Change `JOBTRACKER_HOST` to `RESOURCEMANAGER_HOST`.
2. Copy your edited `clusterproperties.txt` file to all nodes in the cluster.

4.8. Installing HDP 2.0 and Maintaining Your Prior Data

To Install HDP 2.0 on all your nodes while maintaining your prior data:

1. Download the HDP 2.0 for Windows MSI installer on all nodes of the cluster.
2. Run the HDP 2.0 MSI from the command line, and use the `clusterproperties.txt` set in the prior section:

```
> msixec /qn /i "hdp-win-1.3.0.0.msi" /lv "hdp.log" HDP_LAYOUT="D:\config\clusterproperties.txt" HDP_DIR="D:\hdp\hadoop" HDP_USER_PASSWORD=<password for hadoop account> DESTROY_DATA="no"
```

3. Verify that you have installed HDP 2.0 on all nodes of your cluster. Do NOT start any services yet.

4.9. Upgrading HDFS Metadata

To upgrade the HDFS Metadata, run the following steps on your NameNode process machine.



Note

This next step requires commands that need to be run with the SuperUser - 'hadoop'. If you do not know the password for the `hadoop` user, then reset the password to a known password and continue.

1. To reset, execute these commands in Administrator mode from a Powershell prompt:

```
$username = "hadoop"
$password = "TestUser123"
$user = [ADSI] ("WinNT://./$username")
$user.SetPassword($password)
Get-WmiObject win32_service -Filter "StartName='.\$username'" | %{ $_.StopService(); $_.Change($null,$null,$null,$null,$null,$null,$null,$password); }
```

2. Start a command prompt in Administrator mode.
3. Switch to the `hadoop` user:

```
>runas /user:hadoop /savecred cmd
```

4. Enter the `hadoop` user password.
5. Execute the NameNode upgrade:

```
>hadoop namenode -upgrade
```

To view the progress of the upgrade, open the NameNode UI. When the upgrade completes, a message similar to the following displays:

```
Upgrade for version -44 has been completed. Upgrade is not finalized.
```



Note

Do not finalize the upgrade at this time. Wait until after you have completed some further verification steps.

6. Close the command shell where the `hadoop namenode -upgrade` command is running to shutdown the NameNode process.

4.10. Upgrading HBase

To upgrade HBase, you must run the following commands as the `hadoop` user on both the HBase Master and the RegionServers hosts:

1. Check for HFiles in V1 format. HBase 0.96.0 discontinues support for HFileV1, but HFileV1 was a common format prior to HBase 0.94. Run the following command to check if there are HFiles in V1 format:

```
>HBASE_HOME\bin\hbase.cmd upgrade -check
```

2. Upgrade HBase.

```
>HBASE_HOME\bin\hbase.cmd upgrade -execute
```

You should see a completed Znode upgrade with no errors.

3. Start services.

4.11. Upgrading Oozie

To upgrade Oozie, run the following commands as the `hadoop` user: "

1. Run `ooziedb.cmd` as the Hadoop user.

```
>runas /user:hadoop "<HDP root dir>\oozie-*\oozie-win-distro\bin\ooziedb.cmd  
upgrade -run"
```

2. Replace your configuration after upgrading. Copy `oozie\conf` from the backup to the `oozie\conf` directory on each server and client.

3. Replace the content of `/user/hadoop/share` in HDFS. On the Oozie server host:
 - a. Back up the `/user/hadoop/share` folder in HDFS and then delete it. If you have any custom files in this folder back them up separately and then add them back after the share folder is updated.

```
> mkdir C:\tmp\oozie_tmp
> runas /user:hadoop "cmd /c hdfs dfs -copyToLocal /user/hadoop/share
C:\tmp\oozie_tmp\oozie_share_backup"
> runas /user:hadoop "cmd /c hdfs dfs -rm -r /user/hadoop/share"
```

- b. Add the latest share libs.

```
> runas /user:hadoop "cmd /c hdfs dfs -copyFromLocal <HDP root dir>\
oozie-*\oozie-win-distro\share /user/hadoop/."
```

4. Start the [Oozie service](#).

4.12. Starting HDP 2.0 Services

After you complete all data upgrades, you can start [HDP 2.0 services](#).

4.13. Validating Your Data

Verify that your data is intact by comparing the HDFS data directory tree with the HDP 1.3 tree.

1. Run an `lsr` report on your upgraded system. Execute the following command from the Hadoop command line:

```
hadoop fs -lsr / > dfs-new-lsr-1.log
```

2. Compare the directory listing to the old directory listing from HDP 1.3. All old directories, files and timestamps should match. There will be some new entries in the HDP 2.0 directory listing:

- `/apps/hbase` is only in HDP 2.0 and is used by HBase
- `/mapred/system/jobtracker` will have a new timestamp

3. Run a `fsck` report on your upgraded system. Execute the following command from the **Hadoop Command Line**:

```
hadoop fsck / -blocks -locations -files fsck-new-report-1.log
```

4. Compare this `fsck` report to the prior to upgrade report to check the validity of your current HDFS data.

4.14. Verifying that HDP 2.0 Services are Working

Run the provided smoke tests as the `hadoop` user or [Create a Smoke Test User](#) in HDFS and run as the `Smoke Test` user to verify that the HDP 2.0 services work as expected:

```
>Run-SmokeTests.cmd
```



Note

You can also verify HDP 2.0 services by running the following Desktop Shortcuts **Hadoop Name Node status**, **HBase Master status**, and **Hadoop YARN status**.

4.15. Finalize Upgrade

When you are satisfied that HDP 2.0 is successfully functioning with the data maintained from the HDP 1.3 cluster, finalize the upgrade.



Note

After you have finalized the upgrade, you cannot revert to your HDP 1.3 cluster.

To finalize the upgrade, execute the following command:

```
> hadoop dfsadmin -finalizeUpgradeTroubleshooting HDP 2.0 service start
```

4.16. Troubleshooting

```
>Run-SmokeTests.cmd
```

4.16.1. Troubleshooting HBase Services not Starting

If the HBase RegionServer and Master is not starting, it could be because a parenthesis in the Path variable caused a problem during setup of the services.

To fix this, run the following commands from an Administrator command prompt:

```
hbase.cmd --service master start > master.xml
hbase.cmd --service regionserver start > regionserver.xml
hbase.cmd --service rest > rest.xml
hbase.cmd --service thrift > thrift.xml
hbase.cmd --service thrift2 > thrift2.xml
```

Then restart the HBase services.

4.16.2. Troubleshooting Flume Services not Starting

If Flume is not starting, it could be because `flumeservice.xml` is missing.

To fix this, Navigate to `%FLUME_HOME%\bin` and locate the `flumeagent.xml` file. If the file does not exist, locate `flumeservice.xml` file and rename it to `flumeagent.xml`.

After the file is renamed, go to **Windows Services** and restart the **Flume agent service**.

5. Managing HDP on Windows

This section describes how to manage HDP on Windows services.

5.1. Starting the HDP Services

The HDP on Windows installer sets up Windows services for each HDP component across the nodes in a cluster. Use the instructions given below to start HDP services from any host machine in your cluster.

Complete the following instructions as the administrative user:

1. Start the HDP cluster.

Navigate to the install directory (as specified by the `HADOOP_NODE_INSTALL_ROOT` environment variable) and execute the following command from any host machine in your cluster.

```
cd %HADOOP_NODE_INSTALL_ROOT%
start_remote_hdp_services.cmd
```

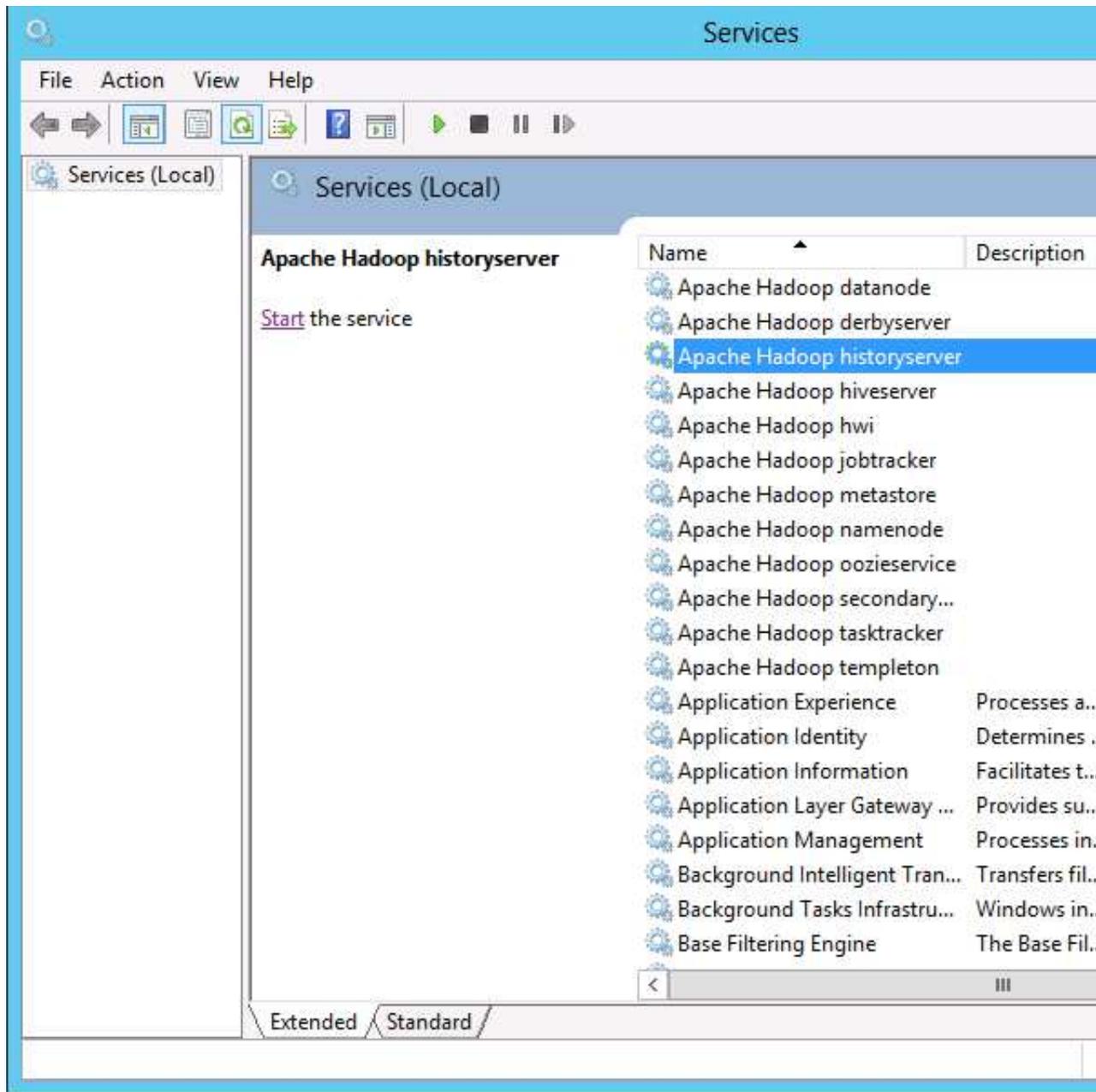


Important

If you want to [Enable NameNode High Availability](#), do so while starting HDP services. Do NOT wait until all services have started.

2. Open the **Services** administration pane, **Control Panel > Administrative Tools > Services**.

Against all the services that are installed successfully, you should see the following message as highlighted in the screenshot below:



5.2. Enabling NameNode High Availability

Optionally, to enable NameNode High Availability in a multi-node cluster, run the following commands on the primary and standby hosts while services are starting.



Important

Log in to every host and run these commands as an administrator user.

1. On the primary host, run

```
hdfs.cmd namenode -format -force
```

2. On each standby host, run

```
hdfs.cmd namenode -bootstrapStandby -force
```

```
hdfs.cmd zkfc -formatZK -force
```

5.3. Validating HA Configuration

1. Verify the state of each NameNode, using one the following methods:

- Open the web page for each NameNode in a browser, using the configured URL.

The HA state of the NameNode should appear in the configured address label. For example: NameNode 'example.com.8020' (standby) .



Note

The NameNode state may be "standby" or "active". After bootstrapping, the HA NameNode state is initially "standby".

- Query the state of a NameNode, using JMX(tag.HAState)
- Query the service state, using the following command:

```
hdfs haadmin -getServiceState
```

2. Verify automatic failover.

- a. Locate the Active NameNode.

Use the NameNode web UI to check the status for each NameNode host machine.

- b. Cause a failure on the Active NameNode host machine.

- i. Turn off automatic restart of the service.

A. In **Windows Services** pane, locate the **Apache Hadoop NameNode** service, right-click, and choose **Properties**.

B. On the **Recovery** tab, select **Take No Action for First, Second, and Subsequent Failures**, then choose **Apply**.

- ii. Simulate a JVM crash.

For example, you can use the following command to simulate a JVM crash:

```
'taskkill.exe /t /f /im namenode.exe'
```

Alternatively, power-cycle the machine, or unplug its network interface to simulate outage.

The Standby NameNode state should become Active within several seconds.



Note

The time required to detect a failure and trigger a failover depends on the configuration of `ha.zookeeper.session-timeout.ms` property. The default value is 5 seconds.

- iii. Verify that the Standby NameNode state is Active.
 - A. If a standby NameNode does not activate, verify that HA settings are configured correctly.
 - B. Check log files for `zkfc` daemons and NameNode daemons to diagnose issues.

5.4. Stopping the HDP Services

The HDP on Windows installer sets up Windows services for each HDP component across the nodes in a cluster. Use the instructions given below to stop HDP services from any host machine in your cluster.

Complete the following instructions as the administrative user:

1. Stop the HDP cluster.

Navigate to the install directory, and execute the following command from any host machine in your cluster.

```
cd %HADOOP_NODE_INSTALL_ROOT%
stop_remote_hdp_services.cmd
```

6. Troubleshoot Deployment

Use the following instructions on troubleshooting installation issues encountered while deploying HDP on Windows platform:

- [Collect Troubleshooting Information](#)
- [File locations, Ports, and Common HDFS Commands](#)

6.1. Collect Troubleshooting Information

Use the following commands to collect specific information from a Windows based cluster. This data helps to isolate specific deployment issue.

1. **Collect OS information:** This data helps to determine if HDP is deployed on a supported operating system (OS).

Execute the following commands on Powershell as an Administrator user:

```
(Get-WmiObject -class Win32_OperatingSystem).Caption
```

This command should provide you information about the OS for your host machine. For example,

```
Microsoft Windows Server 2012 Standard
```

Execute the following command to determine OS Version for your host machine:

```
[System.Environment]::OSVersion.Version
```

2. **Determine installed software:** This data can be used to troubleshoot either performance issues or unexpected behavior for a specific node in your cluster. For example, unexpected behavior can be the situation where a MapReduce job runs for longer duration than expected.

To see the list of installed software on a particular host machine, go to **Control Panel -> All Control Panel Items -> Programs and Features**.

3. **Detect running processes:** This data can be used to troubleshoot either performance issues or unexpected behavior for a specific node in your cluster.

You can either press **CTRL + SHIFT + DEL** on the affected host machine or you can execute the following command on Powershell as an Administrator user:

```
tasklist
```

4. **Detect Java running processes:** Use this command to verify the Hadoop processes running on a specific machine.

As `$HADOOP_USER`, execute the following command on the affected host machine:

```
su $HADOOP_USER  
jps
```

You should see the following output:

```
988 Jps
2816 -- process information unavailable
2648 -- process information unavailable
1768 -- process information unavailable
```

Note that no actual name is given to any process. Ensure that you map the process IDs (pid) from the output of this command to the `.wrapper` file within the `C:\hdp\hadoop-1.1.0-SNAPSHOT\bin` directory.



Note

Ensure that you provide complete path to the Java executable, if Java `bin` directory's location is not set within your `PATH`.

- 5. Detect Java heap allocation and usage:** Use the following command to list Java heap information for a specific Java process. This data can be used to verify the heap settings and thus analyze if a particular Java process is reaching the threshold.

Execute the following command on the affected host machine:

```
jmap -heap $pid_of_Hadoop_process
```

For example, you should see output similar to the following:

```
C:\hdp\hadoop-1.1.0-SNAPSHOT>jmap -heap 2816
Attaching to process ID 2816, please wait...
Debugger attached successfully.
Server compiler detected.
JVM version is 20.6-b01

using thread-local object allocation.
Mark Sweep Compact GC

Heap Configuration:
  MinHeapFreeRatio = 40
  MaxHeapFreeRatio = 70
  MaxHeapSize      = 4294967296 (4096.0MB)
  NewSize          = 1310720 (1.25MB)
  MaxNewSize      = 17592186044415 MB
  OldSize         = 5439488 (5.1875MB)
  NewRatio        = 2
  SurvivorRatio   = 8
  PermSize        = 21757952 (20.75MB)
  MaxPermSize     = 85983232 (82.0MB)

Heap Usage:
New Generation (Eden + 1 Survivor Space):
  capacity = 10158080 (9.6875MB)
  used     = 4490248 (4.282234191894531MB)
  free     = 5667832 (5.405265808105469MB)
  44.203707787298384% used
Eden Space:
  capacity = 9043968 (8.625MB)
  used     = 4486304 (4.278472900390625MB)
  free     = 4557664 (4.346527099609375MB)
  49.60548290307971% used
From Space:
  capacity = 1114112 (1.0625MB)
```

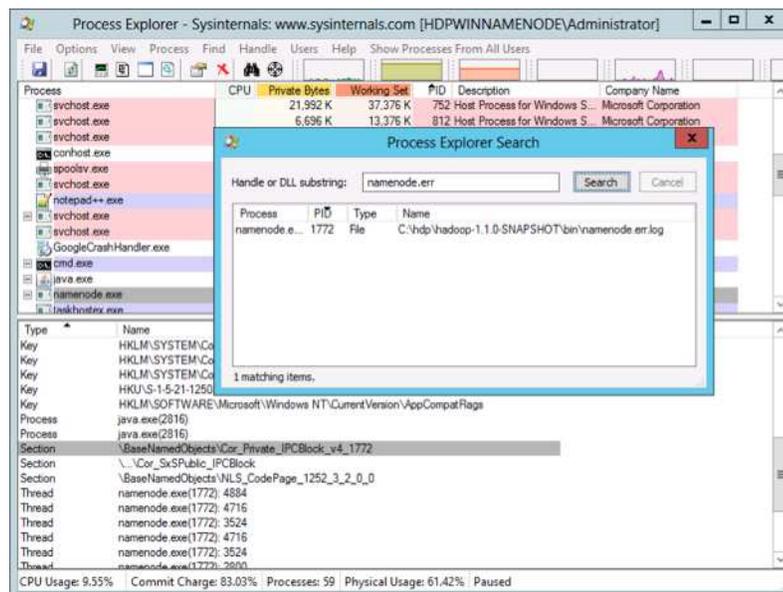
```

used      = 3944 (0.00376129150390625MB)
free      = 1110168 (1.0587387084960938MB)
0.35400390625% used
To Space:
capacity = 1114112 (1.0625MB)
used      = 0 (0.0MB)
free      = 1114112 (1.0625MB)
0.0% used
tenured generation:
capacity = 55971840 (53.37890625MB)
used      = 36822760 (35.116920471191406MB)
free      = 19149080 (18.261985778808594MB)
65.7880105424442% used
Perm Generation:
capacity = 21757952 (20.75MB)
used      = 20909696 (19.9410400390625MB)
free      = 848256 (0.8089599609375MB)
96.10139777861446% used

```

6. **Show open files:** Use Process Explorer to determine which processes are locked on a specific file. See [Windows Sysinternals - Process Explorer](#) for information on using Process Explorer.

For example, you can use Process Explorer to troubleshoot the file lock issues that prevent a particular process from starting as shown in the screenshot below:



7. **Verify well-formed XML:**

Ensure that the Hadoop configuration files (for example, `hdfs-site.xml`, etc.) are well formed.

You can either use **Notepad++** or any third-party tools like **Oxygen**, **XML Spy**, etc. to validate the configuration files. Use the following instructions:

- Open the XML file to be validated in Notepad++ and select **XML Tools -> Check XML Syntax**.

b. Resolve validation errors, if any.

8. **Detect AutoStart Programs:** This information helps to isolate errors for a specific host machine.

For example, a potential port conflict between auto-started process and HDP processes, might prevent launch for one of the HDP components.

Ideally, the cluster administrator must have the information on auto-start programs handy. Use the following command to launch the GUI interface on the affected host machine:

```
C:\Windows\System32\msconfig.exe
```

Click **Startup** tab. Ensure that no startup items are enabled on the affected host machine.

9. **Collect list of all mounts on the machine:** This information determines the drives that are actually mounted or available on the host machine for use. To troubleshoot disks capacity issues, use this command to determine if the system is violating any storage limitations.

Execute the following command on Powershell:

```
Get-Volume
```

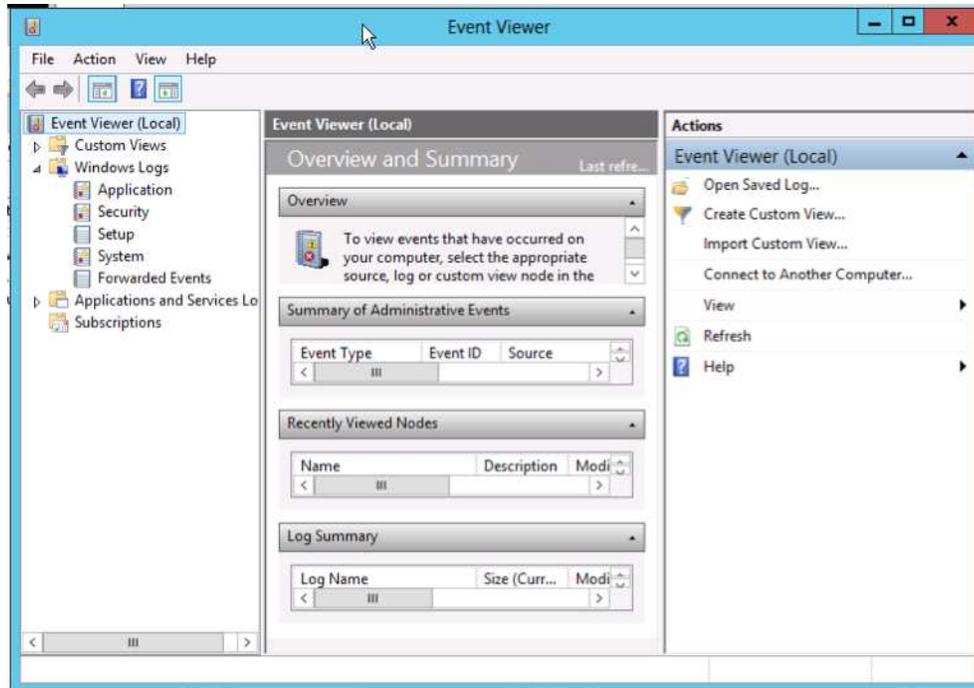
You should see output similar to the following:

DriveLetter	FileSystemLabel	FileSystem	DriveType	HealthStatus
	SizeRemaining	Size		
	System Reserved	NTFS	Fixed	Healthy
	108.7 MB	350 MB		
C	10.74 GB	NTFS	Fixed	Healthy
		19.97 GB		
D	HRM_SSS_X64FR...	UDF	CD-ROM	Healthy
	0 B	3.44 GB		

10. **Operating system messages** Use Event Viewer to detect messages with a system or an application.

Event Viewer can determine if a machine was rebooted or shut down at a particular time. Use the logs to isolate issues for HDP services that were non-operational for a specific time.

Go to **Control Panel -> All Control Panel Items -> Administrative Tools** and click the **Event Viewer** icon.



11.Hardware/system information: Use this information to isolate hardware issues on the affected host machine.

Go to **Control Panel -> All Control Panel Items -> Administrative Tools** and click the **System Information** icon.

12.Network information: Use the following commands to troubleshoot network issues.

- **ipconfig:** This command provides the IP address, validates if the network interfaces are available, and also validates if an IP address is bound to the interfaces. To troubleshoot communication issues between the host machines in your cluster, execute the following command on the affected host machine:

```
ipconfig
```

You should see output similar to the following:

```
Windows IP Configuration

Ethernet adapter Ethernet 2:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::d153:501e:5df0:f0b9%14
    IPv4 Address. . . . . : 192.168.56.103
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.56.100

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : test.tesst.com
    IPv4 Address. . . . . : 10.0.2.15
    Subnet Mask . . . . . : 255.255.255.0
```

```
Default Gateway . . . . . : 10.0.2.2
```

- **netstat -ano:** This command provides a list of used ports within the system. Use this command to troubleshoot launch issues with HDP master processes. Execute the following command on the host machine to resolve potential port conflict:

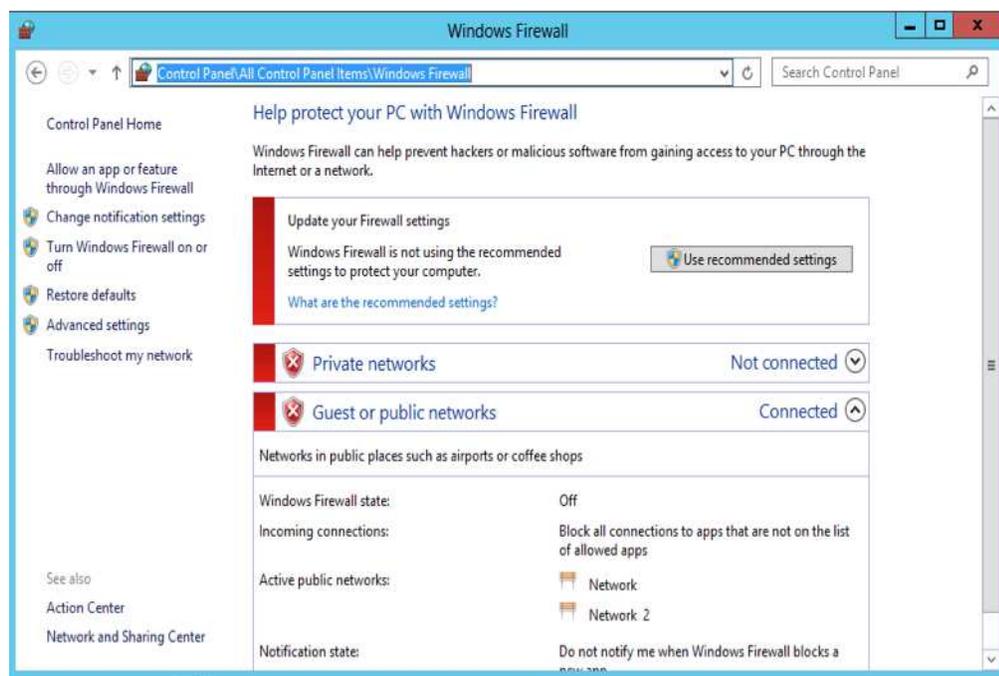
```
netstat -ano
```

You should see output similar to the following:

TCP	0.0.0.0:49154	0.0.0.0:0	LISTENING	752
TCP	[::]:49154	[::]:0	LISTENING	752
UDP	0.0.0.0:500	*:*		752
UDP	0.0.0.0:3544	*:*		752
UDP	0.0.0.0:4500	*:*		752
UDP	10.0.2.15:50461	*:*		752
UDP	[::]:500	*:*		752
UDP	[::]:4500	*:*		752

- **Verify if firewall is enabled on the host machine:** Go to **Control Panel -> All Control Panel Items -> Windows Firewall** .

You should see the following GUI interface:



6.2. File locations, Ports, and Common HDFS Commands

This section provides a list of files and their locations, port information, and HDFS commands that help to isolate and troubleshoot issues:

- [File Locations](#)

- [Ports](#)
- [Common HDFS Commands](#)

6.2.1. File Locations

- **Configuration files:** These files are used to configure a hadoop cluster.

1. `core-site.xml`:

All Hadoop services and clients use this file to locate the NameNode. Therefore, this file must be copied to each node that is either running a Hadoop service or is a client.

The Secondary NameNode uses this file to determine location for storing `fsimage` and edits log `<name>fs.checkpoint.dir</name>` locally and location of the NameNode `<name>fs.default.name</name>`. Use the `core-site.xml` file to isolate communication issues with the NameNode host machine.

2. `hdfs-site.xml`:

HDFS services use this file. Some important properties of this file are as listed below:

- HTTP addresses for the two services
- Replication for DataNodes `<name>dfs.replication</name>`
- DataNode block storage location `<name>dfs.data.dir</name>`
- NameNode metadata storage `<name>dfs.name.dir</name>`

Use `hdfs-site.xml` file to isolate NameNode startup issues. Typically, NameNode startup issues are caused when NameNode fails to load the `fsimage` and edits log to merge. Ensure that the values for all the above properties in `hdfs-site.xml` file are valid locations.

3. `datanode.xml`:

DataNode services use the `datanode.xml` file to specify the maximum and minimum heap size for the DataNode service. To troubleshoot issues with DataNode, change the value for `-Xmx` to change the maximum heap size for DataNode service and restart the affected DataNode host machine.

4. `namenode.xml`:

NameNode services use the `namenode.xml` file to specify the maximum and minimum heap size for the NameNode service. To troubleshoot issues with NameNode, change the value for `-Xmx` to change the maximum heap size for NameNode service and restart the affected NameNode host machine.

5. `secondarynamenode.xml`:

Secondary NameNode services use the `secondarynamenode.xml` file to specify the maximum and minimum heap size for the Secondary NameNode service. To troubleshoot issues with Secondary NameNode, change the value for `-Xmx` to change the maximum heap size for Secondary NameNode service and restart the affected Secondary NameNode host machine.

6. `hadoop-policy.xml`:

Use the `hadoop-policy.xml` file to configure service-level authorization/ACLs within Hadoop. NameNode accesses this file. Use this file to troubleshoot permission related issues for NameNode.

7. `log4j.properties`:

Use the `log4j.properties` file to modify the log purging intervals of the HDFS logs. This file defines logging for all the Hadoop services. It includes information related to appenders used for logging and layout. See [log4j documentation](#) for more details.

- **Log Files:** The following are sets of log files for each of the HDFS services. They are typically stored in `C:\hadoop\logs\hadoop` and `C:\hdp\hadoop-1.1.0-SNAPSHOT\bin` by default.
- **HDFS .out files:** The log files with the `.out` extension for HDFS services are located in `C:\hdp\hadoop-1.1.0-SNAPSHOT\bin` and have the following naming convention:
 - `datanode.out.log`
 - `namenode.out.log`
 - `secondarynamenode.out.log`These files are created and written to when HDFS services are bootstrapped. Use these files to isolate launch issues with DataNode, NameNode, or Secondary NameNode services.
- **HDFS .wrapper files:** The log files with the `.wrapper` extension are located in `C:\hdp\hadoop-1.1.0-SNAPSHOT\bin` and have the following file names:
 - `datanode.wrapper.log`
 - `namenode.wrapper.log`
 - `secondarynamenode.wrapper.log`These files contain startup command string to start the service and they also provide the output of the process ID on service startup.
- **HDFS .log and .err files:**

The following files are located in `C:\hdp\hadoop-1.1.0-SNAPSHOT\bin`:

 - `datanode.err.log`
 - `namenode.err.log`
 - `secondarynamenode.err.log`following files are located in `C:\hadoop\logs\hadoop`:

- `hadoop-datanode-$Hostname.log`
- `hadoop-namenode-$Hostname.log`
- `hadoop-secondarynamenode-$Hostname.log`

These files contain log messages for the running Java service. If there are any errors encountered while the service is already running, the stack trace of the error is logged in the above files.

\$Hostname is the host where the service is running. For example, on a node where the hostname is `namemode.example.com`, the file would be saved as `hadoop-namenode-namemodehost.example.com.log`.



Note

By default, these log files are rotated daily. Use `C:\hdp\hadoop-1.1.0-SNAPSHOT\conf\log4j.properties` file to change log rotation duration.

- **HDFS `.<date>` files:**

The log files with the `.<date>` extension for HDFS services have the following format:

- `hadoop-namenode-$Hostname.log.<date>`
- `hadoop-datanode-$Hostname.log.<date>`
- `hadoop-secondarynamenode-$Hostname.log.<date>`

When a `.log` file is rotated, it is appended with the current date.

An example of the file name would be: `hadoop-datanode-hdp121.localdomain.com.log.2013-02-08`.

Use these files to compare the past state of your cluster with the current state in order to troubleshoot potential patterns of occurrence.

6.2.2. Ports

This section provides information on the ports used by HDFS services.

- **HTTP Ports:** The NameNode and DataNode services have a web interface and therefore listening on an Hypertext Transfer (HTTP) port. This makes it possible for any client with network access to the nodes to go to the webpage and view specific information on the node regarding the HDFS service running on it. Each of the HDFS services can be configured to listen on a specific port.

These ports are configured in the `hdfs-site.xml` file. Use the following table to determine ports for HDFS service, corresponding property in the `hdfs-site.xml` file, and default value of that port.

Table 6.1. HDFS HTTP Ports

HDFS service	Configuration property in <code>hdfs-site.xml</code> file	Default value
NameNode	<code><name>dfs.http.address</name></code>	50070
DataNode	<code><name>dfs.datanode.http.address</name></code>	50075

- **IPC Ports:** Interprocess Communication (IPC) is the communication used between the HDFS services. IPC is a client server architecture. The following table lists the ports that the NameNode and DataNode use for Remote Procedure Call (RPC) calls.

Table 6.2. HDFS IPC Ports

HDFS service	Configuration property	Default value
NameNode	<code><name>fs.default.name</name></code> in the <code>core-site.xml</code>	hdfs:// hostname: 8020
DataNode	<code><name>dfs.datanode.address</name></code> in the <code>hdfs-site.xml</code>	50010
DataNode	<code><name>dfs.datanode.ipc.address</name></code> in the <code>hdfs-site.xml</code>	8010

DataNode uses two different IPC ports. Port 50010 is for data transfer. When a client tries to get or put a file into HDFS the file stream transfer is completed through this port.

HBase uses port 8010 port for short circuit feature. The 8010 port lets a `dfsclient` (located on the same machine as the particular block) access that file directly after making the request on the 8010 port of DataNode to release any holds on the block.

Additionally, this port is also used for DataNodes to communicate with each other when needed.

6.2.3. Common HDFS Commands

This section provides common HDFS commands to troubleshoot HDP deployment on Windows platform. An exhaustive list of the commands is available at [here](#).

- **Get Hadoop version:**

Execute the following command on your cluster host machine:

```
hadoop version
```

- **Check block information:** This command provides a directory listing and displays which node contains the block. Use this command to determine if a block is under replicated.

Execute the following command on your HDFS cluster host machine:

```
hadoop fsck / -blocks -locations -files
```

You should see output similar to the following:

```
FSCK started by hdfs from /10.0.3.15 for path / at Tue Feb 12 04:06:18 PST
2013
/ <dir>
```

```

/apps <dir>
/apps/hbase <dir>
/apps/hbase/data <dir>
/apps/hbase/data/-ROOT- <dir>
/apps/hbase/data/-ROOT-/.tableinfo.0000000001 727 bytes, 1 block(s):
Under replicated blk_-3081593132029220269_1008.
Target Replicas is 3 but found 1 replica(s). 0.
blk_-3081593132029220269_1008
len=727 repl=1 [10.0.3.15:50010]
/apps/hbase/data/-ROOT-/.tmp <dir>
/apps/hbase/data/-ROOT-/70236052 <dir>
/apps/hbase/data/-ROOT-/70236052/.oldlogs <dir>
/apps/hbase/data/-ROOT-/70236052/.oldlogs/hlog.1360352391409 421 bytes,
1 block(s): Under
replicated blk_709473237440669041_1006.
Target Replicas is 3 but found 1
replica(s). 0. blk_709473237440669041_1006 len=421 repl=1 [10.0.3.
15:50010] ...

```

- **HDFS report:** Use this command to receive HDFS status.

Execute the following command as an HDFS user:

```
hadoop dfsadmin -report
```

You should see output similar to the following:

```

-bash-4.1$ hadoop dfsadmin -report
Safe mode is ON
Configured Capacity: 11543003135 (10.75 GB)
Present Capacity: 4097507328 (3.82 GB)
DFS Remaining: 3914780672 (3.65 GB)
DFS Used: 182726656 (174.26 MB)
DFS Used%: 4.46%
Under replicated blocks: 289
Blocks with corrupt replicas: 0
Missing blocks: 0

-----
Datanodes available: 1 (1 total, 0 dead)

Name: 10.0.3.15:50010
Decommission Status : Normal
Configured Capacity: 11543003135 (10.75 GB)
DFS Used: 182726656 (174.26 MB)
Non DFS Used: 7445495807 (6.93 GB)
DFS Remaining: 3914780672 (3.65 GB)
DFS Used%: 1.58%
DFS Remaining%: 33.91%
Last contact: Sat Feb 09 13:34:54 PST 2013

```

- **Safemode:** Safemode is a state where no changes can be made to the blocks. HDFS cluster is in safemode state during start up because the cluster needs to validate all the blocks and their locations. Once validated, the safemode is then disabled.

The options for safemode command are:

```
hadoop dfsadmin -safemode [enter | leave | get]
```

To enter the safemode, execute the following command on your NameNode host machine:

```
hadoop dfsadmin -safemode enter
```

7. Uninstalling HDP

Choose one of the following options to uninstall HDP:

- [Option I - Use Windows GUI](#)
- [Option II - Use command line utility](#)

7.1. Option I - Use Windows GUI

1. Open the **Programs and Features** Control Panel Pane.
2. Select the program listed: **Hortonworks Data Platform 2.0.6.0 for Windows**.
3. With that program selected, click on the **Uninstall** option.

7.2. Option II - Use Command Line Utility

1. On each cluster host, execute the following command from the command shell:

```
msiexec /x "$MSI_PATH" /lv "$PATH_to_Installer_Log_File"
```

where

- `$MSI_PATH` is the full path to MSI.
 - `$PATH_to_Installer_Log_File` is full path to Installer log file.
2. Optionally, you can also specify if you want delete the data in target data directories.

To do this, use the `DESTROY_DATA` command line option as shown below:

```
msiexec /x "$MSI_PATH" /lv "$PATH_to_Installer_Log_File" DESTROY_DATA="yes"
```



Note

During uninstall if `DESTROY_DATA` is not specified or set to `no`, data directories are preserved as well as the `hadoop` user that owns them.

8. Appendix A: Create a Smoke Test User

Creating a smoke test user lets you run HDP smoke tests without having to run them as the hadoop user. To create a smoke test user:

1. Create a smoke test user in HDFS.

- a. Switch to a command prompt as the hadoop user. For example:

```
>runas /user:hadoop cmd
```

- b. From the hadoop directory, create a smoketest user. For example:

```
>D:\hdp\hadoop-2.2.0.2.0.6.0-0009\bin\hadoop dfs -mkdir /user/  
smoketestuser
```

- c. Change ownership to the smoketest user.

```
>D:\hdp\hadoop-2.2.0.2.0.6.0-0009\bin\hadoop dfs chown -R smoketestuser /  
user/smoketestuser
```

2. Create a smoketestuser user account in Windows.

- a. Navigate to Computer Management.

- b. Select **Local Users and Groups > File > Action > New User** on Windows Server 2008 or **Local Users and Groups > Action > New User** on Windows Server 2012.

The New User dialog displays:

The screenshot shows a 'New User' dialog box with the following fields and options:

- User name: smoketestuser
- Full name: Smoke Test User
- Description: QA Smoke Test User
- Password: (masked with dots)
- Confirm password: (masked with dots)
- User must change password at next logon
- User cannot change password
- Password never expires
- Account is disabled

Buttons at the bottom: Help, Create, Close.

- c. Create the username and password for your smoke test user. Determine password requirements and select **Create**.
3. Validate the smoketest user by running smoketests as the smoketest user.
 - a. Switch to a command prompt as the smoketestuser. For example:

```
>runas /user:smoketestuser cmd
```

- b. As smoketestuser, run the smoke tests:

```
>Run-Smoktests.cmd
```