This guide describes a comprehensive procedure for installing, configuring, and building an IBM® WebSphere® Portal v8.0 cluster using:

- IBM WebSphere Application Server 8.0.0.3 – 64-bit
- Red Hat Enterprise Linux 5.0 update 5
- DB2 v9.7 fixpack 4
- IBM Tivoli Directory Server v6.3
- IBM HTTP Server 8.0
# Table of Contents

A Step-By-Step Guide to Configuring a WebSphere Portal v8.0 Cluster ........................................................................1
   Table of Contents .........................................................................................................................................................2
   Introduction .................................................................................................................................................................3
   Cluster Concepts ..........................................................................................................................................................5
   Using this Guide ..........................................................................................................................................................6
   Before you begin ..........................................................................................................................................................9

Main Guide ...................................................................................................................................................................10
   1 – Install IBM WebSphere Portal v8 on the Primary node .........................................................................................10
   2 - Configure the Primary Portal node to an external database ..................................................................................22
   3 - Create the WebSphere Portal profile template ....................................................................................................27
   4 - Install the Deployment Manager ........................................................................................................................29
   5 - Configure the Deployment Manager ....................................................................................................................41
   6 - Federate and Cluster the Primary Node ................................................................................................................45
   7 - Configure the Portal Cluster for Federated LDAP Security ...................................................................................49
   8 - Install an additional Portal Node .........................................................................................................................55
   9 - Federate and Cluster an additional Portal node ...................................................................................................66
  10 - Configure the Portal Cluster with an external web server ..................................................................................72

Appendix A – Alternate Setup Paths ..........................................................................................................................88
   A-1 - Installing WebSphere Portal and Deployment Manager on the same server .......................................................88
   A-2 – Creating a Deployment Manager profile on an existing Portal installation ......................................................103
   A-3 – Federating Portal to a Deployment Manager that has LDAP security enabled ..................................................109

Appendix B – Supplemental Information ..................................................................................................................116
   B-1 – Script to create and setup DB2 databases ...........................................................................................................116
   B-2 – Adding a Vertical Cluster member ....................................................................................................................120
   B-3 – Using the Configuration Wizard ....................................................................................................................125
       Tips for Configuration Wizard ..................................................................................................................................125
       Running ConfigEngine scripts using the Configuration Wizard ..........................................................................126
       Creating Workflows ..............................................................................................................................................129
       Using the ConfigTrace Log Viewer ........................................................................................................................133
   B-4 – How to properly extract the WebSphere Portal Installation media .................................................................135

About The Author ..........................................................................................................................................................137
Acknowledgments ..........................................................................................................................................................137
Change History ............................................................................................................................................................138
Introduction

Higher Versions of Portal and WebSphere Application Server
Although this guide is specifically written for 64-bit Portal v8.0 and WebSphere Application Server (WAS) v8.0.0.3, the same approach will apply to any Portal v8.0.x version or higher and any WAS v8.0.0.x version higher than 8.0.0.3, 32 or 64-bit.

Windows/Unix Differences
This guide was written using Linux as the base operating system, however the steps/concepts listed in this guide are independent of operating system.

The only significant difference is that for Windows, you must use the batch file commands instead of the UNIX shell commands listed in this guide. For example:

UNIX:  ./startServer.sh WebSphere_Portal

Windows:  startServer.bat WebSphere_Portal

or

UNIX:  ./ConfigEngine.sh cluster-node-config-cluster-setup

Windows:  ConfigEngine.bat cluster-node-config-cluster-setup

Database and LDAP examples
In the instructions for configuring Portal with the database and LDAP, screenshots show valid examples. Use values which are appropriate for your database and LDAP.

ConfigEngine vs Configuration Wizard
In this guide, I use the ConfigEngine.bat/sh script for all ConfigEngine tasks. A new and improved Configuration Wizard is included in WebSphere Portal v8 and you are welcome to use that instead to run any ConfigEngine script found in this guide. Refer to Appendix B-3 for details on how to use the new Configuration Wizard in v8. You may find it easier to use that instead of editing property files and running command line scripts.
**Hostnames Used in this Guide**

To avoid confusion with my own hostnames, I've replaced each instance of the hostnames of my servers with a sample value that corresponds to the server it belongs to so that it may be easier to understand which server I'm referring to in my examples.

I use the following values:

Primary Portal Node - myprimaryportal.ibm.com  
Secondary Portal Node – mysecondaryportal.ibm.com  
Deployment Manager – mydmgr.ibm.com  
Database Server – mydbserver.ibm.com  
LDAP Server – myldapserver.ibm.com  
IBM HTTP Server – mywebserver.ibm.com
**Cluster Concepts**

**Server** – A Java Virtual Machine (JVM) that manages user applications (such as WebSphere Portal and Web Content Management).

**Node** – A logical grouping of one or more application servers. A node does not necessarily mean a single physical server.

**Cell** – A logical grouping of one more nodes.

**Cluster** – A logical grouping of one or more servers across one or more nodes. The servers are managed together and participate in workload management. Servers in a cluster share resources, such as applications. Multiple clusters can exist in a single cell, but a single cluster cannot exist across multiple cells.

Figure 1 – WebSphere Portal cluster with two nodes, each with three cluster members.
**Using this Guide**

There are many different ways to build a WebSphere Portal cluster. In this guide, I use what I consider the easiest approach to building a cluster with a remote Deployment Manager (DMGR) from scratch. This process includes all of the following chapters:

1. Installing the Primary Portal node
2. Configuring the Primary node for a remote database
3. Enabling Portal profiles
4. Installing the Deployment Manager on a separate server
5. Configuring the Deployment Manager
6. Federating and Clustering the Primary Node
7. Enabling LDAP Security
8. Installing an additional Portal Node
9. Federating and Clustering an additional Portal node
10. Configure a web server

This approach however, may not be appropriate for everyone. What if you need to put your Deployment Manager on the same server as your Primary Portal node? What if you already have LDAP security enabled on your Deployment Manager prior to creating a cluster? What if you want a vertical cluster instead?

Each of these variations require an alternate set of steps to follow. To accommodate this, I have included appendixes describing a few of the more common 'alternate paths'. This guide is designed so that you may seamlessly swap out an irrelevant section of the main guide, and replace it with the relevant Appendix.

For example, suppose you already have a Deployment Manager setup and configured with your LDAP. There is no need for you to follow Chapter 4 for installing the Deployment Manager; you already have one. There is also no need to re-enable LDAP security; it's already enabled in your DMGR. Instead you would take this approach using these chapters:

1. Installing the Primary Portal node
2. Configuring the Primary node for a remote database
3. Enabling Portal profiles
5. Configuring the Deployment Manager
A-2. Federating and Clustering the Primary Node when DMGR LDAP Security is enabled
8. Installing an additional Portal Node
9. Federating and Clustering an additional Portal node
10. Configure a web server

You would skip Chapters 4 for installing the DMGR and 7 for enabling LDAP Security. You would replace Chapter 6 with Appendix A-2 for Federating and Clustering the Primary Node when DMGR LDAP Security is enabled.
The following scenarios are covered in this Guide (along with the order of Chapters you need to swap or remove):

**Building a WebSphere Portal cluster using a remote DMGR all from scratch?**

Use these chapters:

1 – Install IBM WebSphere Portal v8 on the Primary node  
2 - Configure the Primary Portal node to an external database  
3 - Create the WebSphere Portal profile template  
4 - Install the Deployment Manager  
5 - Configure the Deployment Manager  
6 - Federate and Cluster the Primary Node  
7 - Configure the Portal Cluster for Federated LDAP Security  
8 - Install an additional Portal Node  
9 - Federate and Cluster an additional Portal node  
10 - Configure the Portal Cluster with an external web server

**Building a WebSphere Portal cluster using a local DMGR all from scratch?**

**NOTE:** This is probably the quickest way to build a cluster if you are not concerned with using a remote Deployment Manager.

Use these chapters:

A-1 - Installing WebSphere Portal and Deployment Manager on the same server  
2 - Configure the Primary Portal node to an external database  
3 - Create the WebSphere Portal profile template  
6 - Federate and Cluster the Primary Node  
7 - Configure the Portal Cluster for Federated LDAP Security  
8 - Install an additional Portal Node  
9 - Federate and Cluster an additional Portal node  
10 - Configure the Portal Cluster with an external web server
Using an existing Deployment Manager that already has LDAP security enabled?

Use these chapters:

1 - Installing the Primary Portal node
2 - Configuring the Primary node for a remote database
3 - Enabling Portal profiles
A-3 - Federating and Clustering the Primary Node when DMGR has LDAP Security enabled.
8 - Installing an additional Portal Node
9 - Federating and Clustering an additional Portal node
10 - Configure a web server

Adding a local Deployment Manager after installing Portal?

Use these chapters:

1 - Installing the Primary Portal node
2 - Configuring the Primary node for a remote database
3 - Enabling Portal profiles
A-2 - Creating a DMGR profile on an existing Portal installation
6 - Federate and Cluster the Primary Node
7 - Configure the Portal Cluster for Federated LDAP Security
8 - Installing an additional Portal Node
9 - Federating and Clustering an additional Portal node
10 - Configure a web server


**Before you begin**

This guide does **NOT** cover the following:

- Installing DB2
- Installing IBM Tivoli Directory Server
- Configuring the cluster with Web Content Management
- Creating multiple clusters in a single cell
- Advanced Security configuration

For more information on these and other topics, please visit the IBM WebSphere Portal v8.0 Product Documentation:


To perform the tasks described in this document, you need basic WebSphere Portal and WebSphere Application Server knowledge and administration skills. Some steps might require the assistance of another system administrator, such as the database administrator or LDAP administrator.

The following references to WebSphere Portal and WebSphere Application Server file paths will be used throughout the guide:

- `<AppServer root>` - The root path of the AppServer directory, for example: `/opt/WebSphere/AppServer`
- `<PortalServer root>` - The root path of the PortalServer directory, for example: `/opt/WebSphere/PortalServer`
- `<wp_profile>` - The root path of the wp_profile directory, for example: `/opt/WebSphere/wp_profile`
- `<dmgr_profile>` - The root path of the dmgr profile directory, for example: `/opt/WebSphere/AppServer/profiles/Dmgr01`
- `<plugin root>` - The root path of the WebSphere Plugin directory, for example: `/opt/WebSphere/Plugins`
1 – **Install IBM WebSphere Portal v8 on the Primary node**

In this section, you will install the IBM Installation Manager and WebSphere Portal on the server you intend to use as your primary portal server.

Before installing WebSphere Portal, please ensure you review the Planning documentation:

http://www-10.lotus.com/ldd/portalwiki.nsf/dx/Planning_to_install_WebSphere_Portal_wp8

In this guide, the installation was completed as the 'root' user using installation images on a network drive.

**NOTE:** If you have downloaded the Portal media from Passport Advantage, refer to Appendix B-4 for instructions on how to properly extract the downloaded images.

1. Open a terminal window and enter:

   ```
   ping yourserver.yourcompany.com
   ```

   where yourserver.yourcompany.com is your actual fully qualified hostname.

2. In the same terminal window, enter:

   ```
   ping localhost
   ```

   to verify the "localhost" network settings are configured properly on your machine.

3. **Linux/UNIX environments only.** Ensure ulimit -n is set to 10240 or higher.

   ```
   ulimit -n 10240
   ```

4. From the WebSphere Portal v8 Setup DVD or directory, run the following command:

   ```
   ./setup.sh
   ```
5. When the setup wizard launches, select 'Install Portal':

6. Choose the Installation option that is appropriate for your environment. In this guide, we will select 'Install IBM WebSphere Portal from the network'.

7. You will see a prompt for the network location. Point this to the Setup/Repository directory and click OK.
8. If IBM Installation Manager is already installed and upgraded to the level Portal requires (v1.5.2), Installation Manager will launch. You can skip to Step 15.

If IBM Installation Manager is not already installed or at the level Portal requires, you will be prompted to install/upgrade it:

9. Click Next.

10. Accept the License Agreement and click Next.
11. Choose an installation directory for IBM Installation Manager:

12. Click Next.

13. On the Summary screen, click Install to begin the installation.

14. Once Installation completes, click Restart Installation Manager.
15. When Installation Manager launches, you should see this screen:

![IBM Installation Manager](image.png)

16. Go to File → Preferences → Repositories

17. Add the repositories for the Setup directory location. These should each point to the following location:

   `<Portal Media root>/Setup/eimage/repository.config`

   Doing this will tell IIM to automatically load the Portal directory, WAS directory, and Offering directory.

   Refer to Appendix B-4 if necessary to see how the Portal Media directory structure should be set up.

18. Click OK to save changes.

20. Check the boxes to install WebSphere Application Server and WebSphere Portal Server and WebSphere Portal Enable:

![Install Packages](image)

**Note:** This screen may vary depending on the Offering you are installing. In this example, we are installing Portal Enable, so we select both Server and Enable. If you were installing Extend, you would select both Server and Extend. If you were installing just Server, you would only select Server.

21. Click Next.
22. Check the box to install the required WebSphere Application Server fixes:

![Image of Installation Manager](image-url)

23. Accept the license agreement and click Next.

24. Select the location of the Shared Resources directory for Installation Manager and click Next:
25. Click IBM WebSphere Application Server to set the installation directory for WebSphere Application Server.

26. Click IBM WebSphere Portal Server to set the installation directory for WebSphere Portal Server.

27. Select any additional translations to install, if required. For this guide, no additional translations were selected.
28. Review the features to install for both WebSphere Application Server and WebSphere Portal. For this guide, all of the defaults were selected.

**NOTE:** Do not de-select any WebSphere Application Server features
**NOTE:** Ensure you install a WebSphere Portal profile (selected by default).
**NOTE:** Expand IBM WebSphere Application Server Network Deployment 8.0.0.3 → IBM Software Development Kit to select 32-bit or 64-bit WAS if desired.

29. Click Next
30. For the Profile Templates Type selection, select either Full or Base. For this guide, Base is used.

31. Click Next.
32. For Profile Configuration Details, set the Node Name, Cell Name, Administrator User ID and Administrator User Password.

**Optional**: If you select the Advanced Configuration radio button at the top of this screen (not shown), you can also set the Context Root, Default Home, Personalized Home, starting Port range, Profile Name, and Profile Path. For this guide, these were all left as the defaults but you are welcome to configure these as you see fit.

33. Click Next

34. Click Install to install the products.
35. When the installation completes, select None for 'Which program do you want to start?' and click Finish.

36. Verify you can access your Portal in a web browser:


At this point, you have successfully installed WebSphere Portal v8.0 with WebSphere Application Server 8.0.0.3.
2 - Configure the Primary Portal node to an external database

In this section, Portal will be configured to use an external database. For the purposes of this document, DB2 will be used as the external database with Type 4 drivers. This may vary in your environment. For more information about other databases that can be used with Portal, please visit the WebSphere Portal v8.0 Product Documentation for configuring external databases at this link and follow the instructions there as appropriate:

http://www-10.lotus.com/ldd/portalwiki.nsf/dx/Linux_clustered_server_Configuring_your_portal_to_use_a_database_wp8

In the environment used for this guide, 6 databases were created following the instructions in the Product Documentation:

RELDB
COMDB
CUSDB
JCRDB
FDBKDB
LMDB

In addition, the database administrator user “db2inst1” will be used as both the Configuration and Runtime user ID for each database.

If you choose to use DB2, the contents of the SQL file used to create and prepare the databases is included in Appendix B-1.

NOTE: In order to create the databases in DB2, you must be logged into the system as the database administrator.

1. From the primary Portal node, ensure the WebSphere_Portal and server1 servers are stopped by executing the following commands from the terminal window in the <wp_profile>/bin directory:

```
./stopServer.sh WebSphere_Portal -user <admin user> -password <admin pwd>
./stopServer.sh server1 -user <admin user> -password <admin pwd>
```
2. Ensure the database client is installed and configured on the node. Since we are using Type 4 drivers for DB2, all that is needed is to copy the db2jcc4.jar and db2jcc_license_cu.jar files from the DB2 server to some directory on the primary Portal server.

**NOTE:** For Portal v8.0, it is recommended that you place the Type 4 drivers into the following directory:

<wp_profile>/PortalServer/dbdrivers/

You will need to create a directory called 'dbdrivers'. This will save you the step of manually copying drivers over when adding future secondary nodes to your cluster.

3. Ensure the remote DB2 server is started.

4. From the <wp_profile>/ConfigEngine/properties directory, make a backup of the following files:

    wkplc.properties
    wkplc_dbtype.properties
    wkplc_dbdomain.properties

5. Edit the wkplc_dbtype.properties file and make the following changes:

    db2.DbDriver=com.ibm.db2.jcc.DB2Driver
    db2.DbLibrary=/opt/IBM/WebSphere/wp_profile/PortalServer/dbdrivers/db2jcc4.jar
    db2.JdbcProviderName=wpdbJDBC_db2

    **NOTE:** The entry for db2.DbLibrary is an example only. Please ensure this is a valid path on your system.

    **NOTE:** If using Windows, ensure the jar files in the DbLibrary path are separated by a semi-colon. Linux/Unix requires a colon.

6. Edit the wkplc_dbdomain.properties file and make the following changes:

    feedback.DbType=db2
    feedback.DbName=fdbkdb
    feedback.DbSchema=FEEDBACK
    feedback.DataSourceName=wpdbDS_feedback
    feedback.DbUrl=jdbc:db2://mydbserver.ibm.com:50000/fdbkdb:returnAlias=0;
    feedback.DbUser=db2inst1
    feedback.DbPassword=password
    feedback.DbRuntimeUser=db2inst1
    feedback.DbRuntimePassword=password
likeminds.DbType=db2
likeminds.DbName=lmdb
likeminds.DbSchema=likeminds
likeminds.DataSourceName=wpdbDS_likeminds
likeminds.DbUrl=jdbc:db2://mydbserver.ibm.com:50000/lmdb:returnAlias=0;
likeminds.DbUser=db2inst1
likeminds.DbPassword=password
likeminds.DbRuntimeUser=db2inst1
likeminds.DbRuntimePassword=password

release.DbType=db2
release.DbName=reldb
release.DbSchema=release
release.DataSourceName=wpdbDS_release
release.DbUser=db2inst1
release.DbPassword=password
release.DbRuntimeUser=db2inst1
release.DbRuntimePassword=password

community.DbType=db2
community.DbName=comdb
community.DbSchema=community
community.DataSourceName=wpdbDS_community
community.DbUrl=jdbc:db2://mydbserver.ibm.com:50000/comdb:returnAlias=0;
community.DbUser=db2inst1
community.DbPassword=password
community.DbRuntimeUser=db2inst1
community.DbRuntimePassword=password

customization.DbType=db2
customization.DbName=cusdb
customization.DbSchema=customization
customization.DataSourceName=wpdbDS_customization
customization.DbUrl=jdbc:db2://mydbserver.ibm.com:50000/cusdb:returnAlias=0;
customization.DbUser=db2inst1
customization.DbPassword=password
customization.DbRuntimeUser=db2inst1
customization.DbRuntimePassword=password

jcr.DbType=db2
jcr.DbName=jcrdb
jcr.DbSchema=jcr
jcr.DataSourceName=wpdbDS_jcr
jcr.DbUser=db2inst1
jcr.DbPassword=password
jcr.DbRuntimeUser=db2inst1
jcr.DbRuntimePassword=password
In steps 7 thru 15, we will set up Collation Support for the JCR database. This is only needed for DB2, and is recommended when the language locales of your users do not natively collate correctly in the DB2 database.

7. Copy the following files from the WebSphere Portal server to a temporary directory on the DB2 server:

   <PortalServer>/jcr/wp.content.repository.install/lib/wp.content.repository.install.jar
   <wp_profile root>/PortalServer/jcr/config/registerCollationUDFTemplate.sql

8. From the DB2 server, open a terminal window and change directories to:

   <db2 instance home>/sqllib/function

9. From the DB2 server, Execute the following command:

   <db2 instance home>/sqllib/java/jdk/bin/jar -xvf <temporary location>/wp.content.repository.install.jar

10. From the DB2 server, edit the <temporary location>/registerCollationUDFTemplate.sql file in a text editor.

11. Change all SCHEMA references in this file to the value you set for jcr.DbSchema in wkplc_dbdomain.properties. In this case, the schema value is 'jcr'.

12. Save the registerCollationUDFTemplate.sql file

13. From the DB2 server, connect to the JCR database by executing the following command in a terminal window:

    db2 connect to jcrdb user db2inst1 using password

14. From the same terminal window, execute the SQL script by running the following command:

    db2 -tvf <temporary location>/registerCollationUDFTemplate.sql

15. Disconnect from the JCRDB and restart the DB2 instance.
16. Switch over to the Primary Portal node, and from a terminal window, change directories to <wp_profile root>/ConfigEngine

17. Execute the following ConfigEngine scripts to validate the database properties:
   
   ```bash
   ./ConfigEngine.sh validate-database -DWasPassword=<password>
   ```

18. Execute the following ConfigEngine script to transfer the database from Derby to DB2:

   ```bash
   ./ConfigEngine.sh database-transfer -DWasPassword=<password>
   ```

19. **DB2 only.** After the database-transfer script completes, connect to each database and perform a reorg check to improve performance. You can do that following these steps:

   a. From the DB2 server, connect to the release database and execute the following command:

   ```bash
   db2 reorgchk update statistics on table all > reorgchk.txt
   ```

   b. Review the reorgchk.txt file and note any table names that have an * set in the REORG column.

   c. Execute the following command for each table name:

   ```bash
   db2 reorg table <tablename>
   ```

   d. After you have completed running reorg against all the marked tables for this database, execute the following DB2 commands to rebind the database:

   ```bash
   db2 terminate
   db2rbind <database name> -l db2rbind.out -u <db2admin ID> -p <db2admin password
   ```

   e. Repeat a-d for community, customization, jcr, likeminds and feedback.

20. Back on the Portal server, change directories to <wp_profile>/bin and execute the following command to start the Portal server:

    ```bash
    ./startServer.sh WebSphere_Portal
    ```

21. Verify that you can render Portal successfully in a web browser.


    At this point, you have successfully installed WebSphere Portal and configured it to use an external database.
3 - Create the WebSphere Portal profile template

In this section, you will create a backup of the primary node's wp_profile. You will also enable the Portal profile templates within the WebSphere Application Server Profile Management tool. This will allow you to create new Portal profiles in the future.

**Do not skip this section.** Completing this step is a prerequisite for configuring the Deployment Manager and secondary nodes, which we will do later in this guide.

**WARNING:** The ConfigEngine scripts in this section will write to the PortalServer root directory. By default, this directory is read/execute only (550). If you are using a non-root user, then this script may fail as a result. Before executing as a non-root user, give temporary Write access to the PortalServer root directory for this user. You can reset permissions after the scripts are completed.

1. Start the WebSphere_Portal server from the wp_profile/bin directory if it is not already started:
   ```bash
   ./startServer.sh WebSphere_Portal
   ```

2. Log in to the WebSphere Portal server and go to Administration → Search Administration → Manage Search → Search Collections

3. Click the Delete icon (trash can) for each search collection listed here.

![Manage Search](image-url)
4. Log out of WebSphere Portal

5. Stop the WebSphere_Portal server from the wp_profile/bin directory:

   ./stopServer.sh WebSphere_Portal -user <admin user> -password <admin password>


7. Change this property:

   jcr.textsearch.enabled=true

   to

   jcr.textsearch.enabled=false

8. Save icm.properties.

9. From a terminal window on the primary node, navigate to the <wp_profile root>/ConfigEngine directory.

10. Execute the following ConfigEngine script:

    ./ConfigEngine.sh enable-profiles -DWasPassword=<password>

    **NOTE:** This script will create a backup of your wp_profile configuration named Portal.car and save it to the following directory:

    <PortalServer root>/profileTemplates/default.portal/configArchives

    If you placed your database drivers within the wp_profile/PortalServer directory, then they will be automatically collected.

11. Execute the following ConfigEngine script to package all of the profile templates into a single zip file:

    ./ConfigEngine.sh package-profiles -DWasPassword=<password>

    **NOTE:** This will create a zip file called profileTemplates.zip in the following directory:

    <PortalServer root>/profileTemplates

At this point, the primary node has been installed and the profile templates have been created.
4 - **Install the Deployment Manager**

In this section, you will install the Deployment Manager on a separate server. All of the following steps will be completed on the server you intend to use as your deployment manager.

This installation will be completed using the WebSphere Portal installation media and will install from a network location.

1. From the WebSphere Portal v8 Setup DVD or directory, run the following command:
   ```
   ./setup.sh
   ```

2. When the setup wizard launches, select 'Install Portal':

   ![IBM WebSphere Portal](image)

   **Install IBM WebSphere Portal**

   The installation program installs all the software required to get IBM WebSphere Portal up and running.

   The IBM Installation Manager (IBM) is used to install IBM WebSphere Portal. If this program is already present on your computer, the existing copy will be used. Otherwise, a new copy of IBM will be installed first and that will then be used to complete the Portal installation.

   **Start Installation**

   - Install IBM WebSphere Portal from media
   - Install IBM WebSphere Portal from the network
   - Install IBM Installation Manager only

   Choose this option to install from a repository that has been made available on a network server.

3. Choose the Installation option that is appropriate for your environment. In this guide, we will select 'Install IBM WebSphere Portal from the network'.

4. You will see a prompt for the network location. Point this to the Setup/Repository directory and click OK.
5. If IBM Installation Manager is already installed and upgraded to the level WebSphere Application Server requires (v1.5.2), Installation Manager will launch. You can skip to Step 12.

If IBM Installation Manager is not already installed or at the level WAS requires, you will be prompted to install/upgrade it:

6. Click Next.

7. Accept the License Agreement and click Next.
8. Choose an installation directory for IBM Installation Manager:

9. Click Next.

10. On the Summary screen, click Install to begin the installation.

11. Once Installation completes, click Restart Installation Manager.
12. When Installation Manager launches, you should see this screen:

13. Go to File → Preferences → Repositories

14. Add the repositories for the Setup directory location. These should each point to the following location:

   `<Portal Media root>/Setup/eimage/repository.config`

   Doing this will tell IIM to automatically load the Portal directory, WAS directory, and Offering directory.

15. Click OK to save changes.

17. Since we are installing from the WebSphere Portal media, you will see both the WebSphere Application Server product and WebSphere Portal product in the selection screen.

Select **only** WebSphere Application Server and click Next.
18. Click the box to install the required WebSphere Application Server fixes and click Next:

![WebSphere Application Server installation screen](image)

19. Accept the license agreement and click Next.

20. Select the location of the Shared Resources directory and click Next.
21. Select the Installation Location for your Deployment Manager and click Next:

22. Select any additional languages you want to install and click next. For this guide, no languages were selected.

23. Select any additional features you want to install and click next. For this guide, I used the defaults.

24. On the summary screen, click Install to begin the installation.
25. When the installation completes, select the radio button to start the Profile Management Tool to create a profile and click Finish.

26. When the Profile Management Tool (WebSphere Customization Toolbox) launches, click 'Create...'.

27. Select the 'Management' profile type and click Next.

28. Select 'Deployment Manager' and click Next.

29. Select 'Typical' or 'Advanced' and click Next. For this guide, we will use 'Advanced'. This allows you to customize the node name, cell name, profile name and profile location (among other items).

30. Check the box to deploy the Administrative Console.
31. Select the name and location you would like to use for the Deployment Manager profile.

Important: Deleting the directory a profile is in does not completely delete the profile. Use the manageprofiles command to completely delete a profile.

The following naming rules must be used:
- Names must start and end with alphabetic characters (A-Z, a-z), numbers (0-9), and underscores (_) only.
- Names may contain alphabetic characters (A-Z, a-z), numbers (0-9), periods (.), dashes (-) and underscores (_) only.
- Names must not contain spaces or these characters: /\:*"<>?|\$\{\}()
32. Select the Node Name, Host Name, and Cell Name for your Deployment Manager. Click Next.

**NOTE:** Do not use the same Cell Name or Node Name as your primary Portal node.

33. **IMPORTANT.** Select the checkbox to Enable Administrative security and use the same user ID and password you used when installing Portal on your primary node. Doing this will save you some headaches when creating your cluster.

The cluster setup steps in this guide assume you have used the same ID. If you do NOT use the same ID, you may see unexpected problems when creating your cluster related to the user IDs.
34. Select your security certificates on the next two screens. For this guide, the defaults were used.

35. Change the port numbers if you'd like. For this guide, the port numbers were not changed.

   **NOTE:** Make note of the Administrative Console port and the SOAP port. Both of these ports will be used later in this guide.

36. **Windows/Linux only.** Select whether to run the DMGR as a service. For this guide, this option was not selected.

37. Click CREATE on the summary screen.

38. Click Finish to complete the DMGR profile creation.

At this point, the Deployment Manager has been installed and the DMGR profile has been created. The default URL for the Deployment Manager's Administrative Console is:

5 - Configure the Deployment Manager

In this section, you will configure the Deployment Manager and prepare it for the future Portal cluster.

1. From a terminal window on your Deployment Manager, navigate to `<dmgr_profile>/bin`

2. Ensure the Deployment Manager is stopped by executing the following command:
   
   ```
   ./stopManager.sh -user <admin user> -password <admin password>
   ```

3. From the primary Portal node, copy the following zip file over to a temporary location on your DMGR server:

   `<PortalServer root>/filesForDmgr/filesForDmgr.zip`

4. Back on the DMGR server, extract the `filesForDmgr.zip` that you just copied over into a temporary directory (`<temp>`).

5. **Remote DMGR only.** Copy the `<temp>/bin/ProfileManagement/plugins/com.ibm.wp.dmgr.pmt_7.0.5` directory to the `<DMGR AppServer root>/bin/ProfileManagement/plugins` directory.

6. **Remote DMGR only.** Copy the `<temp>/lib/wkplc.comp.registry.jar` and `wp.wire.jar` to the `<DMGR AppServer root>/lib` directory.

7. **Remote DMGR only.** Copy the `<temp>/plugins/com.ibm.patch.was.plugin.jar`, the `com.ibm.wp.was.plugin.jar`, and the `wp.base.jar` files to the `<DMGR AppServer root>/plugins` directory.

8. **Remote DMGR only.** Copy the `<temp>/profileTemplates/management.portal.augment` directory to the `<DMGR AppServer root>/profileTemplates` directory.

9. Copy the `<temp>/profiles/Dmgr01/config/.repository/metadata.wkplc.xml` file to the `<DMGR profile root>/config/.repository` directory.

   **NOTE:** The “.repository” directory is a hidden directory.
In steps 10 and 11, we will augment the DMGR profile. This process automatically makes the following changes to your DMGR profile:

- Increases the HTTP connection timeouts for the DMGR server
- Increases the SOAP connector timeout for JMX in the DMGR server
- Increases the JVM Maximum Heap size for the DMGR server
- Enables Application Security
- Creates a 'wpsadmins' group in the default file repository
- Adds your administrative user to the 'wpsadmins' group.
- Increases the soap timeout in the soap.client.props file.

10. From a terminal window on your DMGR server, change directories to `<DMGR AppServer root>/bin`

11. Execute the following command to augment the DMGR profile:

   ```
   ./manageprofiles.sh -augment -templatePath <DMGR AppServer root>/profileTemplates/management.portal.augment -profileName <dmgr_profile_name>
   ```

   Where `<DMGR AppServer root>` is the root path of AppServer on your DMGR server, and `<dmgr_profile_name>` is the name of your Deployment Manager profile, Dmgr01 for example.

12. Start the Deployment Manager from `<DMGR profile root>/bin` directory:

   ```
   ./startManager.sh
   ```

13. Launch the Deployment Manager administrative console and login. Default port is 9060:

   ```
   ```
14. Navigate to Security → Global Security

15. Under 'User Account Repository', click 'Configure':

![Global security configuration screen](image)
16. In the 'Primary administrative user name' field, change this value to the full distinguished name of the user using the following format:

uid=<user id>,o=defaultWIMFileBasedRealm

In my example, my user ID is “wpadmin”, therefore my full distinguished name will be:

uid=wpadmin,o=defaultWIMFileBasedRealm

**NOTE:** This change will help prevent user ID conflicts when we add the federated LDAP.

17. Before saving, enter the password for this user when prompted, then save all changes.

18. Restart the Deployment Manager for the changes to take effect.

At this point, your Deployment Manager is configured and ready for Portal federation.

**IMPORTANT:** This cluster guide uses the out of the box file repository for the security configuration to set up the cluster. If your Portal server is configured for a different type of security (such as an LDAP), then you **must** configure your Deployment Manager to use the **exact same user repository** as your Portal node. Once the Portal node is added to the Deployment Manager's cell, it will begin using the Deployment Manager's user repository. **If your Portal is configured for an LDAP and your DMGR is not, then your Portal will not function after adding it to the DMGR.** If you need to configure your DMGR for LDAP security, please do so now. This guide does not cover enabling the DMGR for LDAP at this stage of the cluster creation process.

We will configure the cluster to use an LDAP repository in a later section of this guide.
6 - Federate and Cluster the Primary Node
The next step is to federate and cluster the WebSphere Portal node. In this section, we will add the primary Portal node to the Deployment Manager cell and create the cluster. After the following steps have been completed, you will have a functional one node cluster.

1. Ensure the time on your Portal primary node is within 5 minutes of the time on your Deployment Manager (DMGR). Failure to do so will cause the addNode process to fail.

2. Ensure the DMGR is started. On the DMGR server, execute the following command from the <dmgr_profile>/bin directory:

   ./startManager.sh

3. Stop WebSphere_Portal and server1 by executing the following commands from the <wp_profile root>/bin directory:

   ./stopServer.sh WebSphere_Portal -user <admin user> -password <admin pwd>
   ./stopServer.sh server1 -user <admin user> -password <admin pwd>

4. Execute the following command from the <wp_profile root>/bin to add the Portal node to the DMGR cell:

   ./addNode.sh <dmgr_hostname> <dmgr soap port> -username <dmgr admin ID> -password <dmgr user password> -includeapps

   For example:

   ./addNode.sh mydmgr.ibm.com 8879 -username wpadmin -password wppassword -includeapps

   **NOTE:** If you are not sure what your DMGR's soap port is, you can obtain it by logging into the DMGR and navigating to System Administration → Deployment Manager → Ports.

   **IMPORTANT:** If the addNode script fails for any reason, you must complete the following steps before running addNode again:

   a) Remove the node from the DMGR cell in case AddNode successfully completed that step before failing.

   b) Login to the DMGR and do the following (these may not exist, depending on where the failure occurred):

      i. Remove all Enterprise applications

      ii. Remove the WebSphere_Portal server definition

      iii. Remove the JDBC Provider information for WebSphere_Portal
5. Stop the deployment manager by issuing the following command from the `<dmgr profile>/bin` directory:

   ```
   ./stopManager.sh -user <admin user> -password <admin pwd>
   ```

6. Start the deployment manager by issuing the following command from the `<dmgr profile root>/bin` directory:

   ```
   ./startManager.sh
   ```

7. On the primary node, edit the `<wp_profile>/ConfigEngine/properties/wkplc.properties` file and ensure all of the following properties are set appropriately for your environment:

   ```
   WasUserid=<DMGR admin user ID>
   WasPassword=<DMGR admin password>
   PortalAdminPwd=<password>
   WasRemoteHostName=<fully qualified hostname of DMGR>
   WasSoapPort=<soap port for DMGR; default is 8879>
   ServerName=WebSphere_Portal
   PrimaryNode=true
   ClusterName=PortalCluster
   ```

   **NOTE:** For the primary node, you **must** leave `ServerName` as `WebSphere_Portal`. Do not change it to any other value.

8. Edit `<wp_profile>/ConfigEngine/properties/wkplc_dbdomain.properties` and ensure all database user IDs and passwords are accurate.

9. Update the deployment manager configuration for the new WebSphere Portal server by executing the following ConfigEngine script:

   ```
   ./ConfigEngine.sh cluster-node-config-post-federation
   -DWasPassword=<password>
   ```

10. Create the cluster definition and add the WebSphere_Portal server as a cluster member by executing the following ConfigEngine script:

    ```
    ./ConfigEngine.sh cluster-node-config-cluster-setup -DWasPassword=<password>
    ```
11. Ensure that the cluster definition was created correctly by logging into the DMGR Admin Console and browse to Server -> Clusters -> WebSphere Application Server Clusters. An entry for your Portal cluster should be present.

In Steps 12-16, you will enable Session Persistence for the primary cluster member.

12. While logged in to the DMGR, navigate to Servers → Server Types → WebSphere application servers → WebSphere_Portal → Session Management → Distributed Environment Settings

13. Click the blue link for 'Memory-to-memory replication':

![DMGR Admin Console Screenshot](image)
14. In the Replication Domain drop-down menu, select the one for your cluster (e.g. PortalCluster).

15. In the Replication Mode drop-down menu, select 'Both client and server'.

16. Click OK and Save all changes.

17. Restart the DMGR, NodeAgent, and WebSphere_Portal server.

18. Verify Portal is functional by accessing it in your web browser:

At this point you have successfully completed building a one-node cluster using the out of the box security configuration. In the remaining sections, we will configure the Portal cluster with a federated ldap, add an additional horizontal node to the cluster, and configure a web server with the cluster.
7 - Configure the Portal Cluster for Federated LDAP Security

This section covers adding a federated LDAP Server to the cluster's security configuration. For more details about LDAP/Security configuration, please refer to the Product Documentation:

http://www-10.lotus.com/ldd/portalwiki.nsf/dx/Configuring_WebSphere_Portal_to_use_a_user_registry_on_Linux_in_a_clustered_environment_wp8

In this guide, we will configure security in our cluster to a non-SSL federated ldap server using IBM Tivoli Directory Server v6.3.

1. From the primary node, edit the wp_add_federated_ids.properties file in the <wp_profile>/ConfigEngine/config/helpers directory.

   **NOTE:** Helper files for other LDAP types exist in this directory as well.

2. Modify the following properties in this helper file to match your LDAP configuration. The values used in this guide are listed below:

   ```
   federated.ldap.id=PortalLdap
   federated.ldap.host=myldapserver.ibm.com
   federated.ldap.port=389
   federated.ldap.bindDN=uid=wpbind,cn=users,dc=ibm,dc=com
   federated.ldap.bindPassword=wpbind
   federated.ldap.ldapServerType=IDS
   federated.ldap.baseDN=dc=ibm,dc=com
   ```

   **NOTE:** The remaining properties were left as the default values for the purposes of this guide. If you need to modify these to suit your environment, please do so and refer to the Product Documentation link above as needed.

   **NOTE:** If your LDAP supports a membership attribute, such as ibm-allGroups for IBM Tivoli Directory Server, fill in one additional property as follows:

   ```
   federated.ldap.gc.name=ibm-allGroups
   ```

   Not all LDAPs support a membership attribute. This is an optional parameter, but one that can offer a significant performance enhancement if available in your LDAP. Check with your LDAP administrator to determine if your LDAP supports a membership attribute.
3. From a terminal window, change directories to the `<wp_profile>/ConfigEngine` directory and execute the following ConfigEngine script to validate the properties:

```
./ConfigEngine.sh validate-federated-ldap
-DparentProperties=<wp_profile>/ConfigEngine/config/helpers/wp_add_federated_ids.properties -DSaveParentProperties=true -DWasPassword=<password>
```

**NOTE:** By using the `-DparentProperties=<wp_profile>/ConfigEngine/config/helpers/wp_add_federated_ids.properties -DSaveParentProperties=true` flags, ConfigEngine will automatically save the properties from the helper file into the wkplc.properties file.

4. Execute the following ConfigEngine script to add the federated LDAP to the cluster security configuration:

```
./ConfigEngine.sh wp-create-ldap -DWasPassword=<current password>
```

**NOTE:** This script does not remove or replace the out-of-the-box file user registry. Instead, it adds the ldap to the security configuration, so that both it and the file user registry are in use. Your Portal Administrator User ID, Portal Administrator Group ID and WAS User ID are still in the default out-of-the-box file user registry.

5. Restart the DMGR, the nodeagent on the primary node, and the WebSphere_Portal server on the primary node.

6. **IMPORTANT:** If you happen to have a user in your ldap that shares the same shortname as your current Portal/WAS Administrator from the out-of-the-box-file registry, you will need to execute the following ConfigEngine script before proceeding with the remaining steps:

```
./ConfigEngine.sh wp-modify-realm-enable-dn-login -DWasPassword=<password>
```

Failure to run this script now can cause authentication problems for the remainder of these steps. **Again this is only needed if you have duplicated shortname IDs.**

For example, your original ID is:

```
uid=wpadmin,o=defaultWIMFileBasedRealm
```

and you have another 'wpadmin' ID in your LDAP:

```
uid=wpadmin,o=users,dc=mycompany,dc=com
```

If you try to login to Portal, you will be unable to login to Portal using the shortname. This will only be temporary and will be corrected at the end of these steps.
7. Execute the following ConfigEngine script to verify that all defined attributes are available in your newly added ldap:

    ./ConfigEngine.sh wp-validate-federated-ldap-attribute-config
    -DWasPassword=<current password>

**NOTE:** To manage the attributes, please refer to the following documentation:

    http://www-10.lotus.com/ldd/portalwiki.nsf/dx/Linux_cluster_Adapting_the_attribute_configuration_wp8

8. At this stage, your WebSphere Portal environment is using two user repositories: the out-of-the-box file registry, and the newly configured LDAP user registry. The WebSphere Application Server Administrator ID, the Portal Administrator User ID, and the Portal Administrator Group ID, are all configured for the file registry.

   Execute the following ConfigEngine script to reassign the WebSphere Application Server ID as a user within your LDAP:

    ./ConfigEngine.sh wp-change-was-admin-user -DWasPassword=<current password>
    -DnewAdminId=<full distinguished name from ldap> -DnewAdminPw=<ldap ID password>

   For example, this is the exact command I executed:

    ./ConfigEngine.sh wp-change-was-admin-user -DWasPassword=wpadmin
    -DnewAdminId=uid=wpadmin,cn=users,dc=ibm,dc=com -DnewAdminPw=wpadmin

   **NOTE:** If the full distinguished name of your user has a space in it, then add the 'newAdminId' and 'newAdminPw' values to your wkplc.properties file instead of passing them through the command line.

   **NOTE:** After running this script, the WasUserid value in wkplc.properties will be updated to reflect the new WAS User ID you specified for “newAdminId”.
9. Restart the DMGR, NodeAgent and WebSphere_Portal server for the change to take effect.

**NOTE:** When you stop these servers, you will need to pass in the user ID/pwd of the original WAS admin user. The new user will not take effect until the servers have been restarted.

**NOTE:** If you ran the 'wp-modify-realm-enable-dn-login' script, then you will be required to pass in the full distinguished name of the WAS admin user (since the servers are now using it) in order for authentication to succeed. For example:

```
./stopManager.sh -user uid=wpadmin,o=defaultWIMFileBasedRealm -password <password>
```

After the servers are restarted, the WasUserid and WasPassword will be the ldap user.

10. Execute the following ConfigEngine script to reassign the WebSphere Portal Administrator ID and Group ID to a user and group within your LDAP:

```
./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=<password> -DnewAdminId=<full distinguished name from ldap> -DnewAdminPw=<ldap ID password> -DnewAdminGroupId=<full distinguished name from ldap>
```

For example, this is the exact command I executed:

```
./ConfigEngine.sh wp-change-portal-admin-user -DWasPassword=wpadmin -DnewAdminId=uid=wpadmin,cn=users,dc=ibm,dc=com -DnewAdminPw=wpadmin -DnewAdminGroupId=cn=wpadmins,cn=groups,dc=ibm,dc=com
```

**NOTE:** If the full distinguished name of your user has a space in it, then add the 'newAdminId', 'newAdminPw', and 'newAdminGroupId' values to your wkplc.properties file instead of passing them through the command line.

**NOTE:** After running this script, the PortalAdminId value in wkplc.properties will be automatically updated to reflect the ID value specified for 'newAdminId' and the PortalAdminGroupId value will be automatically updated to reflect the 'newAdminGroupId'.

11. Restart the Deployment Manager, nodeagent, and WebSphere_Portal server on the primary node.

**NOTE:** At this point, your WasUserid, WasPassword, PortalAdminId, PortalAdminPw, and PortalAdminGroupId values will be your ldap user and group values.

**NOTE:** If you ran the 'wp-modify-realm-enable-dn-login' script, then you will be required to pass in the full distinguished name of the new LDAP WAS admin user (since the servers are now using it) in order for authentication to succeed. For example:

```
./stopManager.sh -user uid=wpadmin, cn=users, dc=ibm, dc=com -password <password>
```

52
12. Execute the following ConfigEngine script to list the current user repositories:

```
./ConfigEngine.sh wp-query-repository -DWasPassword=<password>
```

You should see output similar to this:

```
[wplc-query-federated-repository] Existing Federated Repositories
[wplc-query-federated-repository] Repository Name : {Details}
[wplc-query-federated-repository] ****************************************
[wplc-query-federated-repository] InternalFileRepository :
  {repositoryType=File, host=LocalHost}
[wplc-query-federated-repository] PortalLdap : {repositoryType=LDAP,
  specificRepositoryType=IDS, host=myldapserver.ibm.com}
[wplc-query-federated-repository] Status = Complete
```

In this example, I have two repositories:

- InternalFileRepository – The default file user registry
- PortalLdap – The newly added federated ldap

13. In the next steps, we will remove the default file user registry. This is required for production environments. While optional for other environments, it is strongly recommended you remove the file user registry anyway.

First, we need to ensure that new users and groups are created in your LDAP.

Edit the wkplc.properties file in `<wp_profile root>/ConfigEngine/properties` and set the following values (these examples are from my own environment. Ensure you use values that match your LDAP environment):

```
personAccountParent=cn=users,dc=ibm,dc=com
groupParent=cn=groups,dc=ibm,dc=com
personAccountRdnProperties=uid
groupRdnProperties=cn
```

14. Execute the following ConfigEngine script to ensure that new users and groups are created in your LDAP:

```
./ConfigEngine.sh wp-set-entitytypes -DWasPassword=<password>
```

15. Edit wkplc.properties again and set the following properties:

```
federated.delete.baseentry=o=defaultWIMFileBasedRealm
federated.delete.id=InternalFileRepository
```
16. Execute the following ConfigEngine script to remove the default file repository:

   ./ConfigEngine.sh wp-delete-repository -DWasPassword=<password>

17. If you executed the 'wp-modify-realm-enable-dn-login' script earlier, run the following ConfigEngine script to disable it and allow shortname logins to be functional again:

   ./ConfigEngine.sh wp-modify-realm-disable-dn-login -DWasPassword=<password>

18. Stop the NodeAgent and WebSphere_Portal server on this node.

19. Ensure the node is synchronized by executing the following command from the wp_profile/bin directory:

   ./syncNode.sh <dmgr hostname> <dmgr soap port> -user <WAS admin ID> -password <WAS admin password>

   For example:

   ./syncNode.sh mydmgr.ibm.com 8879 -user wpadmin -password wppassword

20. Restart the DMGR, NodeAgent, and WebSphere Portal servers.

    At this point, you have completed building a single node cluster using a remote database and federated LDAP server.
8 - Install an additional Portal Node

In this section, you will install the IBM Installation Manager and WebSphere Portal on the server you intend to use as your second portal server for the cluster.

Before installing WebSphere Portal, please ensure you review the Planning documentation:

http://www-10.lotus.com/ldd/portalwiki.nsf/dx/Planning_to_install_WebSphere_Portal_wp8

In this guide, the installation was completed as the 'root' user using installation images on a network drive.

1. Open a terminal window and enter:

   `ping yourserver.yourcompany.com`

   where yourserver.yourcompany.com is your actual fully qualified hostname.

2. In the same terminal window, enter:

   `ping localhost`

   to verify the “localhost” network settings are configured properly on your machine.

3. **Linux/UNIX environments only**. Ensure ulimit -n is set to 10240 or higher.

   `ulimit -n 10240`

4. From the WebSphere Portal v8 Setup DVD or network drive, run the following command:

   `./setup.sh`
5. When the setup wizard launches, select 'Install Portal':

6. Choose the Installation option that is appropriate for your environment. In this guide, we will select 'Install IBM WebSphere Portal from the network'.

7. You will see a prompt for the network location. Point this to the Setup/Repository directory and click OK.
8. If IBM Installation Manager is already installed and upgraded to the level Portal requires (v1.5.2), Installation Manager will launch. You can skip to Step 15.

If IBM Installation Manager is not already installed or at the level Portal requires, you will be prompted to install/upgrade it:

9. Click Next.

10. Accept the License Agreement and click Next.
11. Choose an installation directory for IBM Installation Manager:

12. Click Next.

13. On the Summary screen, click Install to begin the installation.

14. Once Installation completes, click Restart Installation Manager.
15. When Installation Manager launches, you should see this screen:

![IBM Installation Manager](image)

16. Go to File → Preferences → Repositories

17. Add the repositories for the Setup directory location. These should each point to the following location:

   `<Portal Media root>/Setup/eimage/repository.config`

   Doing this will tell IIM to automatically load the Portal directory, WAS directory, and Offering directory.

   Refer to Appendix B-4 if necessary to see how the Portal Media directory structure should be set up.

18. Click OK to save changes.
19. On the Installation Manager launch screen, click Install. Check the boxes to install WebSphere Application Server and WebSphere Portal Server and WebSphere Portal Enable:

![Installation Manager screen](image.png)

**Note:** This screen may vary depending on the Offering you are installing. In this example, we are installing Portal Enable, so we select both Server and Enable. If you were installing Extend, you would select both Server and Extend. If you were installing just Server, you would only select Server.

20. Click Next.
21. Check the box to install the required WebSphere Application Server fixes:

![Install Packages](image1)

22. Accept the license agreement and click Next.

23. Select the location of the SharedResources directory for Installation Manager

![Install Packages](image2)
24. Click Next.

25. Click IBM WebSphere Application Server to set the installation directory for WebSphere Application Server.
26. Click IBM WebSphere Portal Server to set the installation directory for WebSphere Portal Server.

27. Select any additional translations to install, if required. For this guide, no additional translations were selected.
28. Review the features to install for both WebSphere Application Server and WebSphere Portal. **De-select the option to create a Portal Server Profile.**

**NOTE:** Do not de-select any WebSphere Application Server features. **NOTE:** A WebSphere Portal profile will be created later on in this guide.

29. Click Next

30. Click Install to install the products.
31. When installation completes, select None for 'Which program do you want to start?’ and click Finish.

NOTE: While there is an option here to create a profile, the server is not able to create WebSphere Portal profiles yet so there is no need to select that option now.

At this point, you have successfully installed WebSphere Portal v8.0 with WebSphere Application Server 8.0.0.3.

NOTE: There is no profile on this system yet so there is no Portal you can access on this node.
9 - Federate and Cluster an additional Portal node

This section covers adding the additional node to the Deployment Manager cell and adding a new WebSphere_Portal server as a horizontal cluster member to the previously created cluster. Once this section is completed, you will have a functional two-node horizontal cluster using the federated LDAP security.

1. Copy `<PortalServer root>/profileTemplates/profileTemplates.zip` from the Primary Portal node to the newly created `<PortalServer root>/profileTemplates` directory on the additional node.

   **NOTE:** If you are using a non-root user, give this user temporary write access to the PortalServer directory.

2. Unzip the profileTemplates.zip file into the `<PortalServer root>/profileTemplates` directory on the additional node. Overwrite any duplicated files.

3. Update permissions on the profileTemplates directory by running the following command from the `<PortalServer root>` directory:

   ```
   chmod 755 -R profileTemplates
   ```

   **NOTE:** The 'chmod' command is only needed for Linux/Unix environments. It does not apply to Windows.

4. From the `<PortalServer root>/profileTemplates` directory, execute the following command:

   ```
   ./installPortalTemplates.sh <AppServer root>
   ```

   where `<AppServer root>` is the WebSphere Application Server root path on your system. For example:

   ```
   ./installPortalTemplates.sh /opt/IBM/WebSphere/AppServer
   ```
5. On the WebSphere Portal additional node, execute the following command from the 
<AppServer root>/bin/ directory to create the WebSphere Portal profile on this node:

```
./manageprofiles.sh -create -templatePath <PortalServer root>/profileTemplates/managed.portal -profileName <my_portal_profile> -profilePath <full path to profile> -cellName <cell_name> -nodeName <node_name> -hostName <hostname>
```

For example, if I wanted to create a profile called wp_profile with a cell name of node2Cell and a nodename of node2, I would run this command:

```
./manageprofiles.sh -create -templatePath /opt/IBM/WebSphere/PortalServer/profileTemplates/managed.portal -profileName wp_profile -profilePath /opt/IBM/WebSphere/wp_profile -cellName node2Cell -nodeName node2 -hostName mysecondaryportal.ibm.com
```

**NOTE:** Do NOT use the same node name as your primary node or any other node that may already be part of the DMGR cell. You will be unable to add this node to the DMGR cell if the node names are identical.

**NOTE:** Do NOT use the same cell name as the DMGR cell.

**IMPORTANT:** Do NOT use the option to Federate the profile now. This results in an unusable Portal profile.

**NOTE:** A WebSphere_Portal server will NOT be created during the profile creation. The WebSphere_Portal server will be created after the node is added to the existing cluster.

6. After creating the profile, edit the <PortalServer root>/wps.properties file on the system and add the ProfileName and ProfileDirectory properties to this file:

   `ProfileName=<your profile name>`
   `ProfileDirectory=<your profile directory including the profile name>`

   For example:

   `ProfileName=wp_profile`
   `ProfileDirectory=/opt/IBM/WebSphere/wp_profile`

   **NOTE:** You may need to temporarily give the OS user write access to this file.

   **NOTE:** For reference, you can compare this file to the same file on your primary node, but do NOT copy the file from the primary node.
7. If you did **NOT** place your database drivers in the `wp_profile/PortalServer` directory on your primary node BEFORE running the 'enable-profiles' script there, or if you are using Type 2 drivers, copy the database drivers to the new Portal node.

8. Ensure the Deployment Manager is started. Ensure that the time on the Deployment Manager server and the time on the additional Portal node server are no more than 5 minutes apart. In the next step, we will be federating the profile to the DMGR cell.

9. From the `<wp_profile root>/bin` directory, execute the following command:

   ```
   ./addNode.sh <dmgr_hostname> <dmgr soap port> -username <dmgr admin ID>
   -password <dmgr user password>
   ```

   For example:

   ```
   ./addNode.sh mydmgr.company.com 8879 -username wpadmin -password wppassword
   ```

10. Edit the `wkplc.properties` file in the `<wp_profile root>/ConfigEngine/properties` directory and ensure all of the following properties are set:

    ```
    WasUserid=<DMGR admin user ID>
    WasPassword=<DMGR admin password>
    PortalAdminPwd=<password>
    WasRemoteHostName=<fully qualified hostname of DMGR>
    WasSoapPort=<soap port for DMGR; default is 8879>
    ServerName=WebSphere_Portal_2
    PrimaryNode=false
    ClusterName=PortalCluster
    ```

    **NOTE:** For additional nodes, ServerName can be any value you want besides 'WebSphere_Portal'. This server will be created by the cluster-setup script.

    **NOTE:** Ensure ClusterName matches the value of the existing cluster.

11. Edit the `wkplc_dbdomain.properties` file in the `<wp_profile root>/ConfigEngine/properties` directory and ensure that the database password values are all set correctly.

    **NOTE:** This file should be pre-populated with your database information from running the 'enable-profiles' script on the primary node earlier.
12. Edit the wkplc_dbtype.properties file in the `<wp_profile root>/ConfigEngine/properties` directory and ensure that the `<dbType>.DbLibrary` value is valid for this system.

   **NOTE:** This file should be pre-populated with your database information from running the 'enable-profiles' script on the primary node earlier.

13. Ensure the NodeAgent is started on this node by running the following command from the `<wp_profile>/bin` directory:

   ```
   ./startNode.sh
   ```

14. Execute the following ConfigEngine script to create a second WebSphere_Portal cluster member:

   ```
   ./ConfigEngine.sh cluster-node-config-cluster-setup-additional -DWasPassword=password
   ```

   **NOTE:** This will automatically create a secondary cluster member to your existing cluster based on whatever value you set for ServerName.

15. Edit the `<profile root>/ConfigEngine/properties/wkplc.properties` file and set the following values:

   - `PortalAdminId=<your Full DN LDAP ID>`
   - `PortalAdminGroupId=<your Full DN LDAP ID>`

16. Execute the following ConfigEngine script to update the Portal Administrative user for the new cluster member with the LDAP administrative user:

   ```
   ./ConfigEngine.sh update-jcr-admin -DWasPassword=<password>
   ```
17. Log in to the DMGR Administrative Console and browse to:

   Servers -> Clusters -> WebSphere Application Server Clusters -> ClusterName
   -> Cluster Members

   An entry for WebSphere_Portal_2 should be available.

   ![Application servers screenshot]

   New port numbers have been assigned to the WebSphere_Portal_2 server. To check what ports
   are in use with this server, navigate to:

   Servers -> Server Types -> Application Servers -> WebSphere_Portal_2 -> Ports

   The WC_defaulthost is the port used to access Portal. The default port in this case is 10039.

   If you need to change these port numbers, you can do so from this screen.

   In Steps 18-23, you will enable Session Persistence for the new cluster member.

18. From the Cluster Members screen, click the link for your new cluster member.

19. Navigate to Session Management → Distributed Environment Settings
20. Click the blue link for 'Memory-to-memory replication':

![Image of Memory-to-memory replication settings]

21. In the Replication Domain drop-down menu, select the one for your cluster (e.g. PortalCluster).

22. In the Replication Mode drop-down menu, select 'Both client and server'.

![Image of Replication settings with selected modes]

23. Click OK and Save all changes.

24. Restart the DMGR, both nodeagents, and both WebSphere_Portal servers.

25. Verify functionality of the new additional node by accessing it in a web browser:

   http://mysecondaryportal.ibm.com:10039/wps/portal

At this point, you have successfully built a two-node WebSphere Portal cluster using a remote database and federated ldap security.
This section describes how to configure the Portal cluster with an external web server. For more details about web server configuration, please visit the WebSphere Portal Server Product Documentation at this link:

http://www-10.lotus.com/ldd/portalwiki.nsf/dx/Preparing_a_remote_Web_server_when_portal_is_installed_on_Linux_in_a_clustered_environment_wp8

In this guide, we will configure the Portal cluster with IBM HTTP Server v8.0.

**NOTE:** WebSphere Portal provides installation media for installing and configuring IBM HTTP Server. If you are using electronic images, refer to this page for your specific offering:

http://www-10.lotus.com/ldd/portalwiki.nsf/dx/Electronic_images_wp8

On your offering's page, find the table under **Optional Software** titled 'List of eAssembly images for IBM WebSphere Application Server V8.0.0.3'.

There are 4 images you will need, and these all should be extracted into the same parent directory (hereafter referred to as `<IHS media root>`).

If you are using DVDs, the DVDs will be titled: 'IBM WebSphere Application Server V8.0 Supplements'.

1. If you do not have IBM Installation Manager installed on the server you will use as your web server, we will install it first from the Portal Images/DVDs. From the Portal Setup Image/DVD, run the following command:

   ```bash
   ./setup.sh
   ```
2. When the Setup Wizard launches, select 'Install Portal'.

![Image of the Install IBM WebSphere Portal window]

- Select 'Install IBM Installation Manager Only.'
4. Click Next on the Package screen.

5. Accept the License Agreement and click Next.
6. Choose an Installation Directory for IIM and click Next.

7. Click Install to install IIM.

8. When the installation finishes, click Restart Installation Manager.
9. When Installation Manager launches, you should see this screen:

![IBM Installation Manager](image)

10. Go to File → Preferences → Repositories

11. Add the repositories for the IBM HTTP Server, WebSphere WebServer Plugin and WebSphere Customization Toolbox. This should point to the following location:

   `<IHS media root>/repository.config`

12. Click OK to save the changes.

14. Check the box for IBM HTTP Server for WebSphere Application Server, Web Server Plug-ins, and WebSphere Customization Toolbox. Click Next:

15. Accept the license agreement and click Next.

16. If no other products have been installed with Installation Manager on this server, you will see a screen to select the Installation directory for shared objects used by IIM. Choose a directory and click Next.
17. Click the Package Group Name for IBM HTTP Server and select an Installation Directory:

![IBM HTTP Server installation interface](image1)

18. Click the Package Group Name for 'Web Server Plug-Ins' and select an Installation Directory:

![Web Server Plug Ins installation interface](image2)
19. Click the Package Group Name for 'WebSphere Customization Toolbox' and select an installation directory, then click Next:

![IBM Installation Manager](image)

20. Select any additional features you'd like for the WebSphere Customization Toolbox and click Next. For this guide, the defaults were used.

![IBM Installation Manager](image)
21. Select the Port you'd like the web server to listen on and click Next:

![Configuration for IBM HTTP Server for WebSphere Application Server 8.0.0.3](image)

22. On the summary screen, ensure everything is correct and click Install to begin the installation.
23. Once the installation finishes, select the radio button for WebSphere Customization Toolbox and click Next:

24. When WebSphere Customization Toolbox loads, select 'Web Server Plug-ins Configuration Tool' and click 'Launch Selected tool':
25. In the 'Web Server Plug-in Runtime Location' window, click 'Add'.

26. Provide a Web Server name and the location of the WebSphere Plug-ins directory, and click Finish.

**NOTE:** In this example I just used 'webserver1' but it can be whatever you'd like.

27. In the Web server Plug-in Configurations window, click 'Create'.
28. Select your Web Server type and click Next. In this guide we are using IBM HTTP Server v8.0.

29. Specify the location of the httpd.conf file and the port you would like to use for the web server. Click Next.
30. Select to Setup IBM HTTP Server Administration Server if you'd like. For the purposes of this guide, this was NOT selected.

31. Give a name for your webserver, such as webserver1. Click Next.
32. On the next screen, put the hostname of your Deployment Manager and click Next:

![Configuration Scenario Selection](image)

33. Click 'Configure' to configure the plug-in.

34. When the process completes, de-select 'Launch the plug-in configuration roadmap' and click Finish.

35. Copy this file on your Web Server:

   `<Plugin root>/bin/configurewebserver1.sh`

   to your Deployment Manager server in the `<AppServer root>/bin` directory.

36. On your DMGR server, ensure that the dmgr is running by running this command from the `<dmgr profile root>/bin` directory:

   `./startManager.sh`
37. **Unix only.** Ensure you have execute permissions for the configurewebserver1.sh file in the `<AppServer root>/bin` directory:

    chmod 775 configurewebserver1.sh

38. Execute the following command from the `<AppServer root>/bin` directory:

    ./configurewebserver1.sh -profileName <dmgr profile name> -user <WAS user id> -password <WAS password>

    For example:

    ./configurewebserver1.sh -profileName Dmgr01 -user wpadmin -password wpadmin

39. After the script completes, log in to your Deployment Manager and navigate to Servers → Server Types → Web Servers

40. Check the box next to your web server name (e.g. webserver1) and click 'Generate Plug-in':

![Screenshot of the Deployment Manager with web servers and webserver1 highlighted.]

    **NOTE:** This will be written to the `<dmgr_profile>/config/cells/<cellname>/nodes/<nodename>/servers/webserver1/plugin-cfg.xml` file.

41. Copy the plugin-cfg.xml file to the remote web server at the following directory, overwriting the existing one:

    `<plugin_root>/config/webserver1`
42. Restart the DMGR, web server, and nodeagents, and WebSphere_Portal servers.

43. Verify that you can access the Portal cluster via the web server:

   http://mywebserver.hostname.com/wps/portal

Conclusion

In this guide, you saw how to build a fully functional WebSphere Portal v8.0.0 cluster using an external database and federated LDAP for security. You also saw how to configure a web server to allow for load balancing.
Appendix A – Alternate Setup Paths

A-1 - Installing WebSphere Portal and Deployment Manager on the same server

If you intend to install your Deployment Manager on the same instance as your WebSphere Portal primary node, you can follow these steps to install both at the same time.

This section replaces chapters 1, 4, and 5.

Before installing WebSphere Portal, please ensure you review the Planning documentation:

http://www-10.lotus.com/ldd/portalwiki.nsf/dx/Planning_to_install_WebSphere_Portal_wp8

In this guide, the installation was completed as the 'root' user using installation images on a network drive.

NOTE: If you have downloaded the Portal media from Passport Advantage, refer to Appendix B-4 for instructions on how to properly extract the downloaded images.

1. Open a terminal window and enter:

   \texttt{ping yourserver.yourcompany.com}

   where yourserver.yourcompany.com is your actual fully qualified hostname.

2. In the same terminal window, enter:

   \texttt{ping localhost}

   to verify the “localhost” network settings are configured properly on your machine.

3. \textbf{Linux/UNIX environments only}. Ensure ulimit -n is set to 10240 or higher.

   \texttt{ulimit -n 10240}

4. From the WebSphere Portal v8 Setup DVD or network drive, run the following command:

   \texttt{./setup.sh}
5. When the setup wizard launches, select 'Install Portal':

6. Choose the Installation option that is appropriate for your environment. In this guide, we will select 'Install IBM WebSphere Portal from the network'.

7. You will see a prompt for the network location. Point this to the Setup/Repository directory and click OK.
8. If IBM Installation Manager is already installed and upgraded to the level Portal requires (v1.5.2), Installation Manager will launch. You can skip to Step 15.

If IBM Installation Manager is not already installed or at the level Portal requires, you will be prompted to install/upgrade it:

9. Click Next.

10. Accept the License Agreement and click Next.
11. Choose an installation directory for IBM Installation Manager:

12. Click Next.

13. On the Summary screen, click Install to begin the installation.

14. Once Installation completes, click Restart Installation Manager.
15. When Installation Manager launches, you should see this screen:

16. Go to File → Preferences → Repositories

17. Add the repositories for the Setup directory location. These should each point to the following location:

   `<Portal Media root>/Setup/eimage/repository.config`

   Doing this will tell IIM to automatically load the Portal directory, WAS directory, and Offering directory.

   Refer to Appendix B-4 if necessary to see how the Portal Media directory structure should be set up.

18. Click OK to save changes.

20. Check the boxes to install WebSphere Application Server and WebSphere Portal Server and WebSphere Portal Enable:

![Installation Manager Screen](image)

**Note:** This screen may vary depending on the Offering you are installing. In this example, we are installing Portal Enable, so we select both Server and Enable. If you were installing Extend, you would select both Server and Extend. If you were installing just Server, you would only select Server.

21. Click Next.
22. Check the box to install the required WebSphere Application Server fixes:

![Installation Manager Interface](image)

23. Accept the license agreement and click Next.

24. Select the location of the SharedResources directory for Installation Manager

25. Click Next.
26. Click IBM WebSphere Application Server to set the installation directory for WebSphere Application Server.

27. Click IBM WebSphere Portal Server to set the installation directory for WebSphere Portal Server.
28. Select any additional translations to install, if required. For this guide, no additional translations were selected.

29. Review the features to install for both WebSphere Application Server and WebSphere Portal. To install the DMGR profile, check the box for 'Deployment Manager Profile augmented with WebSphere Portal'.

30. Click Next
31. For the Profile Templates Type selection, select either Full or Base. For this guide, Base is used.

32. Click Next.
33. For Profile Configuration Details, set the Node Name, Cell Name, Administrator User ID and Administrator User Password.

**Optional**: If you select Advanced Configuration (not shown in screenshot), you can also set the Context Root, Default Home, Personalized Home, starting Port range, Profile Name, and Profile Path. For this guide, these were all left as the defaults but you are welcome to configure these as you see fit.

34. Click Next
35. For the Deployment Manager profile configuration screen, pick a Node Name, Cell Name, Admin User ID and Password, and profile name and path.

**NOTE:** Do NOT use the same Cell and Node name you selected for your WebSphere Portal profile. Doing so will prevent you from being able to create a cluster with these profiles.

**NOTE:** Use the same user ID and password you set for the Portal profile. This will save you from some trouble when setting up the cluster later.

**NOTE:** The DMGR and Portal node are going to be on the same server so Host Name should be the same.

36. Click Install to install the products.
37. When installation completes, select None for 'Which program do you want to start?’ and click Finish.

38. Verify you can access your Portal in a web browser:


39. Start the Deployment Manager from <DMGR profile root>/bin directory:

   ./startManager.sh

40. Launch the Deployment Manager administrative console and login. Default port when the DMGR is installed on the Portal Server is 9061:


   **NOTE:** Remember the DMGR is on the primary Portal server, so the hostname is the same as Portal's.
41. Navigate to Security → Global Security

42. Under 'User Account Repository', click 'Configure':

![Global Security Configuration Page]
43. In the 'Primary administrative user name' field, change this value to the full distinguished name of the user using the following format:

uid=<user id>,o=defaultWIMFileBasedRealm

In my example, my user ID is "wpadmin", therefore my full distinguished name will be:

uid=wpadmin,o=defaultWIMFileBasedRealm

**NOTE:** This change will help prevent user ID conflicts when we add the federated LDAP.

44. Before saving, enter the password for this user when prompted, then save all changes.

45. Restart the Deployment Manager for the changes to take effect.

At this point, you have successfully installed WebSphere Portal v8.0 with WebSphere Application Server 8.0.0.3.

A Deployment Manager profile has been created and is now configured and ready for WebSphere Portal federation.
A-2 – Creating a Deployment Manager profile on an existing Portal installation

Suppose you have already installed WebSphere Portal and have decided to put the DMGR on the same server. You could create a Deployment Manager profile manually and configure it following Chapter 5 of this guide.

However, you can use Installation Manager to add and configure a Deployment Manager profile at the same time. This section will cover how to do that. These instructions assume that because you already have WebSphere Portal v8 installed, you also have Installation Manager installed.

1. Ensure that WebSphere_Portal is stopped from the wp_profile/bin directory:

   ./stopServer.sh WebSphere_Portal -user wpadmin -password wpadmin

2. Ensure that server1 is stopped from the cw_profile/bin directory:

   **NOTE:** cw_profile is the Configuration Wizard profile. For more details on this please see Appendix B-3 of this guide.

   ./stopServer.sh server1.sh -user wpadmin -password wpadmin

3. Launch IBM Installation Manager. From the <Installation Manager root>/eclipse directory, run this command:

   ./IBMIM

4. When Installation Manager launches, you should see this screen:
5. Go to File → Preferences → Repositories

6. Add the repository for the Portal media:

   Portal/repository.config

   You can add the Setup, WAS, and <Offering> repositories if you'd like, but they will not be used for this section.

7. Click OK to save changes.

8. Click the 'Modify' button on the Installation Manager main screen.

9. Select the WebSphere Portal package and click Next:
10. On the next screen, expand 'IBM WebSphere Portal Server 8.0.0.0' and select the checkbox for 'Deployment Manager augmented with WebSphere Portal':

![IBM Installation Manager](image)

**NOTE:** Do NOT de-select the Portal Server profile.

11. Set the DMGR hostname, nodename, cellname, user ID, password, profile name and profile Path.

**NOTE:** To make things easier on you when you create your cluster, use the exact same user ID and password that you used for the Portal installation.

**NOTE:** Use a unique nodename and cellname. Do NOT use the same nodename or cellname that you used for the Portal installation. This will cause the addNode process to fail later.
12. Review the summary screen. Make sure you are NOT inadvertently removing any features, such as your WebSphere Portal profile. If you see this, THIS IS BAD!

This is what you should see:

![Summary Screen Image]

If anything looks incorrect, go back and make any necessary corrections. If everything is correct, click Modify.

13. When the installation finishes, start the Deployment Manager from the $dmgr\ profile\ root/bin directory:
14. Launch the Deployment Manager administrative console and login. Default port when the DMGR is installed on the same server as Portal is 9061:


**NOTE:** Remember the DMGR is now installed on the primary Portal server, so the hostname is the same as Portal's.

15. Navigate to Security → Global Security

16. Under 'User Account Repository', click 'Configure':
17. In the 'Primary administrative user name' field, change this value to the full distinguished name of the user using the following format:

uid=<user id>,o=defaultWIMFileBasedRealm

In my example, my user ID is “wpadmin”, therefore my full distinguished name will be:

uid=wpadmin,o=defaultWIMFileBasedRealm

**NOTE:** This change will help prevent user ID conflicts when we add the federated LDAP.

18. Before saving, enter the password for this user when prompted, then save all changes.

19. Restart the Deployment Manager for the changes to take effect.

You have successfully created and augmented a Deployment Manager profile on a WebSphere Portal server. This appendix replaces chapters 4 for installing the Deployment Manager and 5 for Configuring the Deployment Manager.
A-3 – Federating Portal to a Deployment Manager that has LDAP security enabled

In the main guide, we enable LDAP security after federating the primary Portal node. This section covers the steps needed if your DMGR already has LDAP security enabled before you add your primary Portal node to it.

This section replaces chapters 4, 5, 6 and 7. This section assumes you know how to create a DMGR profile and enable LDAP security within it, as those steps are not covered here.

These steps can be applied to a DMGR with Standalone LDAP security or Federated LDAP security, it does not matter.

1. **Standalone LDAP security only.** If your DMGR has Standalone LDAP security enabled, you need to update Portal's wkplc.properties with the standalone ldap information. To do that, I used the helper file located here:

   `<wp_profile root>/ConfigEngine/config/helpers/wp_security_ids.properties`

   These were the properties I used:

   ```
   standalone.ldap.id=PortalLdap
   standalone.ldap.host=myldapserver.ibm.com
   standalone.ldap.port=389
   standalone.ldap.bindDN=uid=wpbind, cn=users, dc=ibm,o=com
   standalone.ldap.bindPassword=wpbind
   standalone.ldap.ldapServerType=IDS
   standalone.ldap.uidMap=*
   standalone.ldap.groupIdMap=*
   standalone.ldap.allGroupsMap=*
   standalone.ldap.serverId=uid=wpbind, cn=users, dc=ibm,o=com
   standalone.ldap.serverPassword=wpbind
   standalone.ldap.realm=PortalRealm
   standalone.ldap.primaryAdminId=uid=wpadmin, cn=users, dc=ibm,o=com
   standalone.ldap.primaryAdminPassword=wpadmin
   standalone.ldap.primaryPortalAdminId=uid=wpadmin, cn=users, dc=ibm,o=com
   standalone.ldap.primaryPortalAdminPassword=wpadmin
   standalone.ldap.baseDN=dc=ibm,o=com
   standalone.ldap.et.group.searchFilter=
   standalone.ldap.et.group.objectClasses=groupOfUniqueNames
   standalone.ldap.et.group.objectClassesForCreate=
   standalone.ldap.et.group.searchBases=cn=groups, dc=ibm,o=com
   standalone.ldap.et.personaccount.searchFilter=
   ```
standalone.ldap.et.personaccount.objectClasses/inetOrgPerson
standalone.ldap.et.personaccount.objectClassesForCreate=
standalone.ldap.et.personaccount.searchBases=cn=users,dc=ibm,o=com

standalone.ldap.gm.groupMemberName=uniqueMember
standalone.ldap.gm.objectClass=groupOfUniqueNames
standalone.ldap.gm.scope=direct
standalone.ldap.gm.dummyMember=uid=dummy

standalone.ldap.personAccountParent=cn=users,dc=ibm,o=com
standalone.ldap.groupParent=cn=groups,dc=ibm,o=com
standalone.ldap.personAccountRdnProperties=uid
standalone.ldap.groupRdnProperties=cn

I also altered one 'Advanced Properties' in the helper file and left the rest as the defaults, but you may find that you need to alter more for your LDAP.

standalone.ldap.gc.name=ibm-allGroups

2. **Standalone LDAP only.** Import the helper file contents into the wkplc.properties file by executing this ConfigEngine script from the `<wp_profile root>/ConfigEngine` directory:

```
./ConfigEngine.sh -DparentProperties=<wp_profile root>/ConfigEngine/config/helpers/wp_security_ids.properties -DSaveParentProperties=true
```

3. From a terminal window on your Deployment Manager, navigate to `<dmgr_profile>/bin`

4. Ensure the Deployment Manager is stopped by executing the following command:

```
./stopManager.sh -user <admin user> -password <admin password>
```

5. From the primary Portal node, copy the following zip file over to a temporary location on your DMGR server:

```
<PortalServer root>/filesForDmgr/filesForDmgr.zip
```

6. Back on the DMGR server, extract the filesForDmgr.zip that you just copied over into a temporary directory.

7. **Remote DMGR only.** Copy the `<temp>/bin/ProfileManagement/plugins/com.ibm.wp.dmgr.pmt_7.0.5` directory to the `<DMGR AppServer root>/bin/ProfileManagement/plugins` directory.

8. **Remote DMGR only.** Copy the `<temp>/lib/wkplc.comp registry.jar` and `wp.wire.jar` to the `<DMGR AppServer root>/lib` directory.
9. **Remote DMGR only.** Copy the `<temp>/plugins/com.ibm.patch.was.plugin.jar`, the `com.ibm.wp.was.plugin.jar`, and the `wp.base.jar` files to the `<DMGR AppServer root>/plugins` directory.

10. **Remote DMGR only.** Copy the `<temp>/profileTemplates/management.portal.augment` directory to the `<DMGR AppServer root>/profileTemplates` directory.

11. Copy the `<temp>/profiles/Dmgr01/config/.repository/metadata.wkplc.xml` file to the `<DMGR profile root>/config/.repository` directory.

   **NOTE:** The “.repository” directory is a hidden directory.

   In steps 12 and 13, we will augment the DMGR profile. This process automatically makes the following changes to your DMGR profile:
   
   - Increases the HTTP connection timeouts for the DMGR server
   - Increases the SOAP connector timeout for JMX in the DMGR server
   - Increases the JVM Maximum Heap size for the DMGR server
   - Enables Application Security
   - Increases the soap timeout in the soap.client.props file.

12. From a terminal window on your DMGR server, change directories to `<DMGR AppServer root>/bin`

13. Execute the following command to augment the DMGR profile:

    ```
    ./manageprofiles.sh -augment -templatePath <DMGR AppServer root>/profileTemplates/management.portal.augment -profileName <dmgr_profile_name>
    ```

    Where `<DMGR AppServer root>` is the root path of AppServer on your DMGR server, and `<dmgr_profile_name>` is the name of your Deployment Manager profile, Dmgr01 for example.

14. Ensure the time on your Portal primary node is within 5 minutes of the time on your Deployment Manager (DMGR). Failure to do so will cause the addNode process to fail.

15. Ensure the DMGR is started. On the DMGR server, execute the following command from the `<dmgr_profile>/bin` directory:

    ```
    ./startManager.sh
    ```
16. Stop WebSphere_Portal and server1 by executing the following commands from the `<wp_profile root>/bin` directory:

   ./stopServer.sh WebSphere_Portal -user <admin user> -password <admin pwd>
   ./stopServer.sh server1 -user <admin user> -password <admin pwd>

17. Execute the following command from the `<wp_profile root>/bin` to add the Portal node to the DMGR cell:

   ./addNode.sh <dmgr_hostname> <dmgr soap port> -username <dmgr admin ID>
   -password <dmgr user password> -includeapps

   For example:

   ./addNode.sh mydmgr.ibm.com 8879 -username wpadmin -password wppassword
   -includeapps

   **NOTE:** If you are not sure what your DMGR's soap port is, you can obtain it by logging into the DMGR and navigating to System Administration → Deployment Manager → Ports.

   **IMPORTANT:** If the addNode script fails for any reason, you must complete the following steps before running addNode again:

   a) Remove the node from the DMGR cell in case AddNode successfully completed that step before failing.

   b) Login to the DMGR and do the following (these may not exist, depending on where the failure occurred):
      i. Remove all Enterprise applications
      ii. Remove the WebSphere_Portal server definition
      iii. Remove the JDBC Provider information for WebSphere_Portal

18. Stop the deployment manager by issuing the following command from the `<dmgr profile>/bin` directory:

   ./stopManager.sh -user <dmgr admin user> -password <dmgr admin pwd>

19. Start the deployment manager by issuing the following command from the `<dmgr profile root>/bin` directory:

   ./startManager.sh
NOTE: Now that the node has been federated, it has inherited the DMGR's security configuration. The WebSphere Portal server will not function correctly yet so do not be surprised if you try to start Portal and cannot access it.

20. On the primary node, edit the `<wp_profile>/ConfigEngine/properties/wkplc.properties` file and ensure all of the following properties are set appropriately for your environment:

```properties
WasUserid=<DMGR admin user ID>
WasPassword=<DMGR admin password>
PortalAdminPwd=<Portal password>
WasRemoteHostName=<fully qualified hostname of DMGR>
WasSoapPort=<soap port for DMGR; default is 8879>
ServerName=WebSphere_Portal
PrimaryNode=true
ClusterName=PortalCluster
```

NOTE: For the primary node, you **must** leave `ServerName` as `WebSphere_Portal`. Do not change it to any other value.

21. Edit `<wp_profile>/ConfigEngine/properties/wkplc_dbdomain.properties` and ensure all database user IDs and passwords are accurate.

22. Update the deployment manager configuration for the new WebSphere Portal server by executing the following ConfigEngine script:

```
./ConfigEngine.sh cluster-node-config-post-federation -DWasPassword=<password>
```

23. Create the cluster definition and add the `WebSphere_Portal` server as a cluster member by executing the following ConfigEngine script:

```
./ConfigEngine.sh cluster-node-config-cluster-setup -DWasPassword=<password>
```
Ensure that the cluster definition was created correctly by logging into the DMGR Admin Console and browse to Server -> Clusters -> WebSphere Application Server Clusters. An entry for your Portal cluster should be present.

In Steps 25-29, you will enable Session Persistence for the primary cluster member.

25. Navigate to Servers → Server Types → WebSphere application servers → WebSphere_Portal → Session Management → Distributed Environment Settings

26. Click the blue link for 'Memory-to-memory replication':

27. In the Replication Domain drop-down menu, select the one for your cluster (e.g. PortalCluster).
28. In the Replication Mode drop-down menu, select 'Both client and server'.

![Image of Replication Mode selection](image)

29. Click OK and Save all changes.

30. At this point, the WebSphere Portal server will not function due to the change in security from the standalone node to the DMGR cell. The Portal administrative user and group must be updated to match a user ID and group ID in the DMGR's user repository. Execute the following ConfigEngine script:

```
./ConfigEngine.sh wp-change-portal-admin-user -DnewAdminId=<Portal admin ID in LDAP> -DnewAdminPw=<Portal admin password from LDAP> -DnewAdminGroupId=<Portal group ID in LDAP>
```

For example, this is what I used:

```
./ConfigEngine.sh wp-change-portal-admin-user
-DnewAdminId=uid=wpadmin,cn=users,dc=ibm,dc=com -DnewAdminPw=wpadmin
-DnewAdminGroupId=cn=wpsadmins,cn=groups,dc=ibm,dc=com
```

31. Restart the DMGR, nodeagent, and WebSphere_Portal server.

You have successfully federated and clustered a primary WebSphere Portal server to a Deployment Manager that already had LDAP security enabled.

This section replaces the following chapters:

Chapter 4 – Installing the DMGR. Here we assume the DMGR is already installed.

Chapter 5 – Configuring the DMGR. Some of the steps in the main guide assume default security. That does not apply here.

Chapter 6 – Federating and Clustering the Primary Node. That is the bulk of this section.

Chapter 7 – Configuring LDAP security. Security is already enabled in this section; no need to redo it.
Appendix B – Supplemental Information

B-1 – Script to create and setup DB2 databases

NOTE: The script provided is based on the DB2 commands found on this page in the Product Documentation:

http://www-10.lotus.com/ldd/portalwiki.nsf/dx/Linux_clustered_server_Creating_a_remote_or_local_DB2_database_manually_wp8

The following section contains the contents of the SQL script used to create the WebSphere Portal DB2 databases. To use this script, complete the following steps:

1. Copy the contents of this section into a text file
2. Edit the database names, user names and passwords in the file to match those of your intended environment. Do NOT change the JCR bufferpool or tablespace names. These must be the values listed here.
3. Save the file as a .sql file (for example CreateDatabases.sql)
4. Copy the file to a temporary directory on the DB2 server.
5. As the database administrator, execute the script:

   db2 -tvf <temporary location>/CreateDatabases.sql

This script does all of the following:

- Creates and updates six databases (you may change these names): reldb, comdb, cusdb, jcrdb, lmbdb, fdbkdb
- Creates bufferpools for jcrdb. Do NOT change these names: ICMLSFREQBP4, ICMLSVOLATILEBP4, ICMLSMAINBP32, CMBMAIN4.
- Creates tablespaces for jcrdb. Do NOT change these names: ICMLFQ32, ICMLNF32, ICMVFQ04, ICMSFQ04, CMBINV04, ICMLSSYSTSPACE32, ICMLSSYSTSPACE4, ICMLSUSRTSPACE4
CREATE DB reldb using codeset UTF-8 territory us PAGESIZE 8192;
UPDATE DB CFG FOR reldb USING applheapsz 4096;
UPDATE DB CFG FOR reldb USING app_ctl_heap_sz 1024;
UPDATE DB CFG FOR reldb USING stmtheap 32768;
UPDATE DB CFG FOR reldb USING dbheap 2400;
UPDATE DB CFG FOR reldb USING locklist 1000;
UPDATE DB CFG FOR reldb USING logfilsiz 4000;
UPDATE DB CFG FOR reldb USING logprimary 12;
UPDATE DB CFG FOR reldb USING logsecond 20;
UPDATE DB CFG FOR reldb USING logbufsz 32;
UPDATE DB CFG FOR reldb USING avg_appls 5;
UPDATE DB CFG FOR reldb USING locktimeout 30;
UPDATE DB CFG FOR reldb using AUTO_MAINT off;

CREATE DB comdb using codeset UTF-8 territory us PAGESIZE 8192;
UPDATE DB CFG FOR comdb USING applheapsz 4096;
UPDATE DB CFG FOR comdb USING app_ctl_heap_sz 1024;
UPDATE DB CFG FOR comdb USING stmtheap 32768;
UPDATE DB CFG FOR comdb USING dbheap 2400;
UPDATE DB CFG FOR comdb USING locklist 1000;
UPDATE DB CFG FOR comdb USING logfilsiz 4000;
UPDATE DB CFG FOR comdb USING logprimary 12;
UPDATE DB CFG FOR comdb USING logsecond 20;
UPDATE DB CFG FOR comdb USING logbufsz 32;
UPDATE DB CFG FOR comdb USING avg_appls 5;
UPDATE DB CFG FOR comdb USING locktimeout 30;
UPDATE DB CFG FOR comdb using AUTO_MAINT off;

CREATE DB cusdb using codeset UTF-8 territory us PAGESIZE 8192;
UPDATE DB CFG FOR cusdb USING applheapsz 4096;
UPDATE DB CFG FOR cusdb USING app_ctl_heap_sz 1024;
UPDATE DB CFG FOR cusdb USING stmtheap 32768;
UPDATE DB CFG FOR cusdb USING dbheap 2400;
UPDATE DB CFG FOR cusdb USING locklist 1000;
UPDATE DB CFG FOR cusdb USING logfilsiz 4000;
UPDATE DB CFG FOR cusdb USING logprimary 12;
UPDATE DB CFG FOR cusdb USING logsecond 20;
UPDATE DB CFG FOR cusdb USING logbufsz 32;
UPDATE DB CFG FOR cusdb USING avg_apps 5;
UPDATE DB CFG FOR cusdb USING locktimeout 30;
UPDATE DB CFG FOR cusdb using AUTO_MAINT off;

CREATE DB jcrdb using codeset UTF-8 territory us PAGESIZE 8192;
UPDATE DB CFG FOR jcrdb USING applheapsz 4096;
UPDATE DB CFG FOR jcrdb USING app_ctl_heap_sz 1024;
UPDATE DB CFG FOR jcrdb USING stmtheap 32768;
UPDATE DB CFG FOR jcrdb USING dbheap 2400;
UPDATE DB CFG FOR jcrdb USING locklist 1000;
UPDATE DB CFG FOR jcrdb USING logfilsiz 4000;
UPDATE DB CFG FOR jcrdb USING logprimary 12;
UPDATE DB CFG FOR jcrdb USING logsecond 20;
UPDATE DB CFG FOR jcrdb USING logbufsz 32;
UPDATE DB CFG FOR jcrdb USING avg_apps 5;
UPDATE DB CFG FOR jcrdb USING locktimeout 30;
UPDATE DB CFG FOR jcrdb using AUTO_MAINT off;

CREATE DB lmdb using codeset UTF-8 territory us PAGESIZE 8192;
UPDATE DB CFG FOR lmdb USING applheapsz 4096;
UPDATE DB CFG FOR lmdb USING app_ctl_heap_sz 1024;
UPDATE DB CFG FOR lmdb USING stmtheap 32768;
UPDATE DB CFG FOR lmdb USING dbheap 2400;
UPDATE DB CFG FOR lmdb USING locklist 1000;
UPDATE DB CFG FOR lmdb USING logfilsiz 4000;
UPDATE DB CFG FOR lmdb USING logprimary 12;
UPDATE DB CFG FOR lmdb USING logsecond 20;
UPDATE DB CFG FOR lmdb USING logbufsz 32;
UPDATE DB CFG FOR lmdb USING avg_apps 5;
UPDATE DB CFG FOR lmdb USING locktimeout 30;
UPDATE DB CFG FOR lmdb using AUTO_MAINT off;

CREATE DB fdbkdb using codeset UTF-8 territory us PAGESIZE 8192;
UPDATE DB CFG FOR fdbkdb USING applheapsz 4096;
UPDATE DB CFG FOR fdbkdb USING app_ctl_heap_sz 1024;
UPDATE DB CFG FOR fdbkdb USING stmtheap 32768;
UPDATE DB CFG FOR fdbkdb USING dbheap 2400;
UPDATE DB CFG FOR fdbkdb USING locklist 1000;
UPDATE DB CFG FOR fdbkdb USING logfilsiz 4000;
UPDATE DB CFG FOR fdbkdb USING logprimary 12;
UPDATE DB CFG FOR fdbkdb USING logsecond 20;
UPDATE DB CFG FOR fdbkdb USING logbufsz 32;
UPDATE DB CFG FOR fdbkdb USING avg_appls 5;
UPDATE DB CFG FOR fdbkdb USING locktimeout 30;
UPDATE DB CFG FOR fdbkdb using AUTO_MAINT off;

CONNECT TO jcrdb USER db2inst1 USING password;
CREATE BUFFERPOOL ICMLSFREQBP4 SIZE 1000 PAGESIZE 4 K;
CREATE BUFFERPOOL ICMLSVOATILEBP4 SIZE 16000 PAGESIZE 4 K;
CREATE BUFFERPOOL ICMLSMAINBP32 SIZE 16000 PAGESIZE 32 K;
CREATE BUFFERPOOL CMBMAIN4 SIZE 1000 PAGESIZE 4 K;
CREATE REGULAR TABLESPACE ICMLFQ32 PAGESIZE 32 K MANAGED BY SYSTEM USING
('ICMLFQ32') BUFFERPOOL ICMLSMAINBP32;
CREATE REGULAR TABLESPACE ICMLNF32 PAGESIZE 32 K MANAGED BY SYSTEM USING
('ICMLNF32') BUFFERPOOL ICMLSMAINBP32;
CREATE REGULAR TABLESPACE ICMVFQ04 PAGESIZE 4 K MANAGED BY SYSTEM USING
('ICMVFQ04') BUFFERPOOL ICMLSVOATILEBP4;
CREATE REGULAR TABLESPACE ICMSFQ04 PAGESIZE 4 K MANAGED BY SYSTEM USING
('ICMSFQ04') BUFFERPOOL ICMLSFREQBP4;
CREATE REGULAR TABLESPACE CMBINV04 PAGESIZE 4 K MANAGED BY SYSTEM USING
('CMBINV04') BUFFERPOOL CMBMAIN4;
CREATE SYSTEM TEMPORARY TABLESPACE ICMLSSYSTSPACE32 PAGESIZE 32 K MANAGED BY SYSTEM
USING ('icmlssystsapce32') BUFFERPOOL ICMLSMAINBP32;
CREATE SYSTEM TEMPORARY TABLESPACE ICMLSSYSTSPACE4 PAGESIZE 4 K MANAGED BY SYSTEM
USING ('icmlssystsapce4') BUFFERPOOL ICMLSVOATILEBP4;
CREATE USER TEMPORARY TABLESPACE ICMLSUSRTSPACE4 PAGESIZE 4 K MANAGED BY SYSTEM
USING ('icmlsusrtspac4') BUFFERPOOL ICMLSVOATILEBP4;
UPDATE DB CFG FOR jcrdb USING DFT_QUERYOPT 2;
UPDATE DB CFG FOR jcrdb USING PCKCACHESZ 16384;

DISCONNECT jcrdb;
TERMINATE;

ENDED COPY HERE---DO NOT INCLUDE THIS LINE------------
After creating your cluster, you may need to add additional members to the cluster. This section will describe how to properly add a vertical cluster member to your cluster.

A 'vertical cluster member' is an additional WebSphere_Portal JVM on an existing federated Portal node. You are not required to install an additional WebSphere Portal installation or profile on any server to create a vertical cluster member.

1. From a command window, navigate to <AppServer root>/profiles/Dmgr01/bin
2. Execute the following command:

   ./startManager.sh

3. Once the DMGR is open for e-business, launch a web browser and access the DMGR Administrative Console:


4. Navigate to Servers -> Clusters -> WebSphere Application Server clusters -> PortalCluster -> Cluster Members

5. Click 'New'
6. On the next screen, enter the following information:

   Member Name - The new member name (for example WebSphere_Portal_3)

   NOTE: Do not use any name that contains a space

   Select Node – Select a node that is part of your cluster

   Generate Unique HTTP Ports – Ensure this is checked

7. Click “Add Member” to add the new member to the table on this screen.

8. Click “Next”

9. Review the summary screen and click Finish.
10. Save the changes

11. Navigate to Server Types → WebSphere Application Servers → WebSphere_Portal_3 → Ports and note the following two port values:

   WC_defaulthost
   WC_defaulthost_secure

<table>
<thead>
<tr>
<th>WC_defaulthost</th>
<th>*</th>
<th>10050</th>
<th>View associated transports</th>
</tr>
</thead>
<tbody>
<tr>
<td>WC_defaulthost_secure</td>
<td>*</td>
<td>10053</td>
<td>View associated transports</td>
</tr>
</tbody>
</table>

12. Update the Virtual Hosts to include these two ports if they are not already present:
   a) Navigate to Environment → Virtual Hosts → default_host → Host Aliases
   b) Click “New”
   c) Set Hostname to *
   d) Set Port to the value of WC_defaulthost (in this example, 10050)
   e) Click “OK”
   f) Repeat a-e for WC_defaulthost_secure (in this example, 10053)
   g) Save changes

13. Enable Dynamic Replication on the new cluster member.
   a) Navigate to Servers → Server Types → WebSphere Application Servers → WebSphere_Portal_3 → Container Services → Dynamic Cache Service
   b) Set Cache Size to 3000 entries
   c) Check the Enable Cache Replication Box
   d) Select “Not Shared” from the Replication Type drop-down menu
14. Navigate to Servers → Server Types → WebSphere application servers → WebSphere_Portal → Session Management → Distributed Environment Settings

15. Click the blue link for 'Memory-to-memory replication':

e) Click “OK” and save changes.
16. In the Replication Domain drop-down menu, select the one for your cluster (e.g. PortalCluster).

17. In the Replication Mode drop-down menu, select 'Both client and server'.

18. Click OK and Save all changes.

19. From the Portal node that you created the vertical cluster member on, open a terminal window and change directories to the `<wp_profile root>/ConfigEngine` directory.

20. Execute the following ConfigEngine script to remove server-scoped entries from the new cluster member:

   ```bash
   ./ConfigEngine.sh cluster-node-config-vertical-cluster-setup
   -DServerName=WebSphere_Portal_3 -DWasPassword=password
   
   where ServerName is set to your new vertical cluster member name. In this case, WebSphere_Portal_3 is my new vertical cluster member.

   **NOTE:** The vertical cluster member may not function correctly until this step has been completed successfully.

21. Synchronize the nodes and restart the DMGR, nodeagents and cluster members.

22. Verify you can access your new cluster member in a URL using the port defined for WC_defaulthost in step 11 (assuming the vertical cluster membe was created on the primary portal node):

B-3 – Using the Configuration Wizard

There is a brand new configuration wizard in WebSphere Portal v8. The configuration wizard is used to run ConfigEngine scripts using a GUI interface instead of from the command line with the 'ConfigEngine.sh/bat' script.

In previous releases, the Configuration Wizard was a java application that you launched on your Portal server. In v8, it is a servlet that is deployed to its own profile (cw_profile) and is accessed from the web. It also includes a feature called 'workflows' that allow you to generate custom scripts to accomplish a string of tasks. This is particularly useful if you have to execute the same steps on multiple environments.

I did not use the Configuration Wizard in any of the ConfigEngine steps in this guide, however you may decide that you would rather use the wizard instead of updating property running commands.

In this section, I will briefly explain how to run ConfigEngine commands from the v8 Configuration Wizard and how to generate a basic workflow for cluster setup.

Tips for Configuration Wizard

- Default location of cw_profile is <AppServer root>/profiles/cw_profile.
- Uses server1 in this profile and the default port is 9060.
- File User Repository is set up by default and the same ID you used during Portal installation is used here.
- Even though initially, the Portal profile and Configuration Wizard profile use the same user IDs, the cw_profile user repository is completely separate from the Portal user repository. If you configure Portal to use an LDAP, it does NOT configure cw_profile to use the same LDAP.
- Includes log viewer for ConfigTrace.log that is designed so you can easily see what tasks have been executed, what tasks were successful/failed, and results of each.
- For more details, please visit the WebSphere Portal product documentation:
  http://www-10.lotus.com/ldd/portalwiki.nsf/dx/Configuration_Wizard_wp8
Running ConfigEngine scripts using the Configuration Wizard

1. From the cw_profile/bin directory, launch server1 using the startServer command:

   ./startServer.sh server1

2. Log into the WAS Admin Console for this server:


3. On the left hand side, click the link for Configuration Wizard:
4. When the Wizard loads, click the “Run Tasks” link on the right-hand side of the screen:

5. On the next screen, enter the name of the ConfigEngine task you'd like to execute (e.g. database-transfer) in the first field.

6. In the second step, change whatever properties need to be updated:
   a) Click the 'Change Properties' button
   b) Click the 'Add Property' button
   c) Select the properties you want to add. You can select multiple properties with Ctrl + Left-click.

   **NOTE:** This is only necessary if you need to change any values from what is currently in the wkplc*.properties files.

   **NOTE:** The properties are grouped to match the wkplc*.properties files and are pulled directly from those files.

   If you have a large number of properties to update, you may find it easier to edit them first in the relevant wkplc*.properties file, then use the Configuration Wizard.

   d) The properties you selected will appear in a table. Edit the value column of that table to update the value:
e) Click Save when you've finished adding properties

**NOTE:** Any properties you add here are automatically added to the wkplc*.properties files if the task you execute is **successful**.

7. For the third step, add any additional parameters you might want to include in the ConfigEngine command. For example, `-DWasPassword=<your password>` or `-verbose`.

8. When ready, click 'Run Task Now' to execute the ConfigEngine script:

![Configuration Wizard](image)

**NOTE:** If you click the 'Create Scripts' button, a downloadable zip file will be created. The zip file contains:

- Instructions
- The properties you set in Step 2 as a helper file
- An executable shell/batch script that calls your ConfigEngine command

This allows you to run the shell/batch script on the PortalServer to run the same command you entered in Step 1, plus the properties you set in Step 2.
Creating Workflows
In the Configuration Wizard, workflows are a series of scripts (like ConfigEngine, addNode, stopServer, etc) simplified into simple shell/batch files. This are very useful if you have to execute the same ConfigEngine scripts on multiple Portal servers.

A few sample workflows are included with the Configuration Wizard. This section will show you how to customize and generate scripts for creating a cluster.

When complete, you will have a workflow that adds the primary node to a DMGR, runs cluster-node-config-post-federation and runs cluster-node-config-cluster-setup.

1. From the cw_profile/bin directory, launch server1 using the startServer command:

   ./startServer.sh server1

2. Log into the WAS Admin Console for this server:

   http://<hostname>:9060/ibm/console

3. On the left hand side, click the link for Configuration Wizard:
4. On the right-hand side, click the 'View Workflows' button:

![View Workflows button](image)

5. Click the checkbox next to 'Create a Static Cluster', then click the Customize Workflow button:

![Customize Workflow button](image)

6. At Step 1, select the relevant operating system for the server where you'll be running the workflow. The OS of your Portal server is selected by default. Click Next.

7. Adjust the profile name, profile path, and temporary directory as needed. Click Next.

   **NOTE:** There will be a fourth option here that varies based on OS. For Linux, you will see an option to select the shell type. For Windows, you'll see an option to turn echo on/off.
8. On the next screen, the properties required for the cluster setup workflow are displayed. They are pulled directly from your wkplc*.properties files.

Adjust the values of the listed properties as needed and click Next:

![Configuration Wizard](image)

9. If validation is successful, click Next.

**NOTE:** You can click Save here if you'd like to save your settings. This can be used if you need to redo this workflow later. You can save time by importing your saved settings at Step 1.

10. On Step 4, click the 'Create Files' button.

11. Once the scripts are created, click the 'Download' link to download them. You will download a zip file.
12. Copy the zip file to your Portal server and extract it to a temporary directory. Inside of it you will find the following:

   a) **properties** directory. This contains the helper *.properties files that correspond to any ConfigEngine scripts included in the workflow. In this example, we did the Cluster Setup workflow so there should be two properties files: post-federation.properties and cluster-setup.properties.

   b) **scripts** directory. Contains the shell scripts generated by the workflow. For the Cluster Setup workflow, there should be three scripts: federation.sh, post-federation.sh, cluster-setup.sh.

   c) **CreateStaticCluster.html**. Contains the instructions to run the workflow. The name of this file is the name of the workflow, so it will vary depending on which workflow you customized.

   d) **CreateStaticCluster.wfi**. The workflow definition itself.

13. Follow the *.html file instructions to run the workflow. For the Create Static Cluster workflow, it is three steps:

   NOTE: Ensure the user has execute permissions on these files.

   a) Execute ./federation.sh. This script runs addNode with all the required parameters for you.

   b) Execute ./post-federation.sh. This script runs the ConfigEngine script 'cluster-node-config-post-federation' for you. It uses the helper file + the values already in wkplc*.properties.

   c) Execute ./cluster-setup.sh. This script runs the ConfigEngine script 'cluster-node-config-cluster-setup' for you. It uses the helper file + the values already in wkplc*.properties.

When the workflow was initially created, you provide the profile name and profile path. These values are hardcoded in the scripts themselves. If you need to execute these workflows on multiple Portal servers, ensure you do one of the following:

- Edit the generated shell scripts to use a valid profile on that environment (in case the profile name is different, for example)

or

- Launch the Configuration Wizard again and adjust the profile name and profile template values to match the next Portal environment, then regenerate the scripts.
Using the ConfigTrace Log Viewer

The configuration wizard also contains a handy tool to view the ConfigTrace.log file. This can be very useful if you need to quickly determine why a particular ConfigEngine script may have failed.

1. From the cw_profile/bin directory, launch server1 using the startServer command:
   ```
   ./startServer.sh server1
   ```

2. Log into the WAS Admin Console for this server:
   ```
   http://<hostname>:9060/ibm/console
   ```

3. On the left hand side, click the link for Configuration Wizard:
4. On the right-hand side, click the 'View Logs' link:

Log Viewer

5. The ConfigTrace.log is loaded into the tool and is displayed like this:

The log output is organized in the following way:

**Left Column** - Contains a list of tasks executed. A green check means the task was successful. A red X means the task failed. Click Prev/Next at the bottom to navigate the full set of tasks. Click the + next to any task name to see the sub-tasks executed.

**Center Column** – Contains the output of any task selected from the Left Column. Click Prev/Next at the top of this column to step through the task output.

**Right Column** – Contains the list of properties used for any task selected from the Left Column. Property list can be filtered by typing in the property you are looking for.
**B-4 – How to properly extract the WebSphere Portal Installation media**

This section will detail how to properly extract the WebSphere Portal installation media when it has been downloaded from Passport Advantage. This has caused some confusion in the past and if not done correctly, the installation may fail or not run at all.

The main point to take away from this section is that all downloaded zip/tar files MUST be extracted into the exact same parent directory. If you do that, then you will be fine.

1. First, refer to this link in the product documentation to understand what images you will need to download from Passport Advantage:

   http://www-10.lotus.com/ldd/portalwiki.nsf/dx/Electronic_images_wp8

There are 6 editions of the WebSphere Portal Media: Server, Enable, Extend, Express, Web Content Management, and Web Content Management Standard.

Each edition has a set of **Required Software** and a set of **Optional Software**.

These instructions are purely for the **Required Software**.

2. In this guide, I used WebSphere Portal Enable, so looking at this link:

   http://www-10.lotus.com/ldd/portalwiki.nsf/dx/Getting_WebSphere_Portal_Enable_software_wp8

   There are 10 images I need to download (listed in Tables 2, 3, and 4).

   **NOTE:** Table 1 includes a quick start guide. This is just documentation and is not actually used by the installer. It can be skipped if you'd like.

3. On the server that will hold the Portal Installation media, create a directory anywhere you'd like. For example:

   /opt/Portal8Media

4. Extract each zip file you downloaded into the directory you created.

   **NOTE:** While extracting the images, you may be prompted to overwrite existing directories. Select OK or YES to overwrite any existing directories. This is normal and ok.
5. When this is complete, you should end up with the following directory structure (using my /opt/Portal8Media as an example):

```
/opt/Portal8Media/Setup
/opt/Portal8Media/WAS
/opt/Portal8Media/Portal
/opt/Portal8Media/Enable
```

**NOTE:** If you are installing Portal Server, you will only have three directories: Setup, WAS, Portal. For every other edition, you will have four directories: Setup, WAS, Portal, and your Offering (Enable for example).
About The Author
Hunter Tweed works with the IBM WebSphere Portal Level 2 Support organization and is Team Lead for the Installation and Configuration L2 team. He has authored many other Step-By-Step guides for various Portal deployment scenarios.

If you have any questions about the content of this guide, Hunter can be reached at: httweed@us.ibm.com.

If you encounter any failures following the steps in this guide, you may open a PMR with WebSphere Portal Level 2 support.

Acknowledgments
The WebSphere Portal Information Development team for providing the Product Documentation that this guide is based on

Travis Cornwell, WebSphere Portal L2 Support Engineer for Portal Security, for additional feedback, recommendations and comments for the content in this guide.
## Change History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>What Changed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>05/17/12</td>
<td>Published</td>
</tr>
<tr>
<td>2</td>
<td>06/15/12</td>
<td>Updated Portal repository info to point to Setup/eimage/repository.config</td>
</tr>
<tr>
<td>3</td>
<td>10/09/12</td>
<td>Removed 'You' from Title page</td>
</tr>
</tbody>
</table>