Hortonworks Data Platform

Installing HDP Manually

(Jun 12, 2013)

docs.hortonworks.com

Hortonworks Data Platform : Installing HDP Manually

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.

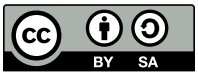


Table of Contents

List of Tables

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**: The basic components of Apache Hadoop.
  - **Hadoop Distributed File System (HDFS)**: A special purpose file system that is designed to work with the MapReduce engine. It provides 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.
- **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 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 pre-requisite.

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 6u31**

   Use the following instructions to install the JDK:

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 **Apply Changes**.

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 Secondanry NameNode	50090	http	Checkpoint for NameNode metadata	No	`dfs.secondary.http.address`

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

Table 1.2. MapReduce Ports

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
JobTracker WebUI	Master Nodes (JobTracker Node and any back-up Job-Tracker node)	50030	http	Web UI for JobTracker	Yes	`mapred.job.tracker.http.address`
JobTracker	Master Nodes (JobTracker Node)	8021	IPC	For job submissions	Yes (All clients who need to submit the MapReduce jobs including Hive, Hive server, Pig)	Embedded in URI specified by `mapred.job.tracker`
Task-Tracker Web UI and Shuffle	All Slave Nodes	50060	http	DataNode Web UI to access status, logs, etc.	Yes (Typically admins, Dev/ Support teams)	`mapred.task.tracker.http.address`
History Server WebUI		51111	http	Web UI for Job History	Yes	`mapreduce.history.server.http.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 (Thrift/ JDBC) connecting to Hive	Yes	`ENV Variable HIVE_PORT`
Hive Server	Hive Server machine (Usually a utility machine)	10000	thrift	Service for programmatically (Thrift/ JDBC) connecting to Hive	Yes (Clients who need to connect to Hive either programatically or through UI SQL tools that use JDBC)	`ENV Variable HIVE_PORT`
Hive Metastore		9083	thrift	Service for programmatically (Thrift/ JDBC)	Yes (Clients that run Hive,	`hive.metastore.uris`

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
				connecting to Hive metadata	Pig and potentially 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	No	`hbase.zookeeper.peerport`

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
				peers 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.		`hbase.zookeeper.leaderport`
		2181		Property from ZooKeeper's config `zoo.cfg`. The port at which the clients will connect.		`hbase.zookeeper.property.clientPort`

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"}'
```

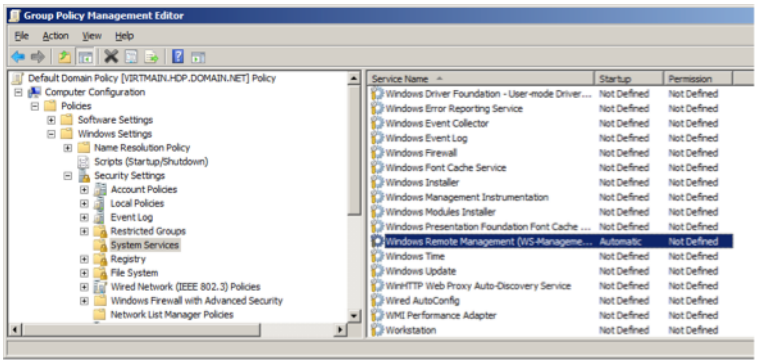
The last argument is a list of comma-separated hostnames in your cluster (for example, "`HadoopHost1`, `HadoopHost2`, `HadoopHost3`").

1.5.6. Enable networking configurations for Active Directory Domains

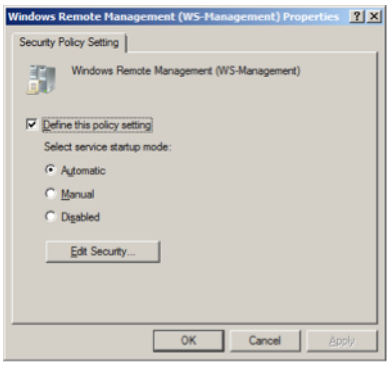
To enable remote scripting and to configure right domain policies for Windows Remote Management complete the following instructions on a domain controller machine (all actions are performed via **Group Policy Management****Default Domain Policy**/**Edit**):

1. Set the WinRM service to auto start.

   - Go to **Computer Configuration** -> **Policies** -> **Windows Settings** -> **Security Settings** -> **System Services** -> **Windows Remote Management (WS-Management)**.

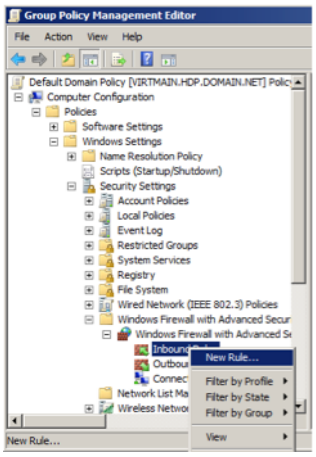


   - Set **Startup Mode** to **Automatic**.

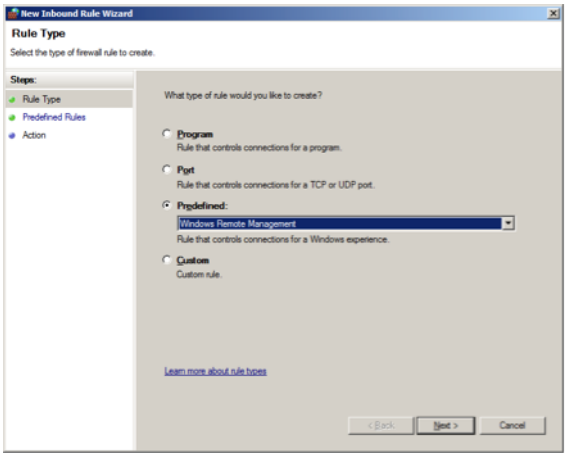


2. Add firewall exceptions to allow the service to communicate.

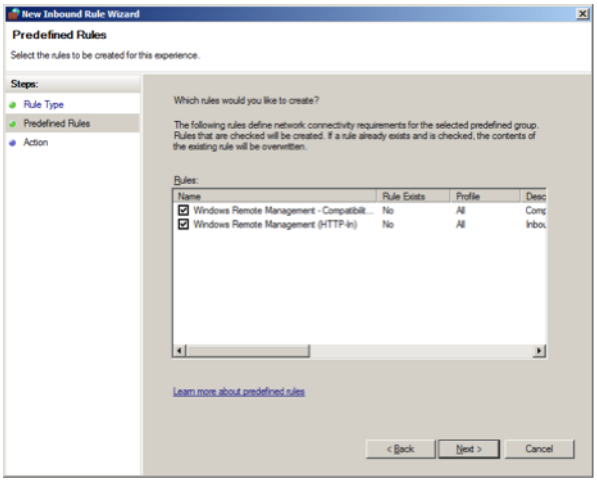
- Go to **Computer Configuration** -> **Policies** -> **Windows Settings** -> **Security Settings** -> **Windows Firewall with Advanced Security** .
- Right click on **Windows Firewall with Advanced Security** to create a new Inbound Rule.



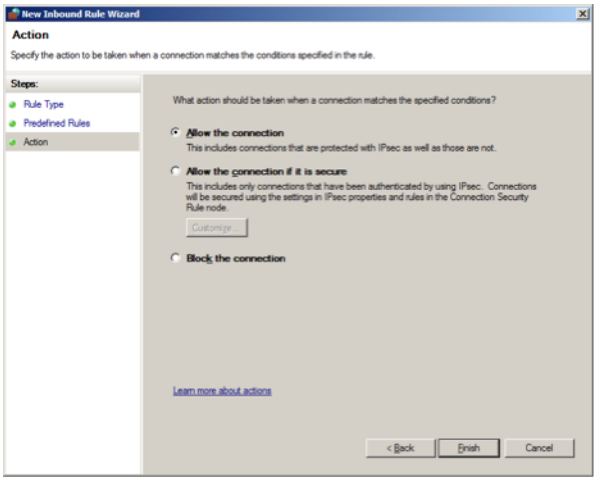
- Select the type of rule as **Predefined** as **Windows Remote Management** .



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



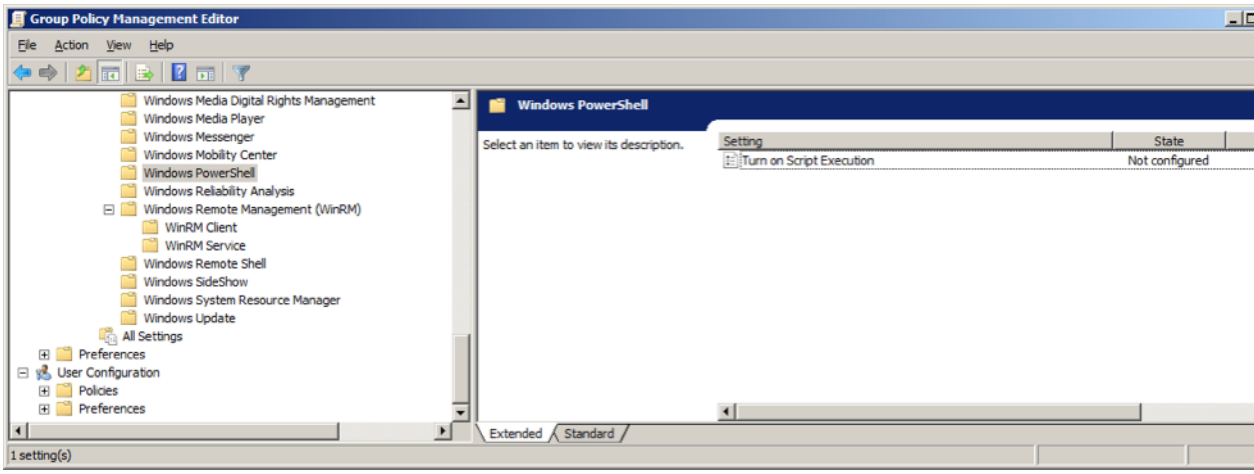
- Configure the **Action** as **Allow the connection** and click **Finish**.



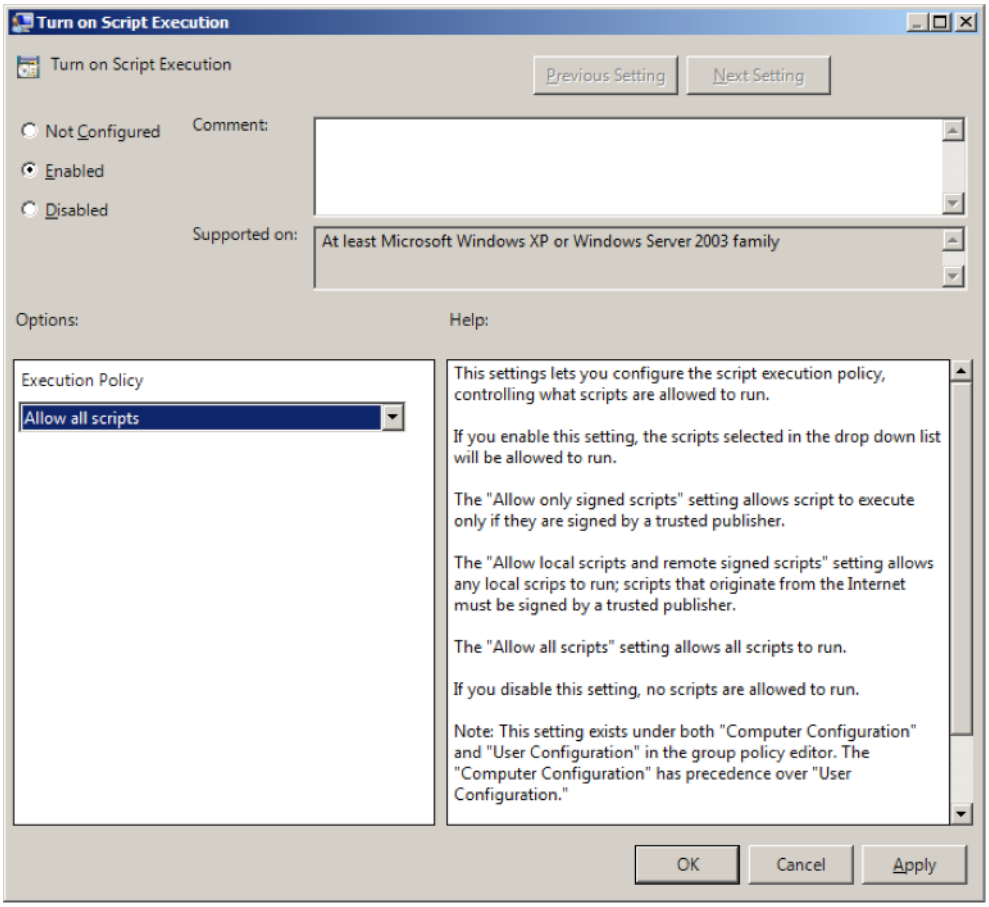
3. Set script execution policy.

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

   - Enable **Script Execution** .

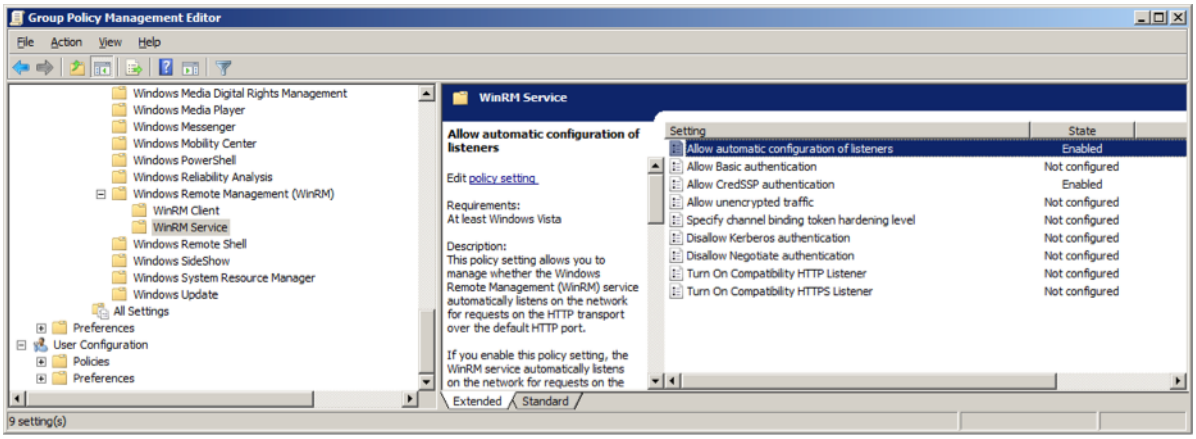


- Set Execution Policy to **Allow all scripts**.



4. Setup WinRM service.

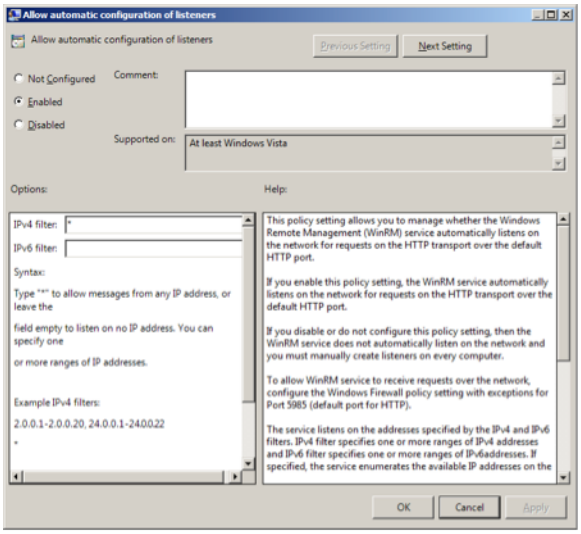
   - Go to **Computer Configuration** -> **Policies** -> **Administrative Templates** -> **Windows Components** -> **Windows Windows Remote Management (WinRM)** -> **WinRM Service**.



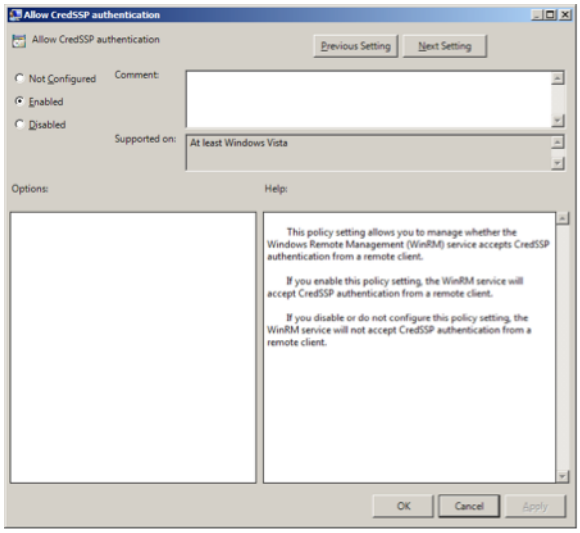
- Create a WinRM listener.

  a. To allow automatic configuration of listeners, select **Enabled**.

  b. Set **IPv4 filter** to * (all addresses or specify range)

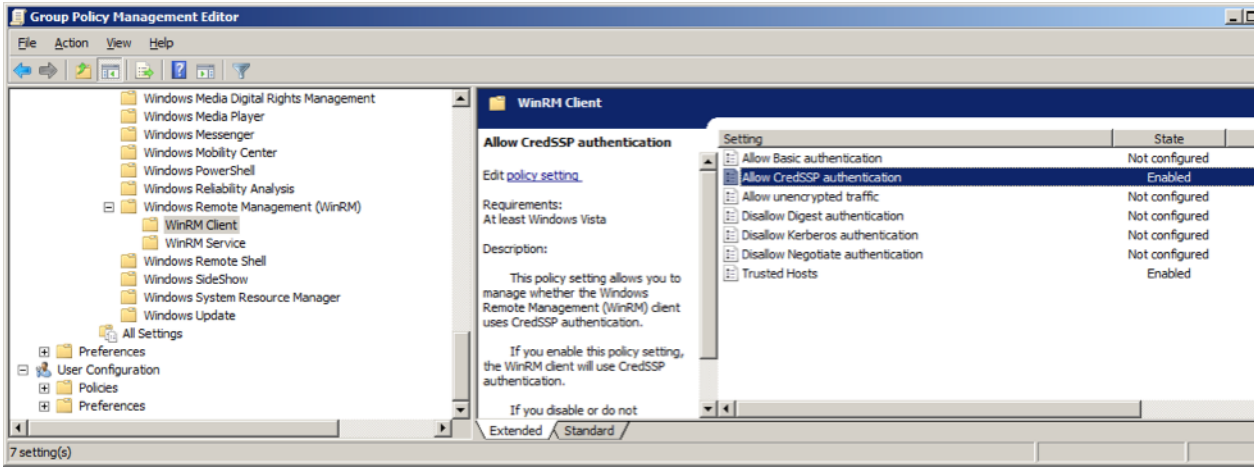


  c. Allow CredSSP authentication and click OK.

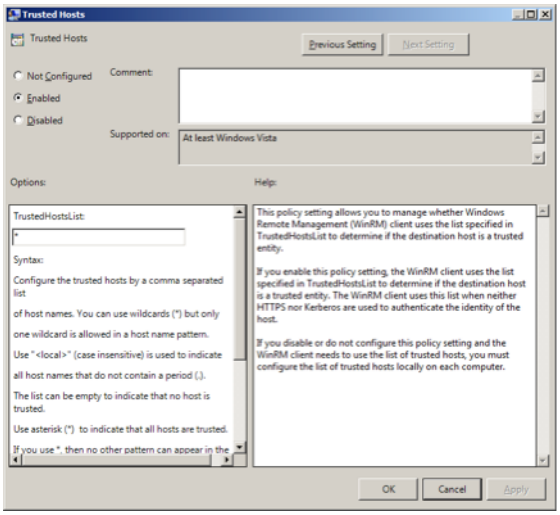


5. Setup WinRM client.

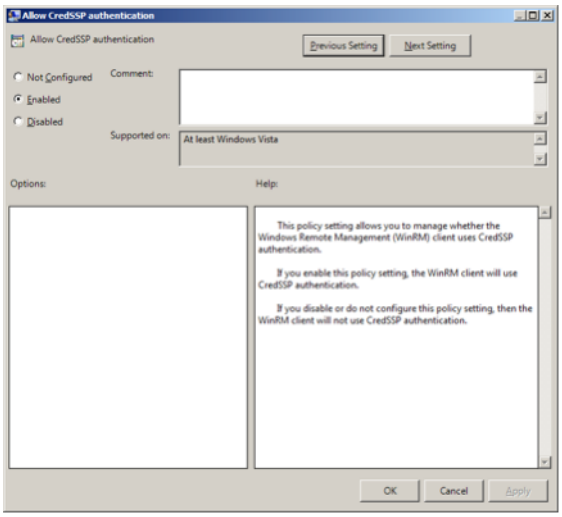
   - Go to **Computer Configuration** -> **Policies** -> **Administrative Templates** -> **Windows Components** -> **Windows Windows Remote Management (WinRM)** -> **WinRM Client**.



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

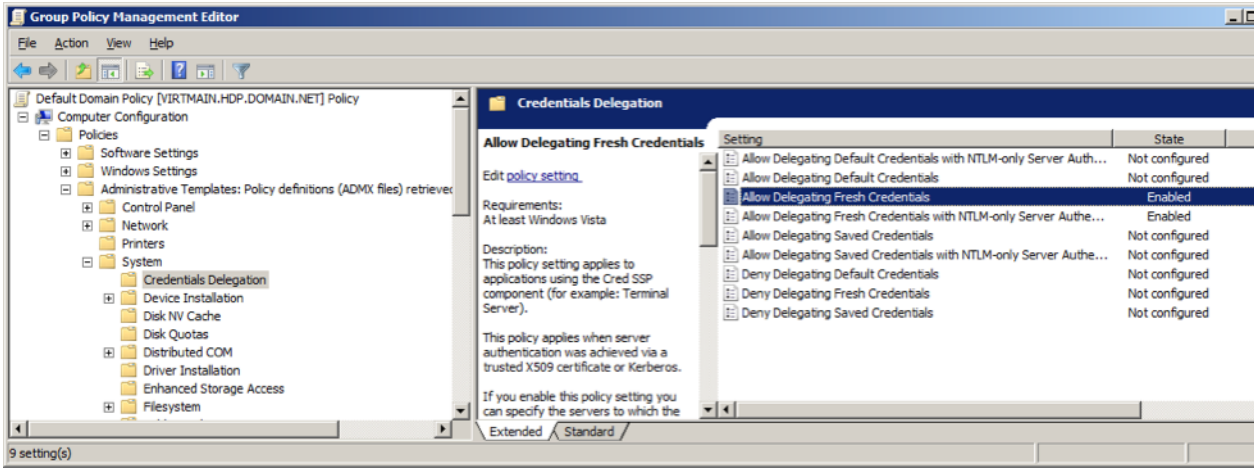


- Allow CredSSP authentication and click OK.

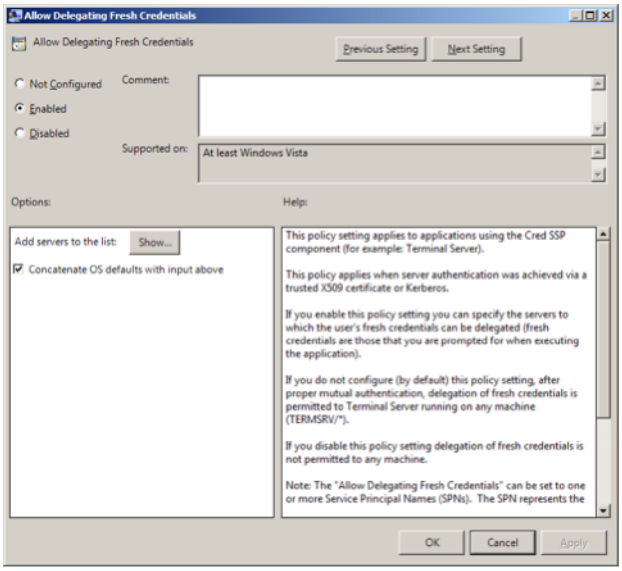


6. Enable credentials delegation.

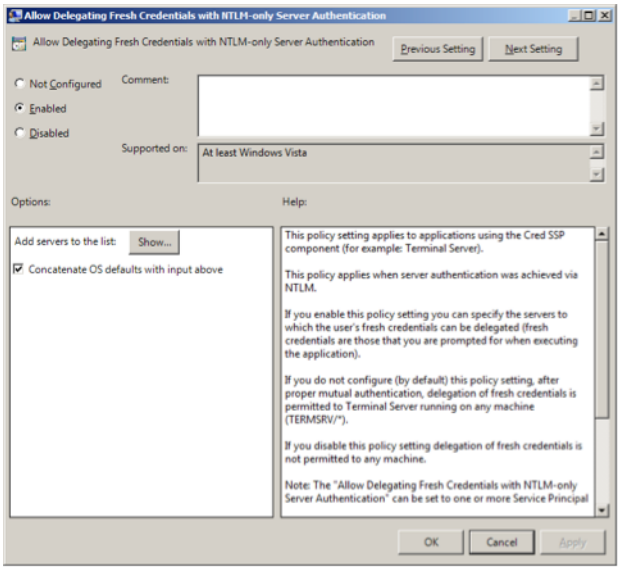
   - Go to **Computer Configuration** -> **Policies** -> **Administrative Templates** -> **System** -> **Credentials Delegation**.



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

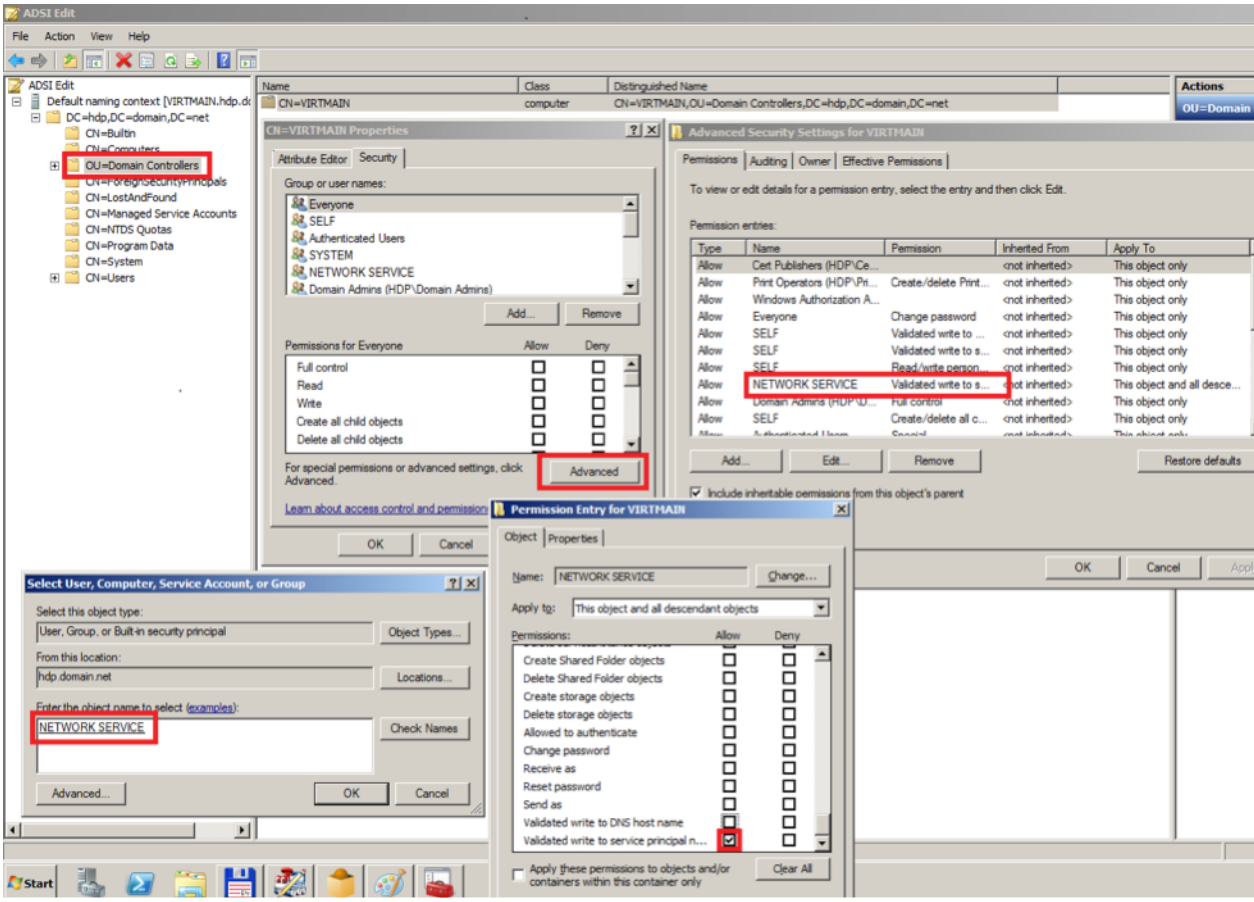


- Select **Enabled** to allow delegation fresh credentials with NTLM-only server authentication.
- Under **Options** click on **Show**. Set **WSMAN** to * (all addresses or specify range). Click on **Finish**.



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
     ```

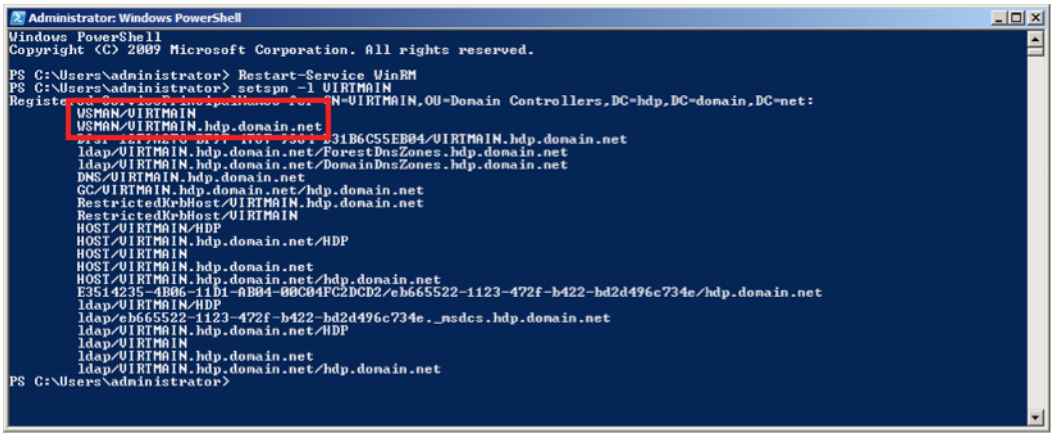
   - On other hosts in domain, execute the following commands:

     ```
     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
     ```

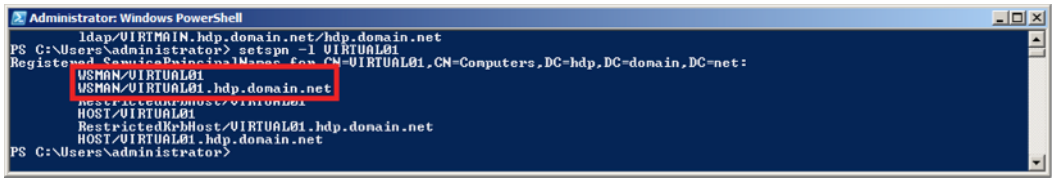
     You should see output similar to the following:



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:



1.6. Define Cluster Configuration

The Hortonworks Data Platform consists of multiple components. These components are installed across the cluster. The cluster properties file specifies the directory locations and node host locations for each of the components. The following section outlines how to construct the cluster properties file to define the cluster blueprint for all HDP components that need to be installed.

Use the following instructions to configure HDP installer for your cluster:

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

Important

- Ensure that all the properties in the `clusterproperties.txt` file are separated by a new line character.
- Ensure that the directory paths do not contain any whitespace character.

  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 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.

Table 1.6. Configuration values for MSI installer

Configuration Property Name	Description	Example value	Mandatory/ Optional/ Conditional
HDP_LOG_DIR	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\logs`	Mandatory
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\data`	Mandatory
NAMENODE_HOST	The FQDN for the cluster node that will run the NameNode master service.	`NAMENODE_MASTER.acme.com`	Mandatory
SECONDARY_NAMENODE_HOST	The FQDN for the cluster node that will run the Secondary NameNode master service.	`SECONDARY_NN_MASTER.acme.com`	Mandatory
JOBTRACKER_HOST	The FQDN for the cluster node that will run the JobTracker master service.	`JOBTRACKER_MASTER.acme.com`	Mandatory
HIVE_SERVER_HOST	The FQDN for the cluster node that will run the Hive Server master service.	`HIVE_SERVER_MASTER.acme.com`	Mandatory
OOZIE_SERVER_HOST	The FQDN for the cluster node that will run the Oozie Server master service.	`OOZIE_SERVER_MASTER.acme.com`	Mandatory
WEBHCAT_HOST	The FQDN for the cluster node that will run the WebHCat master service.	`WEBHCAT_MASTER.acme.com`	Mandatory
FLUME_HOSTS	A comma separated list of FQDN for those cluster nodes that will run the Flume service.	`FLUME_SERVICE1.acme.com,FLUME_SERVICE2.acme.com,FLUME_SERVIC`	Mandatory
HBASE_MASTER	The FQDN for the cluster node that will run the HBase master.	`HBASE_MASTER.acme.com`	Mandatory
HBASE_REGIONSERVERS	A comma separated list of FQDN for those cluster	`slave1.acme.com, slave2.acme.com, slave3.acme.com`	Mandatory

Configuration Property Name	Description	Example value	Mandatory/ Optional/ Conditional
	nodes that will run the HBase Region Server services.		
SLAVE_HOSTS	A comma separated list of FQDN for those cluster nodes that will run the DataNode and TaskTracker services.	`slave1.acme.com, slave2.acme.com, slave3.acme.com`	Mandatory
ZOOKEEPER_HOSTS	A comma separated list of FQDN for those cluster nodes that will run the Zookeeper hosts.	`ZOOKEEEPER_HOST.acme.com`	Mandatory
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 `derby`	Mandatory
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 HIVE_SERVER_HOST.	`sqlserver1.acme.com`	Mandatory
DB_PORT	This is an optional property required only if you are using SQL Server for Hive and Oozie metastores. By default, 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`	Mandatory
HIVE_DB_USERNAME	User account credentials for Hive metastore database instance. Ensure that this user account has appropriate permissions.	`hive_user`	Mandatory
HIVE_DB_PASSWORD		`hive_pass`	Mandatory
OOZIE_DB_NAME	Database for Oozie metastore. If using SQL Server, ensure that you create the database on the SQL Server instance.	`ooziedb`	Mandatory
OOZIE_DB_USERNAME	User account credentials for Oozie metastore database instance. Ensure that this user account has appropriate permissions.	`oozie_user`	Mandatory
OOZIE_DB_PASSWORD		`oozie_pass`	Mandatory

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
```

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 on a machine.
   - 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=135
        ```

      - 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
3. Specify the configuration for HDP on a single node machine.

Create a `clusterproperties.txt` file. (The `clusterproperties.txt` is a text file and contains parameter definitions (like the hostnames of the nodes in your cluster, the roles for each of them, etc.).)

Start creating the `clusterproperties.txt` file by copying the example text provided at the bottom of this section and modify it according to the hostname of your machine

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

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

#Data directory
HDP_DATA_DIR=c:\hdp\data

#Hosts (Roles for the host machines in your cluster)
NAMENODE_HOST=${Hostname for your single node cluster}
SECONDARY_NAMENODE_HOST=${Hostname for your single node cluster}
JOBTRACKER_HOST=${Hostname for your single node cluster}
HIVE_SERVER_HOST=${Hostname for your single node cluster}
OOZIE_SERVER_HOST=${Hostname for your single node cluster}
WEBHCAT_HOST=${Hostname for your single node cluster}
FLUME_HOSTS=${Hostname for your single node cluster}
HBASE_MASTER=${Hostname for your single node cluster}
HBASE_REGIONSERVERS=${Hostname for your single node cluster}
ZOOKEEPER_HOSTS=${Hostname for your single node cluster}
SLAVE_HOSTS=${Hostname for your single node cluster}

#Database host
DB_FLAVOR=derby
DB_HOSTNAME=${Hostname for your single node cluster}

#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
```

More details on the `clusterproperties.txt` configuration options are available here.

4. Install and start HDP,

   a. Download the HDP for Windows MSI from here.

      Open a command prompt with Administrator priveliges and execute the MSI installer command:

```
 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>"
```

The following example illustrates the command, with parameters, to launch the installer:

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

As shown in the example above, the following command parameters should be modified to match your files and directotories:

- HDP_LAYOUT: Absolute path to cluster properties file
- HDP_DIR: Install directory for HDP
- DESTROY_DATA: Whether to preserve or delete existing HDP data

b. Start all HDP services on the single machine.

In a command prompt, navigate to the HDP install directory. This is the 'HDP_DIR' setting from the msiexec command.

Then execute 'start_local_hdp_services.cmd'.

```
 cd <$PATH_TO_HDP_DIR>
start_local_hdp_services
```

c. Validate the install by running the full suite of smoke tests.

```
Run-SmokeTests
```

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 zip folder 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 Configure the HDP Installer.

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

Ensure that you 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 [34]. 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 [35] 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 (to use Administrator privileges), 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 [34].

- Ensure that *<$MSI_PATH>* and *<$PATH_to_clusterproperties.txt_File>* arguments are specified as simple filenames (assume that the `msiexec` command will be run 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 /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>"
```

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 [35] 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 /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>"
```

   Ensure that you provide appropriate values for the following mandatory command line option:

   - **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 will reuse 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 will remove previous data and format 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 /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.

a. Use the instructions provided here to start the HDP Services.

b. 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.

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:

1. Copy existing `clusterproperties.txt` file from any host machine in your cluster.

2. 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>"
```

Ensure that you provide 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. Managing HDP on Windows

This section will describe how to manage HDP on Windows services.

4.1. 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:

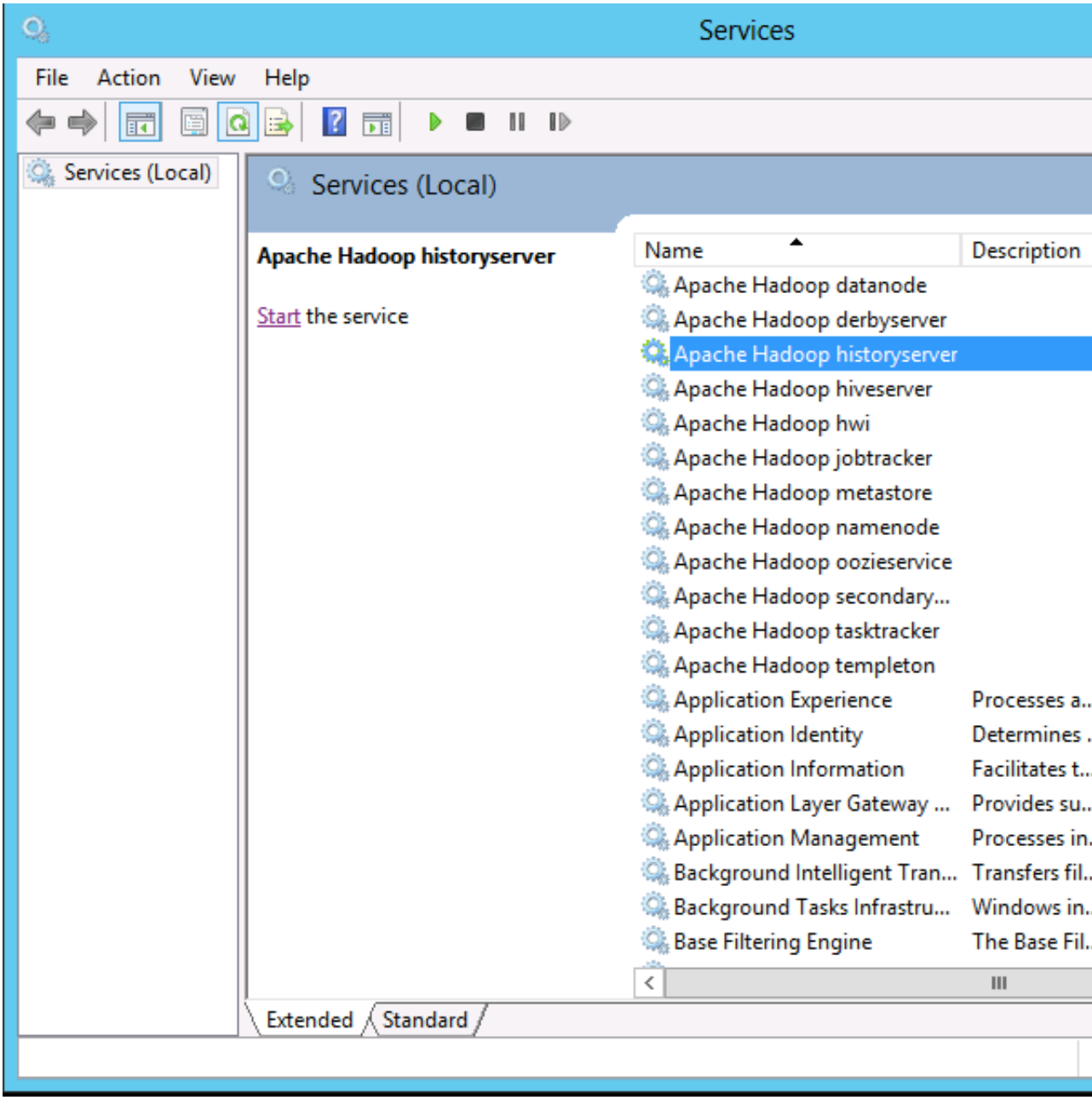
1. Start the HDP cluster.

   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
   ```

2. Open the **Services** administration pane.

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



4.2. 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
```

5. 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

5.1. 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 runnning 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 runnning 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
   ```

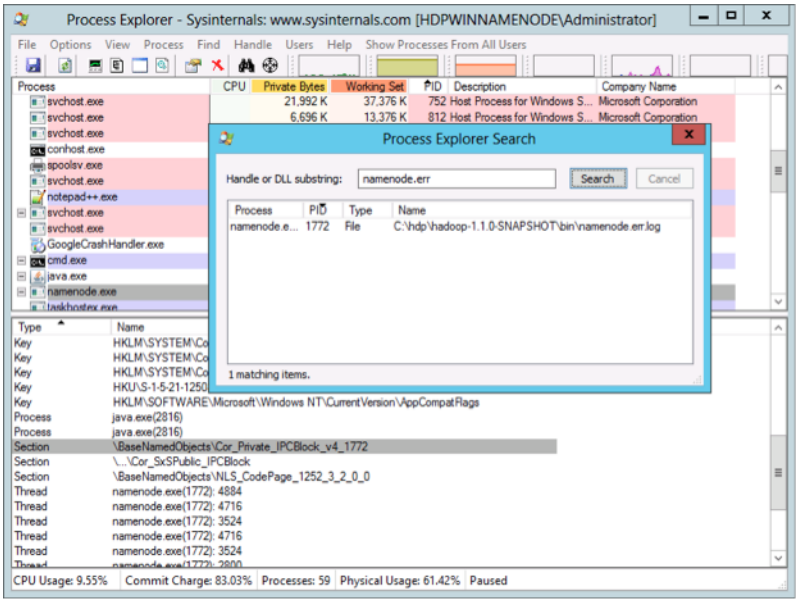
   For example, you should see output similar to the following:

   ```
   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:

   a. 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
   ```

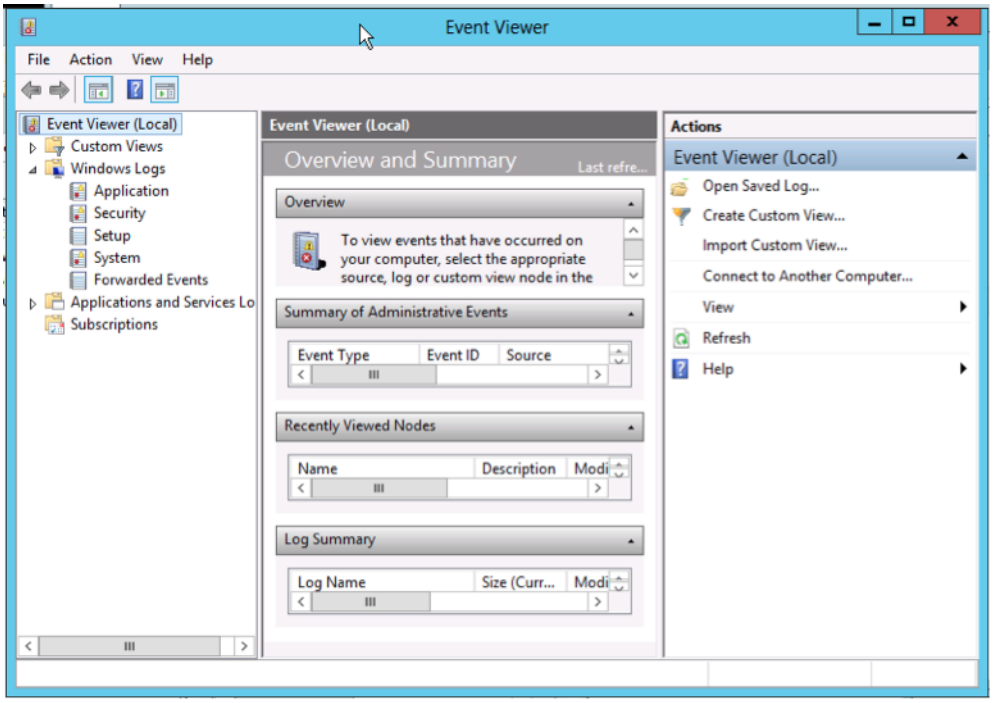
   You should see output similar to the following:

   ```
   DriveLetter       FileSystemLabel  FileSystem       DriveType
    HealthStatus        SizeRemaining              Size
   -----------       ---------------  ----------       ---------
    ------------        -------------               ----
                     System Reserved  NTFS             Fixed            Healthy
                     108.7 MB           350 MB
   C                                  NTFS             Fixed            Healthy
                     10.74 GB         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 issolate 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:

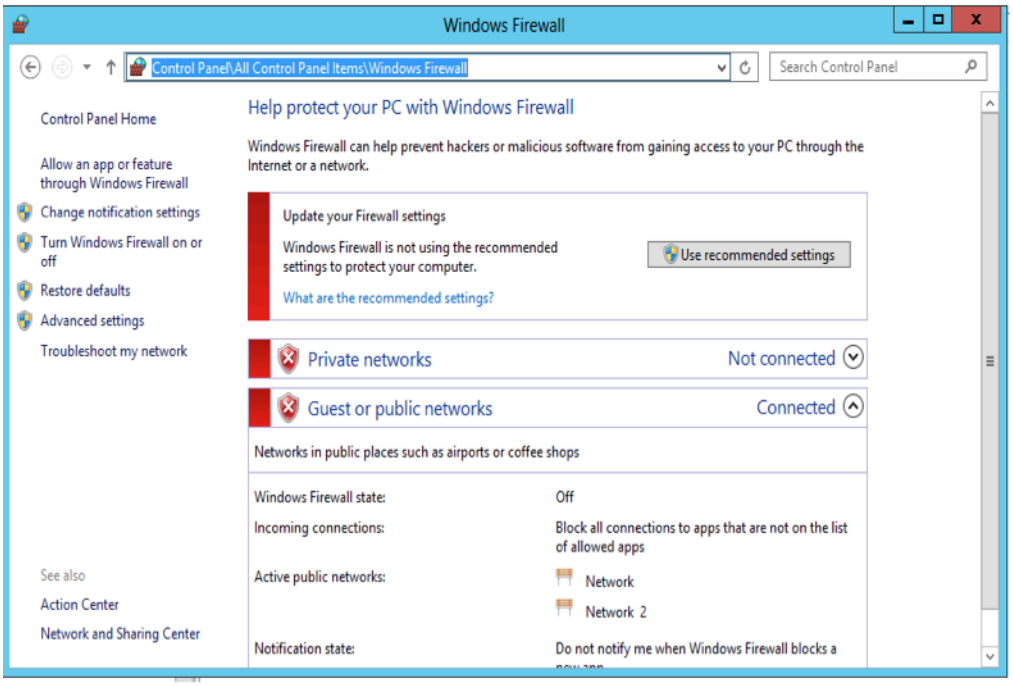
```
netstat -ano
```

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:



5.2. 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

5.2.1. 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.

5.2.2. 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 5.1. HDFS HTTP Ports

HDFS service	Configuration property in `hdfs-site.xml` file	Default value
NameNode	`<name>dfs.http.address</name>`	50070
DataNode	`<name>dfs.datanode.http.address</name>`	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 5.2. HDFS IPC Ports

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

5.2.3. 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 ouput 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
```

6. Uninstalling HDP

Choose one of the following options to uninstall HDP:

- Option I - Use Windows GUI
- Option II - Use command line utility

6.1. Option I - Use Windows GUI

1. Open the **Programs and Features** Control Panel Pane.
2. Select the program listed: **Hortonworks Data Platform 1.1.0 for Windows**.
3. With that program selected, click on the **Uninstall** option.

6.2. 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.