



Fuego Enterprise 5.5 for BEA Web Logic Installation Instructions

Installation and Configuration Guide

Fuego v5.5

Last Revisited: March 10th, 2006.

	2005 Fuego Inc. - Company Proprietary and Confidential.	
	All Rights Reserved.	Page 1 of 92

Table of Contents

Fuego Enterprise 5.5 for BEA Web Logic Installation Instructions.....	3
Introduction.....	3
Minimal Requirements.....	3
Installing Fuego 5.5 Enterprise for Web Logic	3
Configuring the Fuego Directory Service.....	3
Creating Fuego Directory Service	3
Configuring Fuego Web Application Server for Web Console deployment.....	11
Starting Fuego Web Console Windows Service.....	13
Configuring a Fuego Server on Web Logic Application Server.....	16
Creating the Fuego Server backend RDBMS	23
Configuring Fuego Server Resources on Web Logic Application Server.....	24
Creating Fuego Directory Service Connection Pool and Data Source	25
Creating Fuego Server backend Connection Pool and Data Source.....	35
Creating Topics and Queues for Fuego Server for Automatic Activity execution and News	43
Topics and Queues for Fuego server (Web Logic infrastructure)	43
Creating a Connection Factory for ToDoItems Queue (Web Logic infrastructure) ..	43
Creating a Connection Factory for Fuego Server News Topic (Web Logic infrastructure).....	45
Creating JMS server (if necessary)	47
Creating Fuego ToDoItems Queue (Web Logic infrastructure)	48
Creating Fuego Server News Topic (Web Logic infrastructure).....	49
Creating a Connection Factory for ToDoItems Queue (TIBCO EMS 4.1)	51
Creating a Connection Factory for Fuego Server News Topic.....	53
Creating Fuego ToDoItems Queue (TIBCO EMS 4.1)	54
Creating Fuego Server News Topic (TIBCO EMS 4.1)	57
Deploying Fuego J2EE Deployer, Fuego Server and Fuego Work Portal	59
Deploying Fuego Library JAR.....	59
Deploying Fuego J2EE Deployer	60
Deploying Fuego Server	62
Deploying Fuego Portal	72
Publishing and deploying a project into a Fuego Engine.....	79
Deploying the Fuego Project EAR in WebLogic.....	84
Connecting to the Work Portal	90
Troubleshooting	91

Fuego Enterprise 5.5 for BEA Web Logic Installation Instructions

Introduction

The following document will describe all the steps to install and configure Fuego Enterprise 5.5 for BEA Web Logic. It will as well explain how to deploy a Fuego Project implemented in Fuego Studio on the Fuego Server deployed on BEA WebLogic Application Server.

Minimal Requirements

Fuego 5.5 Enterprise for BEA Web Logic runs on Web Logic 8.1 Service Pack 5 or greater. This Service Packs is a minimal requirement for Fuego Server to install and run successfully. This Service Pack contains corrections and patches needed for optimal behavior.

Installing Fuego 5.5 Enterprise for Web Logic

Configuring the Fuego Directory Service

After Fuego 5.5 Enterprise for BEA Web Logic has been successfully installed on your system, the Fuego Administrator will need to proceed with the configuration and setup of the environment to run a Fuego Server on Web Logic Application Server to execute Business Processes modeled on Fuego Studio.

Fuego Server for BEA Web Logic relies on a Directory Service to enforce Business Process Security as well as using it as a centralized repository for Fuego Metadata.

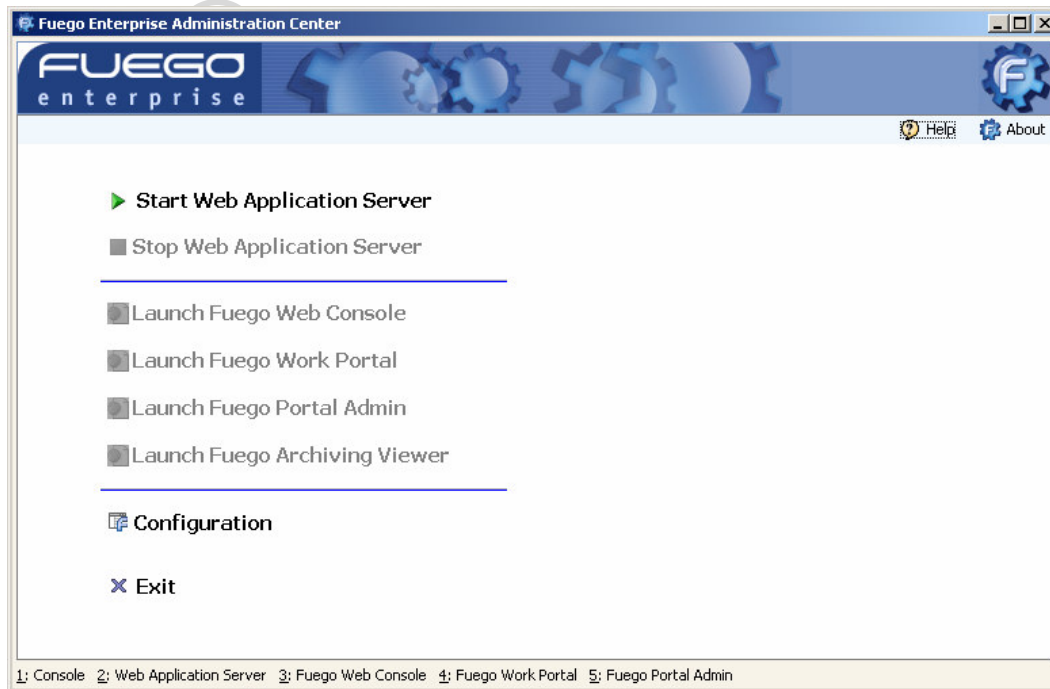
Before proceeding with anything else, the Fuego Administrator will need to create the Directory Service. In our example, the Fuego Administrator will deploy Fuego Directory Service on an Oracle RDBMS. Analogous procedure should be followed when deploying Fuego Directory Service on another RDBMS or LDAP compliant Directory Service.

Creating Fuego Directory Service

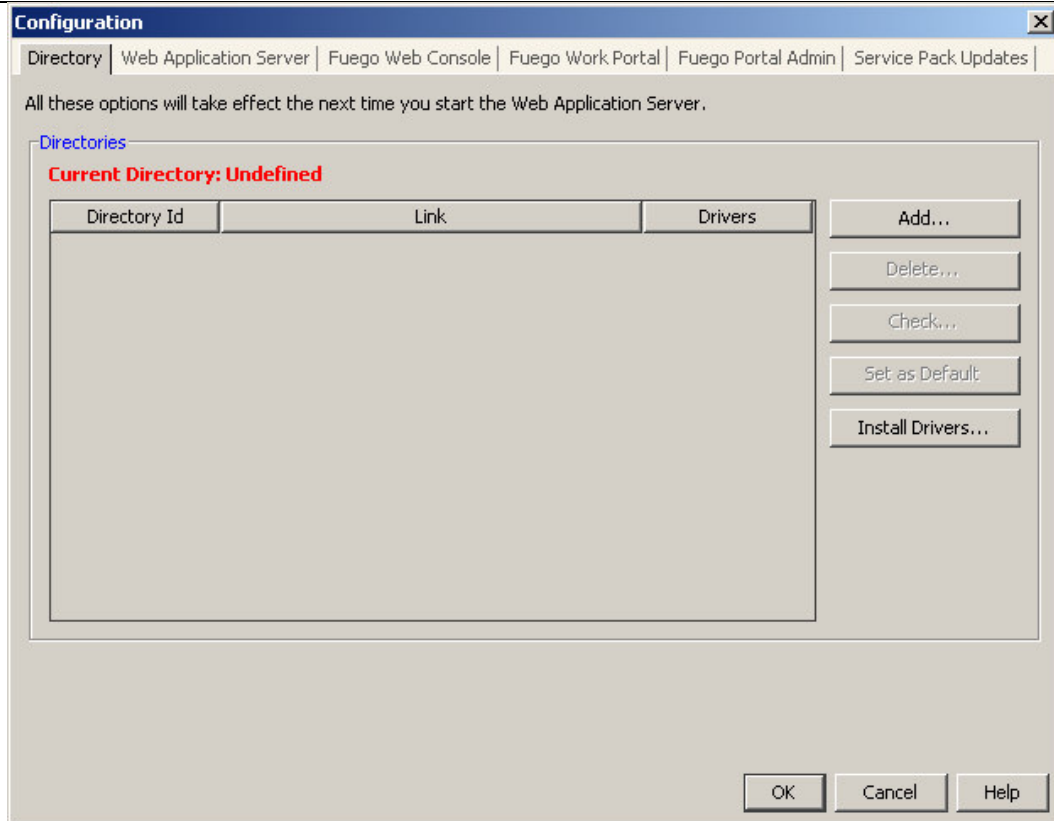
	2005 Fuego Inc. - Company Proprietary and Confidential.	
	All Rights Reserved.	Page 3 of 92

Fuego Enterprise 5.5 for Web Logic provides a desktop application for creating Fuego Directory Service. This application is Fuego Administration Center. The application may be launched using the shortcuts created at installation time. On Windows, the default path to the Admin Center shortcut is: Start -> Programs -> FuegoBPM 5.5 for WL -> Fuego Adm Center.

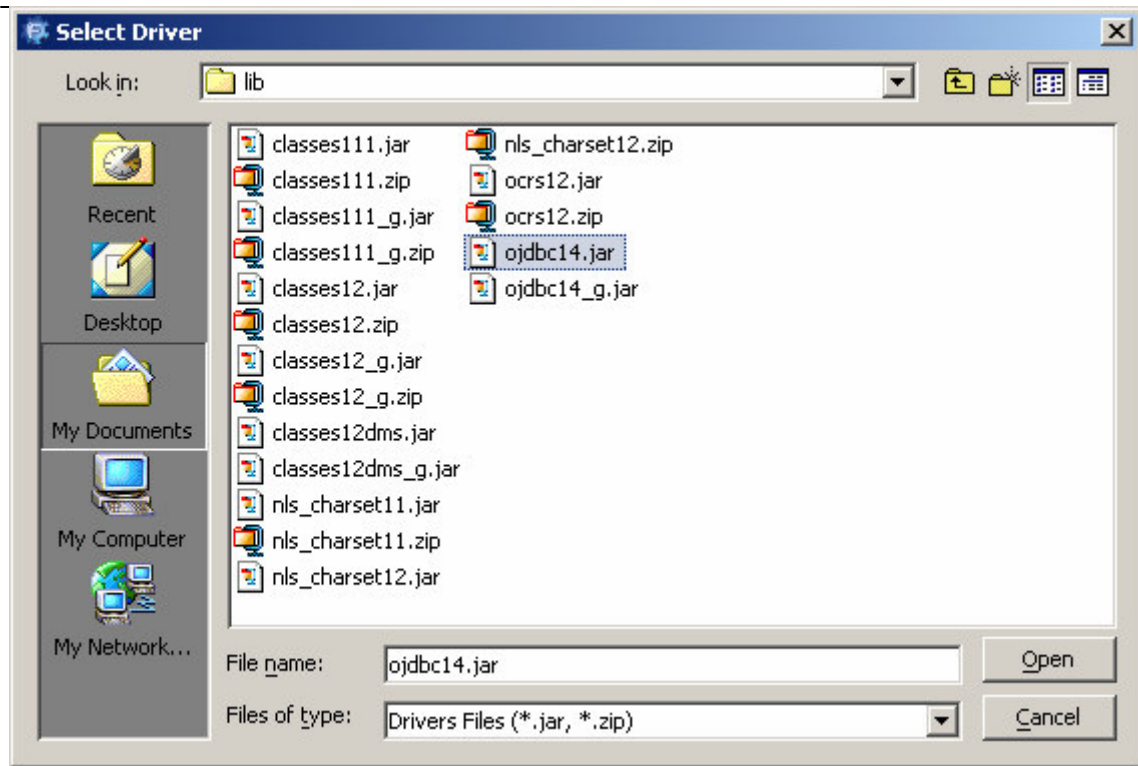
The following panel will be presented when Fuego Adm Center is launched.



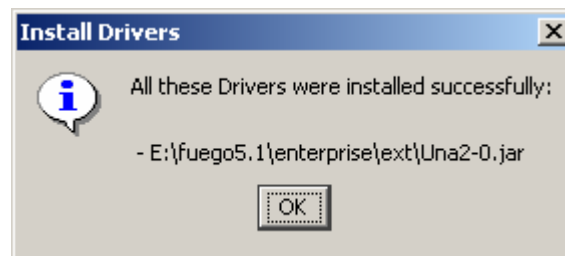
To proceed with the Fuego Directory Service configuration, click on the “Configuration” link at the bottom of the Panel. The following Panel is displayed.



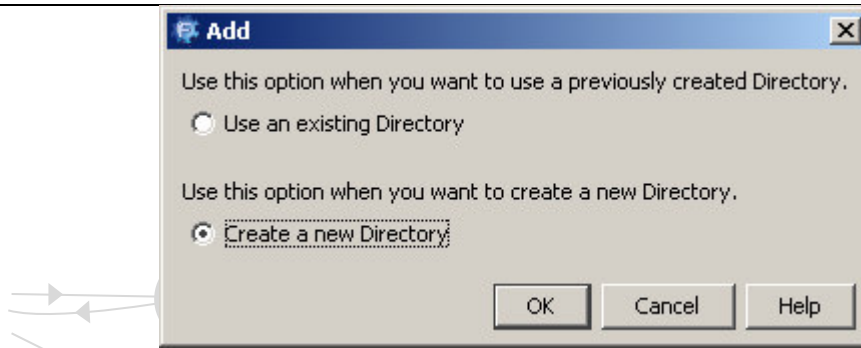
As the Directory Service will be deployed on Oracle, the appropriate JDBC Driver needs to be installed. Click on the “Install Drivers...” button to proceed with the installation of Oracle JDBC Driver. You will need to navigate through the Computer Hard disk looking for the file containing the JDBC Driver classes as depicted in the figure below.



Click on the “Open” button to proceed. After a successful JDBC Driver Installation the following dialog will be presented to the Administrator.



Click “Ok” to continue. To start creating the Directory Service, click on the “Add” button on the “Directory” Tab. The following dialog will be presented to decide whether a new Directory Service should be created or not. As we will be creating a new Directory Service on Oracle Server, the Fuego Administrator will need to select the “Create a New Directory” option.



Click “Ok” to proceed. A new panel will be presented requiring specific values to be provided to configure the Directory Service to be used by the Fuego Enterprise 5.5 Server deployed on BEA Web Logic. The following figure illustrates the panel. A brief explanation is provided for each form field. For more information check the Fuego FDI for Oracle Server Documentation.



The following is a brief description of each one of the fields in the form:

Directory Id: This is a logical name for the Directory Service. It can be any name.

Provider: As Oracle is the target for the Directory Service, the entry “Oracle JDBC” should be selected from the drop down.

Show the SQL sentences: It should be checked if the Administrator wants to get the SQL Script executed to create the Directory Service Database structure. In our case, we will let Fuego Administration Center to create it so it should be unchecked.

Database host: This is the host where the Oracle RDBMS is deployed.

Database port: Oracle JDBC Driver connects with Oracle RDBMS using TCP/IP protocol. The TCP/IP used by the Oracle RDBMS to receive incoming calls should be provided in this field. 1521 is the default TCP/IP for Oracle RDBMS.

Organization logical name: This is the logical name of the Organization being created with this Directory Service. This can be any literal. It is usually the name of the company that has bought Fuego.

Administrator user: This is Oracle Administrator user. This is a user with enough permission to create a Database, a Login and tables on the created Directory Service Database.

Administrator password: This is the password for the Administrator user specified in the previous field.

Schema Name: This is the Oracle user that will host the Directory Service tables.

Schema Password: This is the password for the Schema Name specified in the previous field.

Confirm Password: This is the same value provided in the “Schema Password” field. It is just for consistency when providing passwords as its value is not visualized to the Administrator entering the password.

Now, we will turn to the “Advanced” Tab. The values in this Tab will help us define the data for the Fuego Directory Service Administrator. Below, a list of the fields with a brief description is provided.

Administrator ID: This is the Fuego Directory Service Administrator ID. This is the user id to be used when connecting to Fuego Web Console to administer and manage Fuego Servers as well as publishing and deploying projects.

Administrator password: This is the password to be linked to the Fuego Directory Service Administrator ID.

Confirm password: This is the same value provided in the “Administrator password” field.

Fuego handles authentication: This is the preferred authentication mechanism since all Fuego Business Process participants will be persisted in the Directory Service Database. If this checkbox is not selected, the authentication will rely on Oracle RDBMS authentication.

The panel below shows the “Advanced” panel.

	2005 Fuego Inc. - Company Proprietary and Confidential.	
	All Rights Reserved.	Page 8 of 92

Create a new Directory

FUEGO
DIRECTORY SERVICE

Directory Id:

Provider:

Basic | **Advanced**

Administrator ID:

Administrator Password:

Confirm password:

Fuego handles authentication: ☒

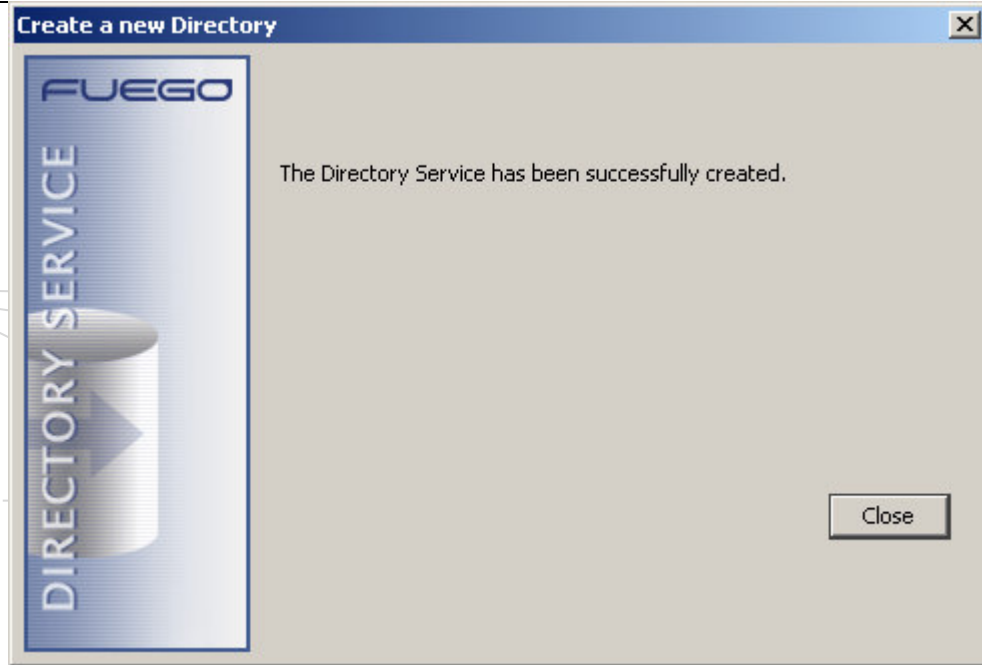
Tablespace:

Temporary Tablespace:

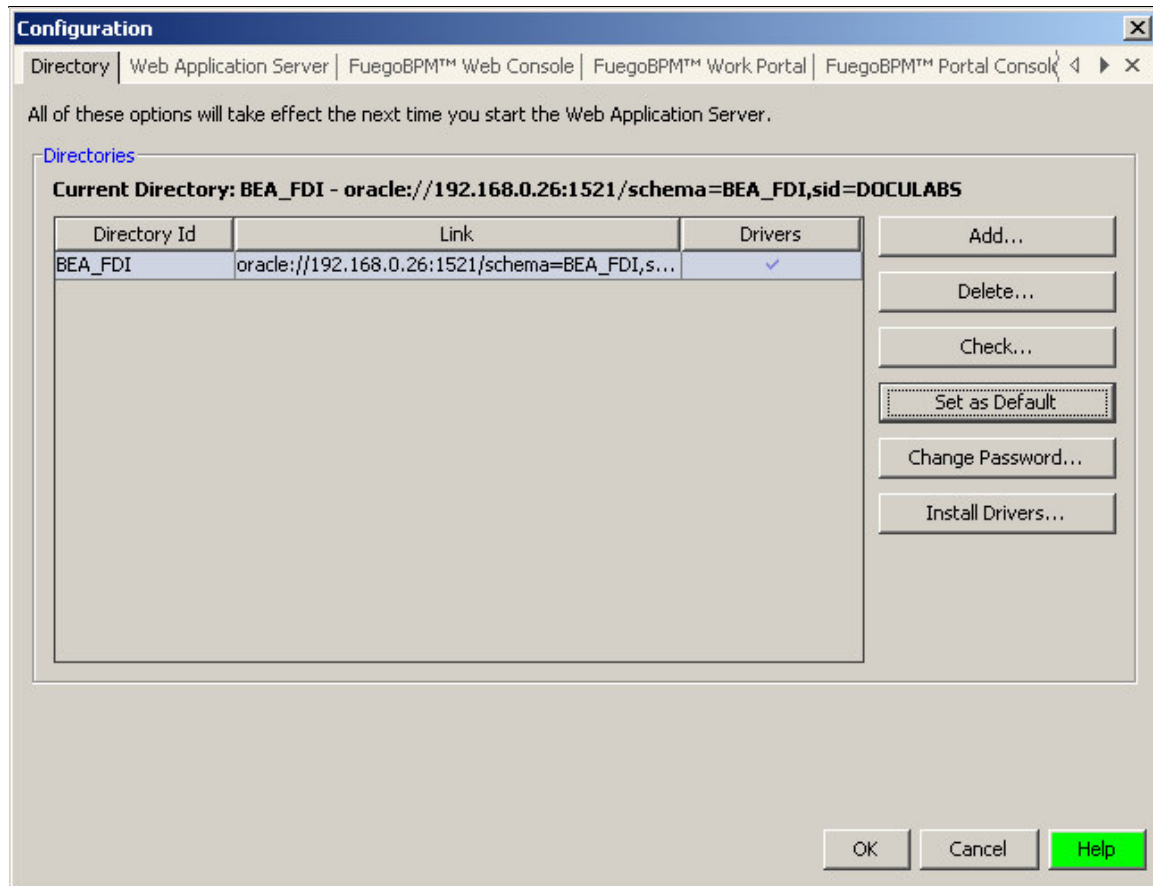
Profile:

Start **Cancel**

Click on the “Start” button to start the Fuego Directory Service creation. After a successful Directory Service creation, the following dialog will be presented.



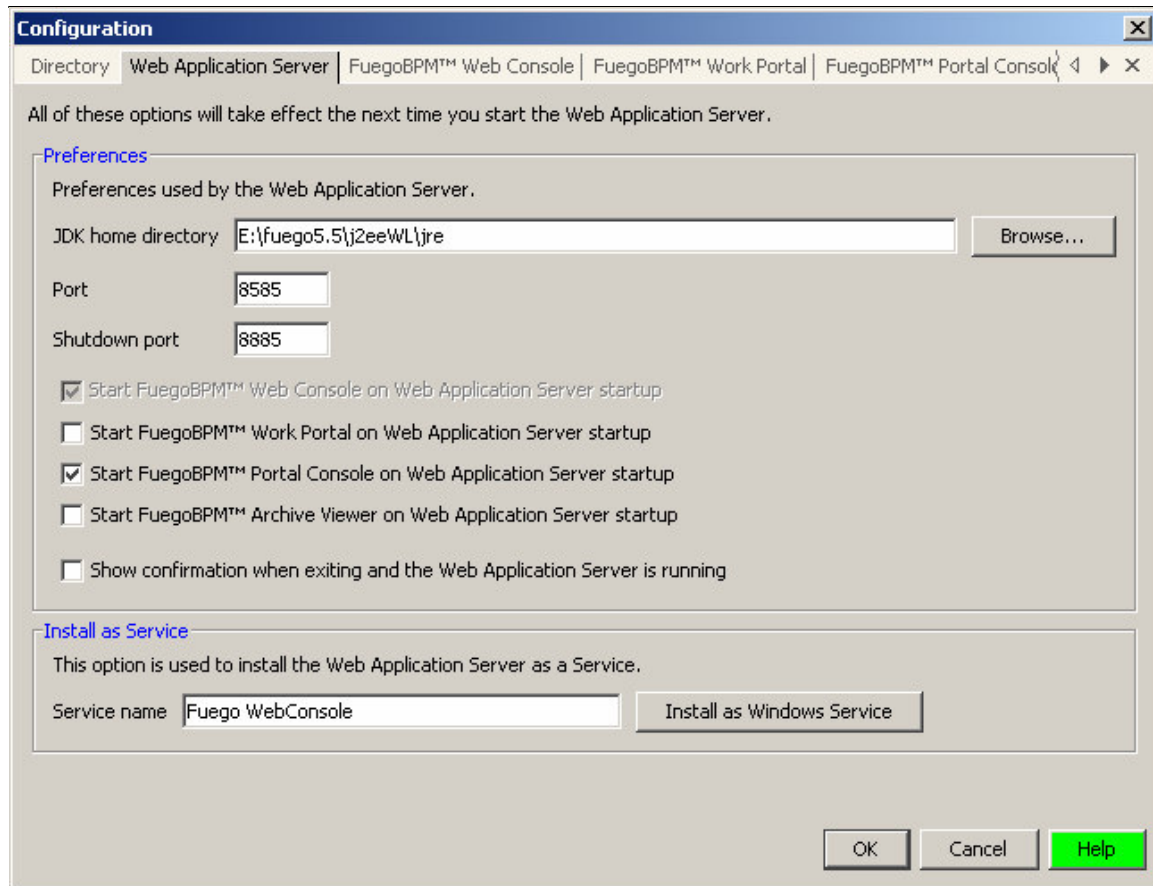
Click on the “Close” button to finish the Directory Service creation. The new created Directory Service will be displayed as a new entry on the “Directory” Tab as follows. It is important to note that this one is as well the default Directory Service.



Configuring Fuego Web Application Server for Web Console deployment

After successfully creating the Fuego Directory Service, it is convenient the Admin Service is installed as Windows Service. The Admin Service starts a Tomcat embedded Web Server with Fuego Web Console deployed on it. Installing it as a Windows Service enforces that when the machine is started, Fuego Web Console is automatically initialized.

To install the Web Console as a Windows Service, the Fuego Administrator should select the “Web Application Server” Tab on the Configuration Panel as shown below.



This panel defines the Web Console configuration parameters. The following list provides a brief description of the form fields.

JDK home directory: This field contains the absolute path to a 1.4.2 JVM. By default and on Windows, it is pointing to the one provided and installed by Fuego

Port: This is the port used by the Web Console embedded Tomcat. 9595 is the default port used by Fuego.

Shutdown port: This is the port used internally to stop the Web Console embedded Tomcat. 8585 is the default port used by Fuego.

Start Fuego Work Portal on Web Application Server startup: This checkbox should be unselected since the Fuego Portal will be deployed on Web Logic Application Server later with the Fuego Server deployment.

Start Fuego Portal Admin on Web application Server startup: This checkbox should be unselected since the Fuego Portal Admin will be deployed on Web Logic Application Server later with the Fuego Server deployment.

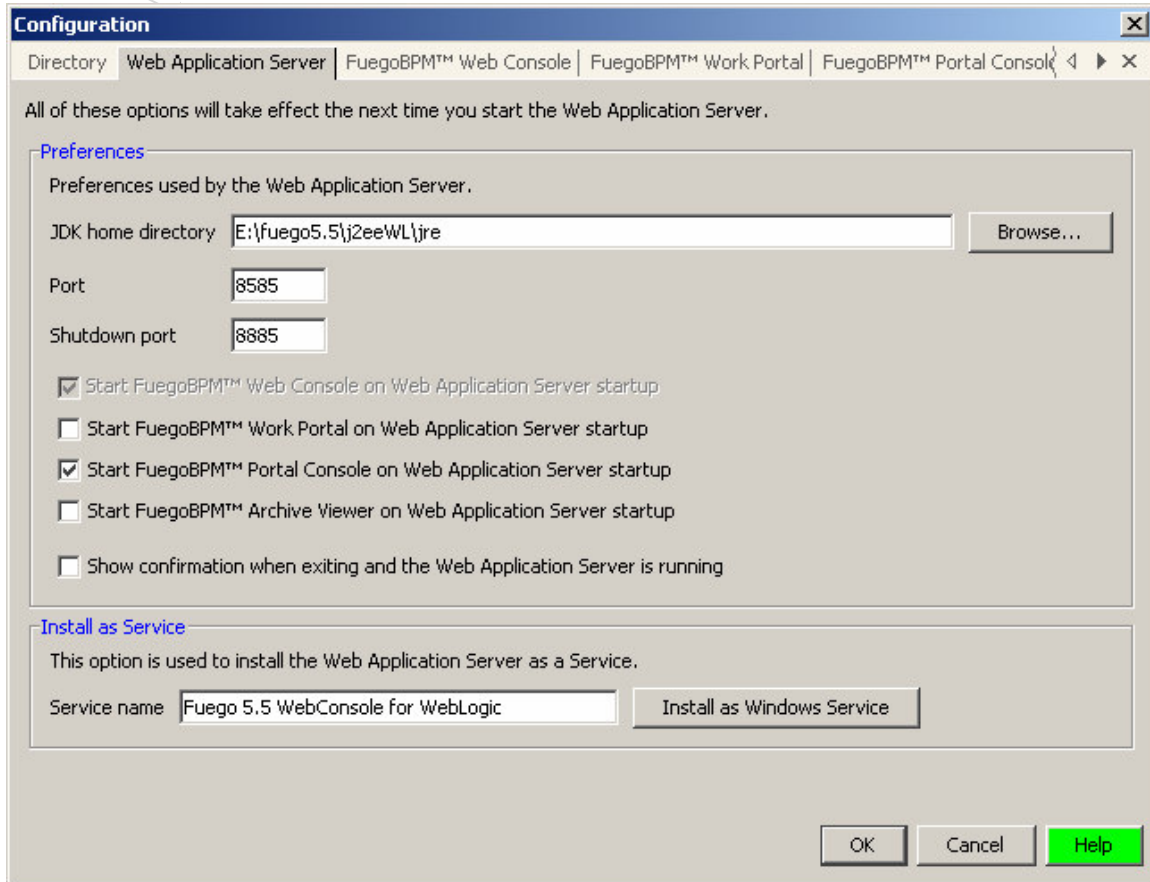
Start Fuego Archiving Viewer on Web Application Server startup: This checkbox should be unselected since the Fuego Portal will be deployed on Web Logic Application Server later with the Fuego Server deployment.

Show confirmation when exiting and the Web Application Server is running: This checkbox should be unselected so that a warning is not displayed when exiting Fuego Admin Center and the Web Application Server is running.

	2005 Fuego Inc. - Company Proprietary and Confidential.	
	All Rights Reserved.	Page 12 of 92

Service Name: This is the name of the Windows Service to be granted to the Fuego Web Console Service. It is a good idea to provide a name different than the default one. As Fuego Enterprise 5.5 for Web Logic has been installed, “Fuego 5.5 WebConsole for WebLogic” is an appropriate name.

Once the configuration for the Web Application Server has been defined, the Fuego Administrator should install the Web Console Windows Service by clicking on the “Install as Windows Service” button. After a successful installation, the “Service Name” field should be grayed out as depicted in the figure below.



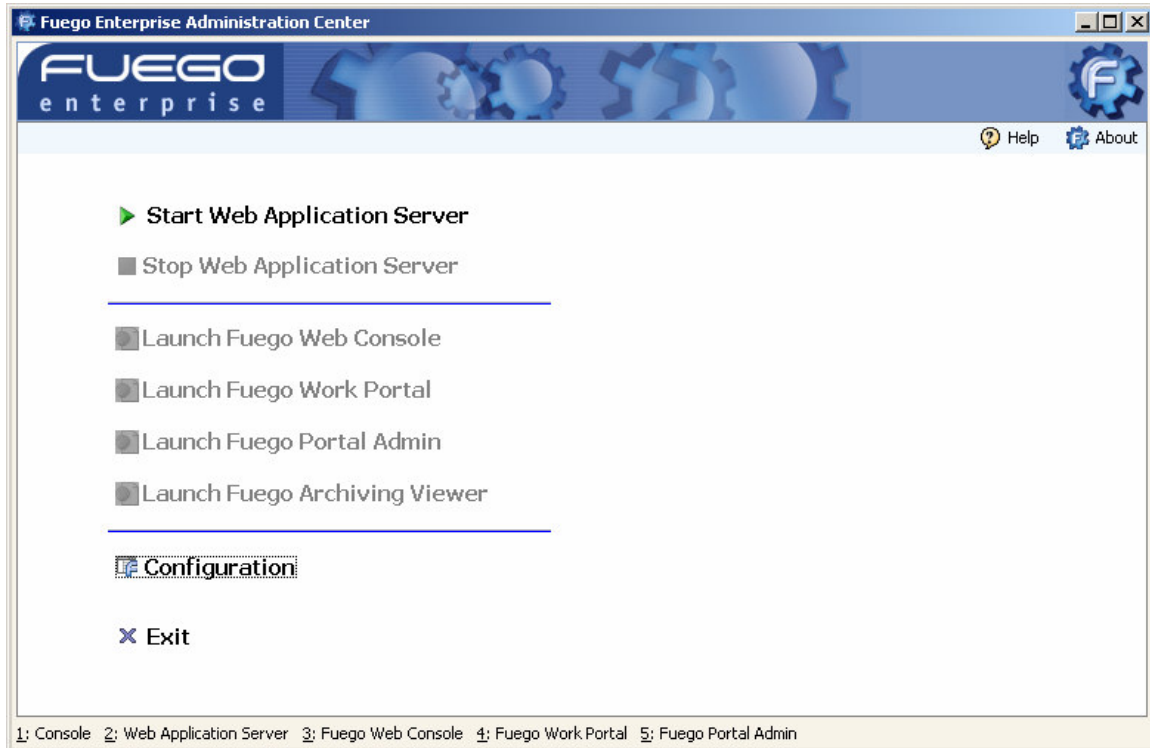
Unless specific configuration for logging is required, it is not necessary to go through the other Tabs in the Configuration Panel. Click “Ok” on the “Web Application Server” Tab Panel to finish the Web Console Configuration.

Starting Fuego Web Console Windows Service

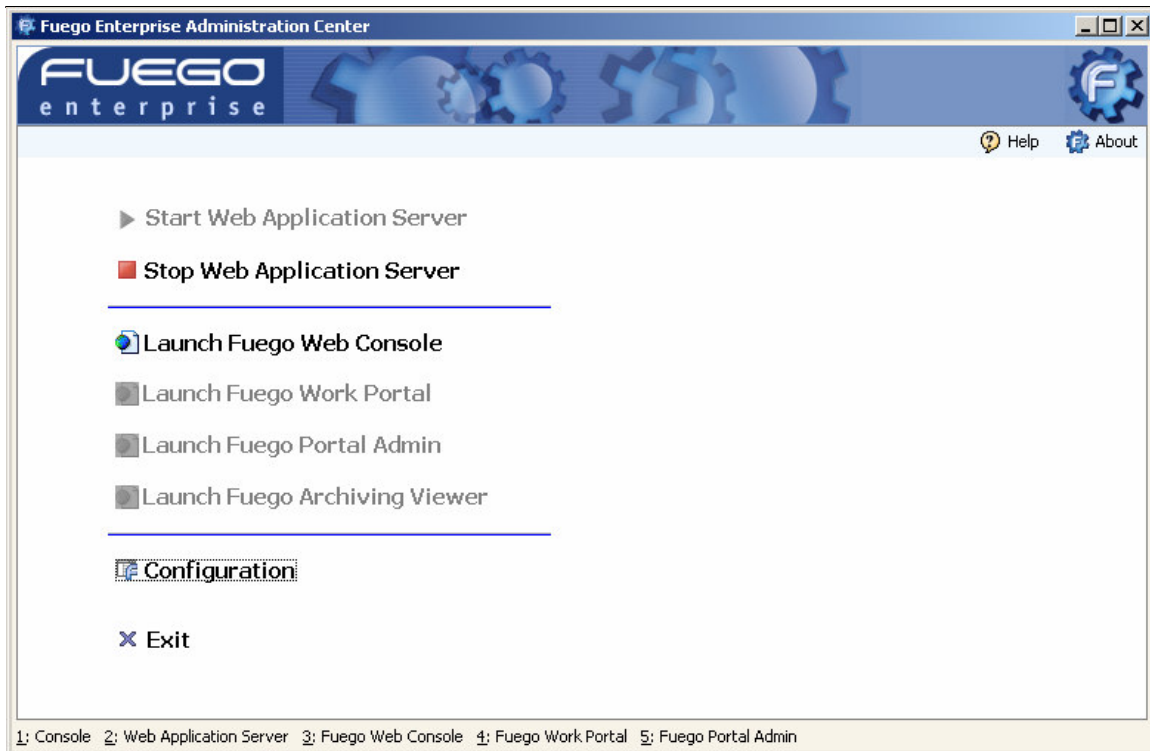
Once the Web Console has been configured, the Fuego Administrator should start the Service. This can be done through Fuego Adm Center application or simply by starting the previously created Windows Server. If the Web Console Service is started from Fuego Adm Center application, the Windows Service will be automatically started.

	2005 Fuego Inc. - Company Proprietary and Confidential.	
	All Rights Reserved.	Page 13 of 92

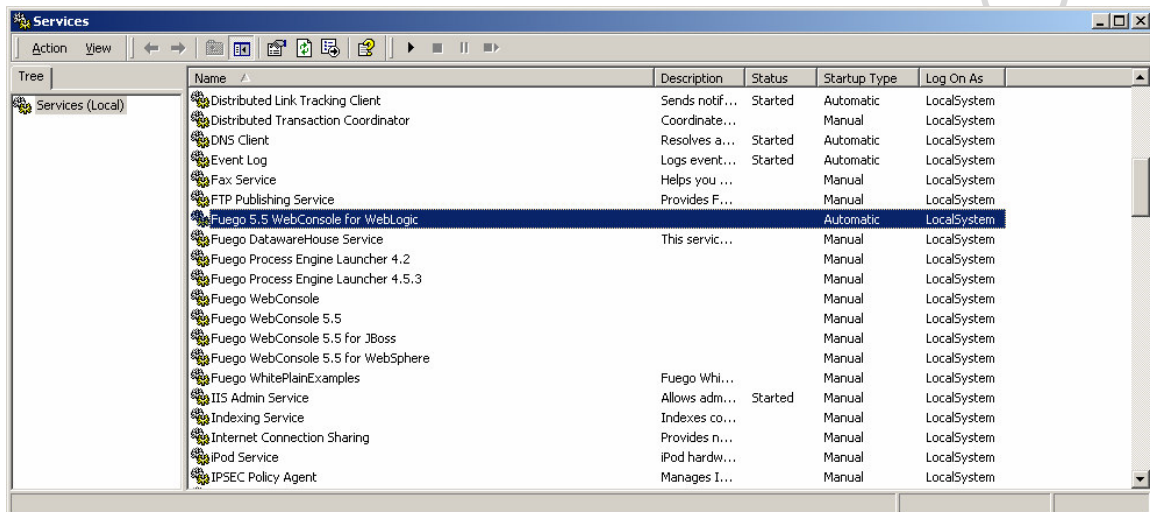
Click on the “Start Web Application Server” link on the main Fuego Adm Center Panel as shown below.



The main panel should look like the one below after a successful Web Console Service startup.



The Fuego Administrator may also want to check the Web Console Windows Service was properly started by checking the Windows Services Panel as depicted in the figure below.

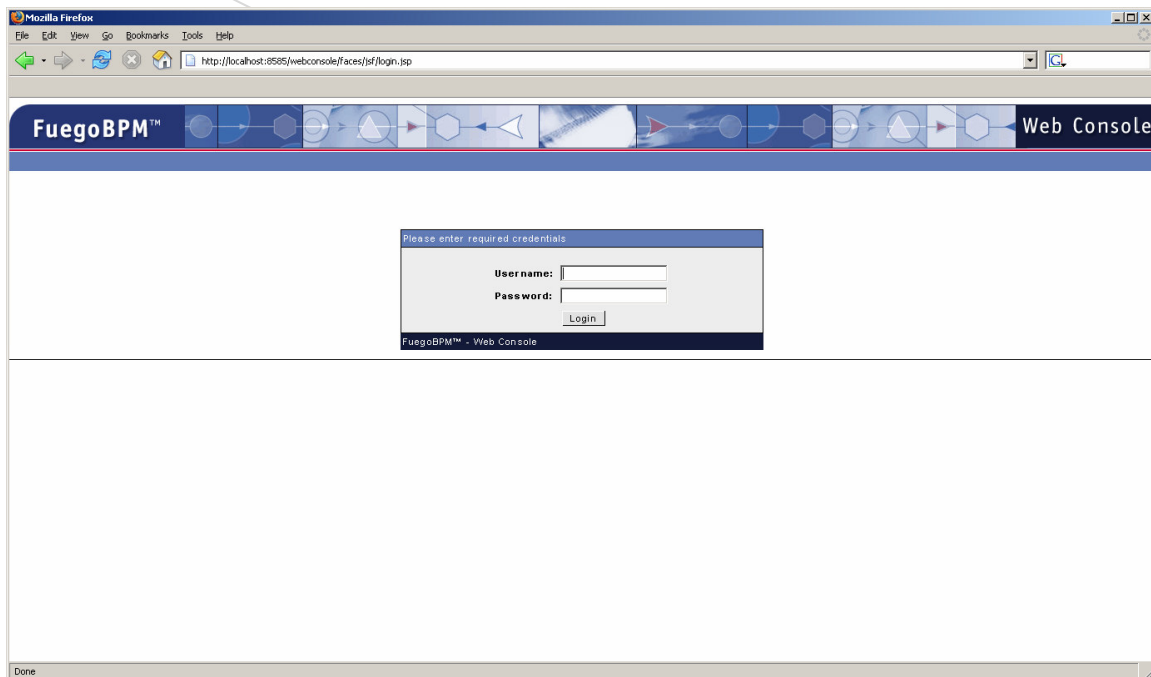


Note that manual start of application from Administration Center is recommended only in test environments (application will stop if you close Admin Center). For production installations, start/stop shells are used (or windows service when installed in that OS).

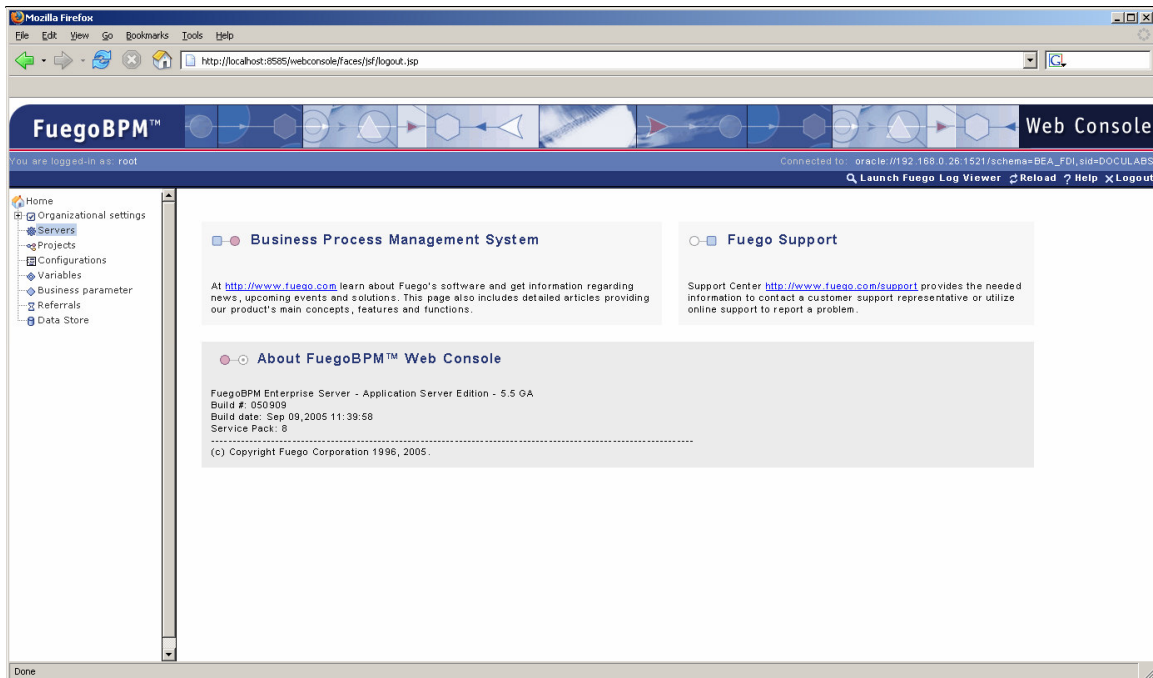
Configuring a Fuego Server on Web Logic Application Server

The next step in the setup is to configure the Fuego Server. This configuration is achieved by using Fuego Web Console. The previous sections of the document helped on configuring and starting this Web Application.

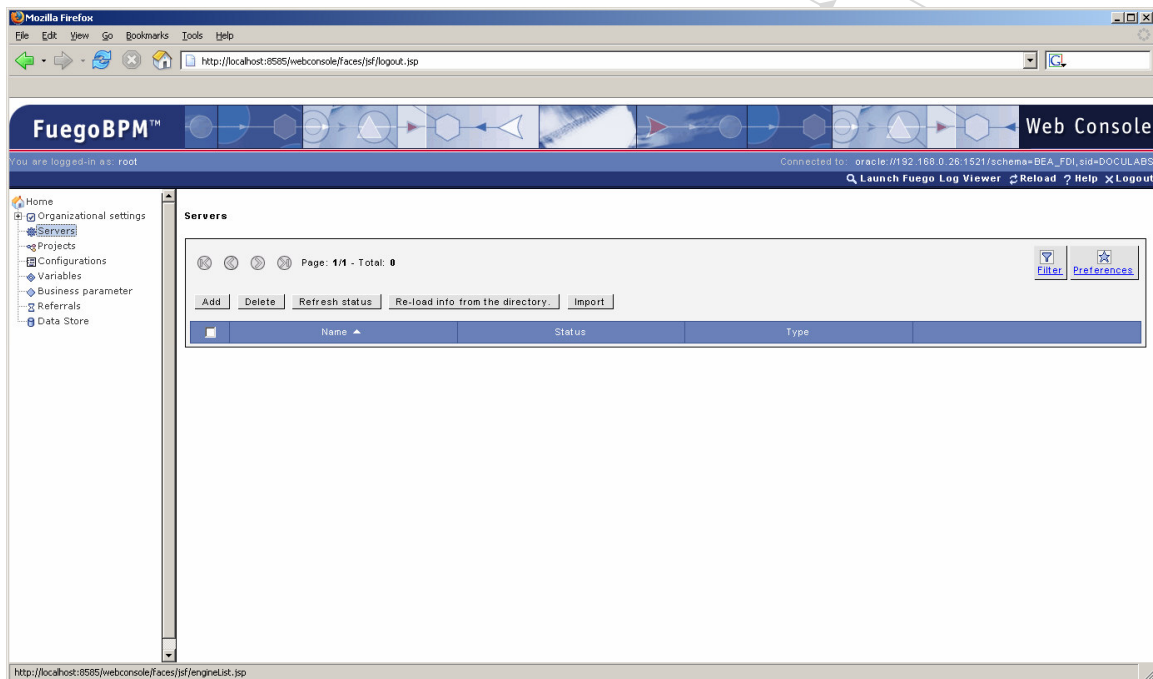
The Fuego Administrator should use a browser to connect to the Fuego Web Console as illustrated below. The URL <http://host:port/webconsole> should be used, where host is the machine in which the Web Console Windows Service has been installed and port is the TCP/IP port configured for the Web Console.



The Administrator user ID provided when creating Fuego Directory Service should be provided in the “Username” field and its password in the “Password” field to log into Fuego Web Console. Click on the “Login” button to proceed with the login. The following is the welcome Web Console Panel.

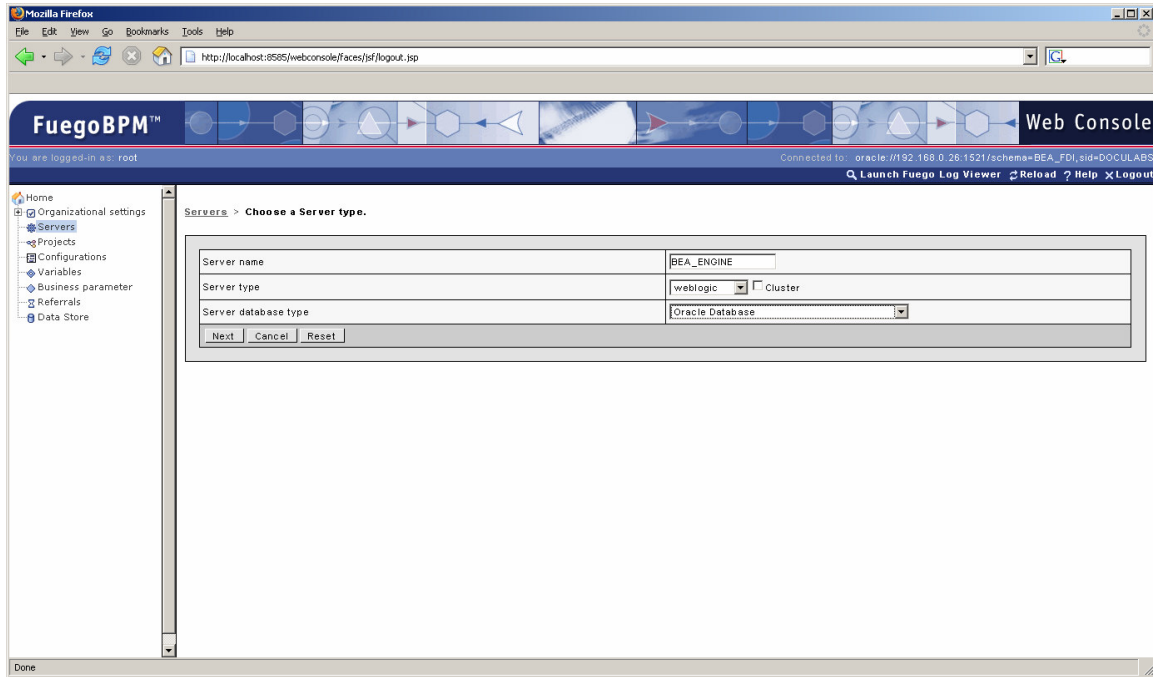


Click on the “Servers” link on the left hand side panel to create a Fuego Server that will run on BEA Web Logic Application Server. Click the “Add” button to start creating the Fuego Server as depicted in the figure below.

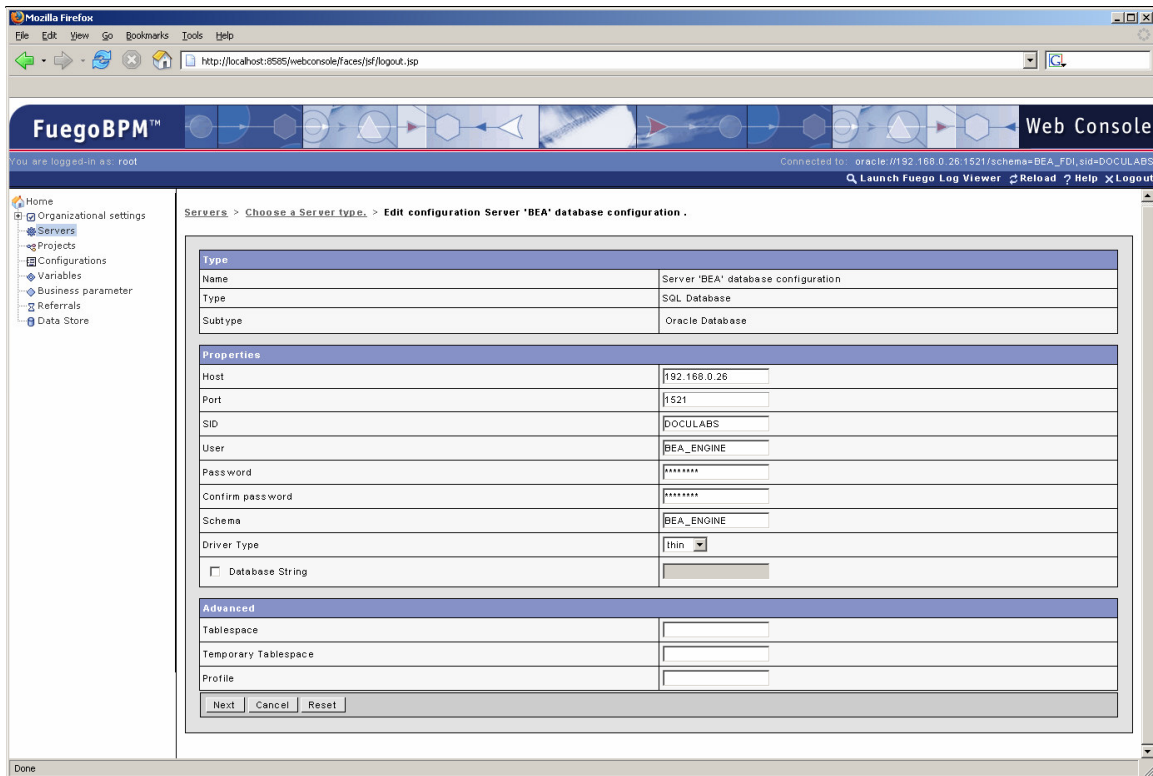


After clicking on the “Add” button, a new panel will be presented to select the Server type to be created. The panel below shows the fields and values selected for our

example. In our case, we will define the Fuego Server to be of webLogic type as well as using Oracle as its target backend RDBMS for business process instance persistence.



Click “Next” to continue. The panel bellows requires the Fuego Administrator to provide information for the Fuego Server backend RDBMS (in our case Oracle) that will keep track of every single event taking place for business process instances.



Web Console

You are logged-in as: root

Connected to: oracle://192.168.0.26:1521/schema=BEA_FDI,sid=DOCULABS

Launch Fuego Log Viewer Reload Help Logout

Servers > Choose a Server type > Edit configuration Server 'BEA' database configuration .

Type	
Name	Server 'BEA' database configuration
Type	SQL Database
Subtype	Oracle Database

Properties	
Host	192.168.0.26
Port	1521
SID	DOCULABS
User	BEA_ENGINE
Password	*****
Confirm password	*****
Schema	BEA_ENGINE
Driver Type	thin
<input type="checkbox"/> Database String	

Advanced	
Tablespace	
Temporary Tablespace	
Profile	

Next Cancel Reset

The following is a list of the panel field with a brief description for each one of them. These fields are specific for Oracle RDBMS. Different fields may be presented if selecting another Fuego Server backend RDBMS.

- **Host:** Host where the Oracle RDBMS is deployed.
- **Port:** TCP/IP port where the Oracle RDBMS is waiting for incoming connections. For Oracle RDBMS, this port is usually 1521.
- **SID:** This is the Oracle SID where the Engine Tables will reside. In our case, the Oracle SID for the Oracle instance being used is: "DOCULABS".
- **User:** This is the Oracle User in the selected SID that will host the Engine tables.
- **Password:** This is the password for the User specified in the previous field.
- **Schema:** It is Fuego convention to use the same value as specified for the User.

The values in the "Runtime" sub-panel should be left unchanged since they are not used when creating a Fuego Server to be deployed on Web Logic. Click "Next" to proceed with the Fuego Server creation.

The following panel will show the name for BEA Web Logic resources used internally by the Fuego Server deployed on Web Logic. These resources may be modified if BEA Administrator already has naming conventions. BEA Administrator will be in charge of creating these resources after successfully completing the Fuego Server creation. Instructions on how to create these resources will be provided in the next section of the document.

The screenshot shows the FuegoBPM Web Console interface. The browser window is Mozilla Firefox, displaying the URL `http://localhost:8505/webconsole/faces/jsf/logout.jsp`. The page title is "FuegoBPM™ Web Console". The user is logged in as "root". The page is connected to an Oracle database at `oracle://192.168.0.26:1521/schema=BEA_FDI, sid=DOCULABS`. The navigation bar includes "Launch Fuego Log Viewer", "Reload", "Help", and "Logout".

The main content area shows the "Servers" configuration page, specifically the "Edit configuration Server 'BEA' database configuration" page. The page is titled "J2EE info".

The configuration summary table is as follows:

Summary of the configuration needed in your weblogic application Server	
Server runtime datasource lookup name	BEA_XABEA_ENGINEDS
Server FDI datasource look-up name	BEA_XAFDIDS
JMS Queue name	BEA_ToDoQueue
JMS Queue connection factory	BEA_XAConnectionFactory
JMS Topic name for Server news	BEA_EngineNews
JMS Topic connection factory	BEA_XATopicConnectionFactory

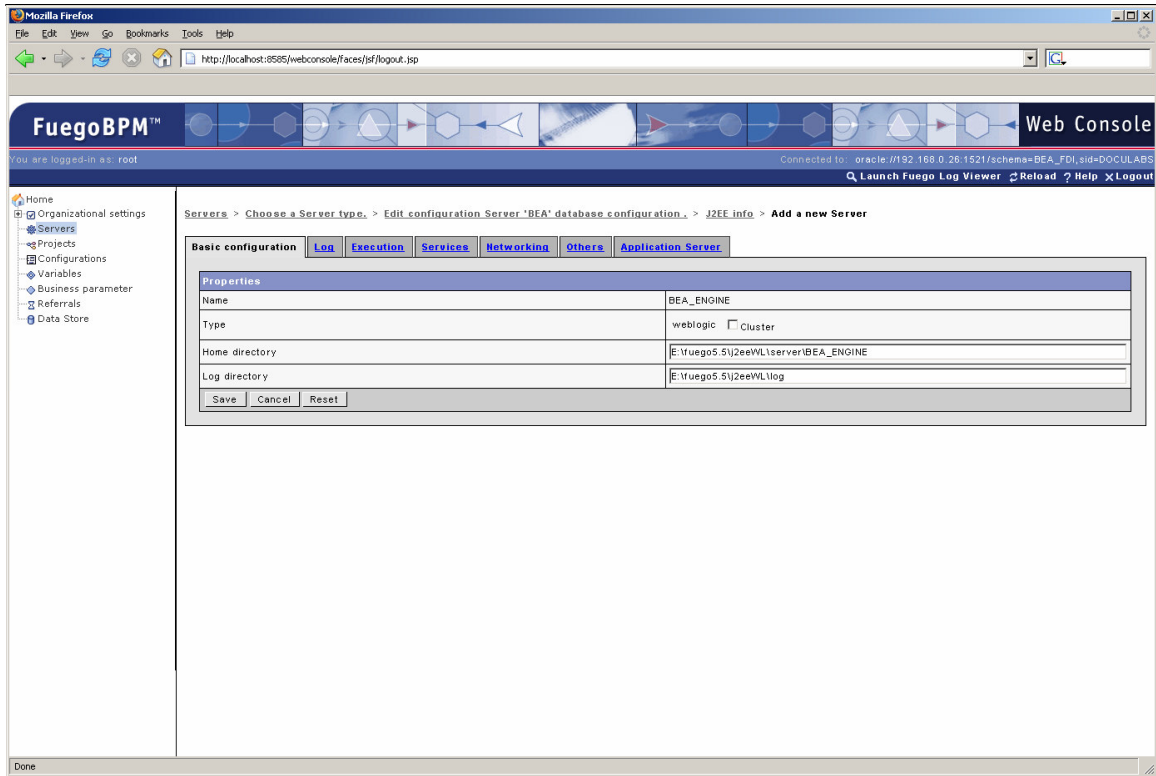
Below the summary table is the "FuegoBPM™ deployer for WebLogic configuration" section:

FuegoBPM™ Deployer URL	http://192.168.0.91:7001/fuego/deployer/servlet/worker
WebLogic server or cluster name	

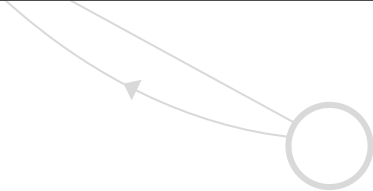
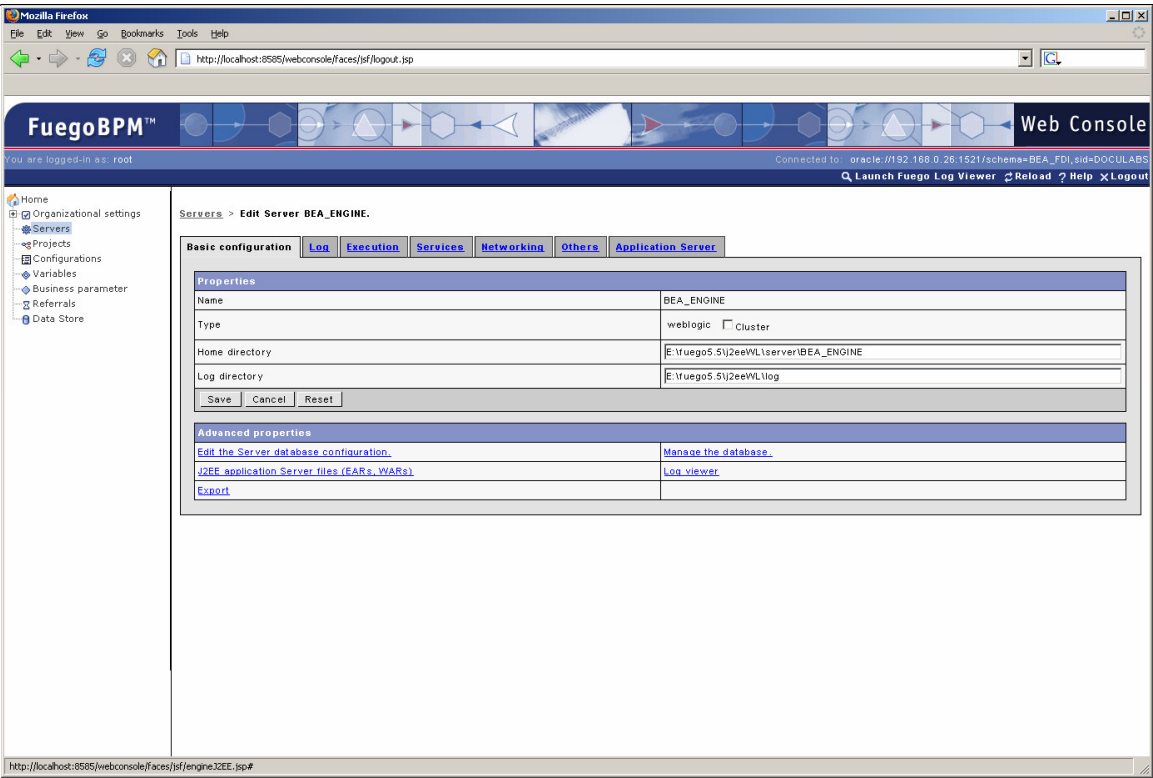
At the bottom of the configuration section are three buttons: "Next", "Cancel", and "Reset". A callout line points from the "Next" button to the text below.

Click “Next” to continue with the Fuego Server creation.

Next, Fuego Administrator should define the basic configuration parameters for the recently created Fuego Server. The following panel shows the minimal mandatory fields to complete the Fuego Server creation.



Click “Save” to persist the Fuego Server settings and finish the Fuego Server creation. The following panel will be displayed after successfully creating the Fuego Server settings.

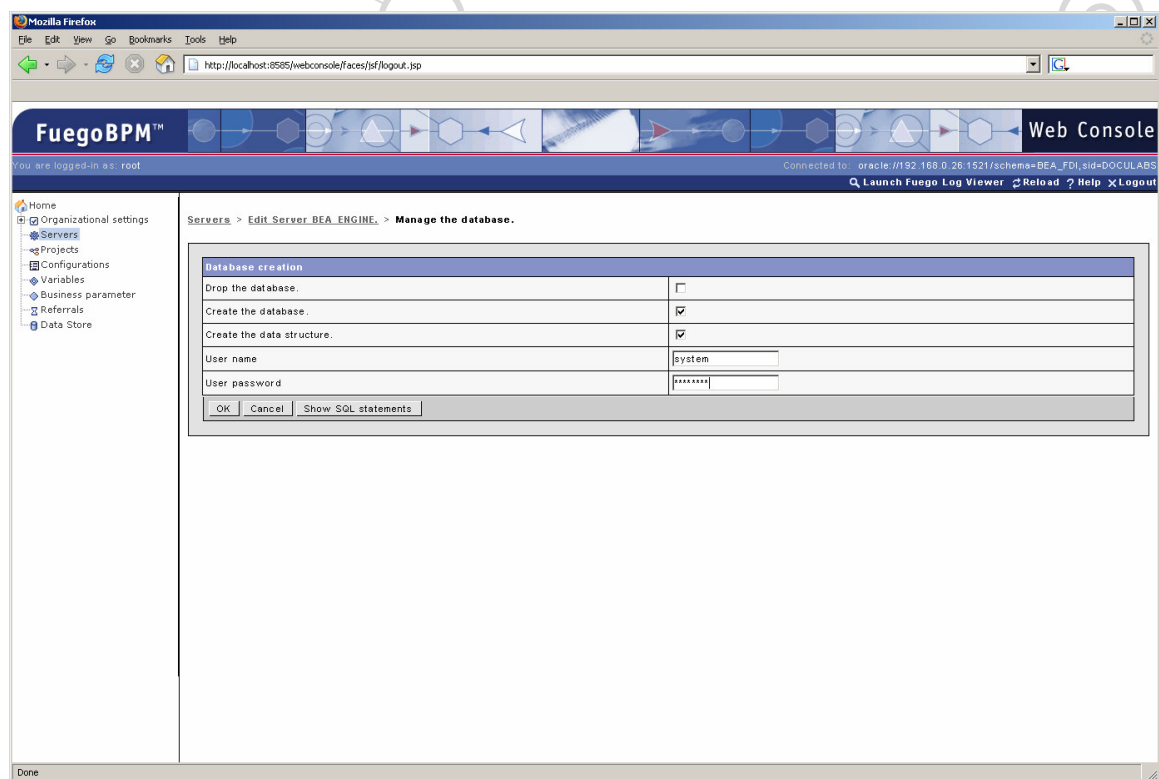


Creating the Fuego Server backend RDBMS

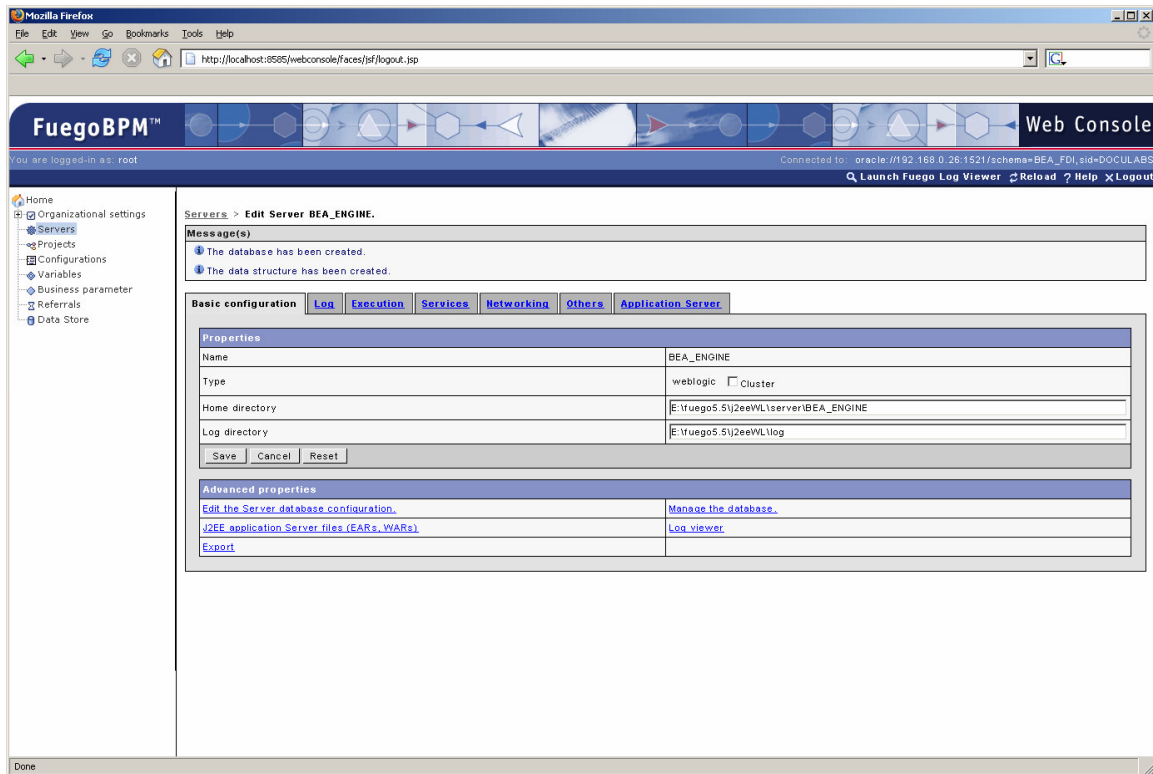
After creating the Fuego Server settings, the Fuego Administrator should proceed to create the backend RDBMS for business process persistence. This is achieved by clicking on the “Manage database” link on the bottom of the panel shown above.

The figure below shows the panel presented to the Fuego Administrator to define the RDBMS creation actions. The checkboxes “create database” and “create data structure” should be selected. Furthermore, a DB administrator user and password with enough permission should be provided by the Fuego Administrator for a proper Database creation.

Click “Ok” to proceed with the Fuego Server creation in the panel shown below.



Upon completion of the triggered action, Fuego Administrator should see the notice messages asserting the Database as well as tables successfully created. The figure below shows this such panel.



In addition it is very important that the Oracle User used for the Engine, you grant SELECT permissions over the following tables for a correct transaction management by BEA WebLogic.

DBA_2PC_PENDING
DBA_2PC_NEIGHBORS
DBA_PENDING_TRANSACTIONS

Configuring Fuego Server Resources on Web Logic Application Server

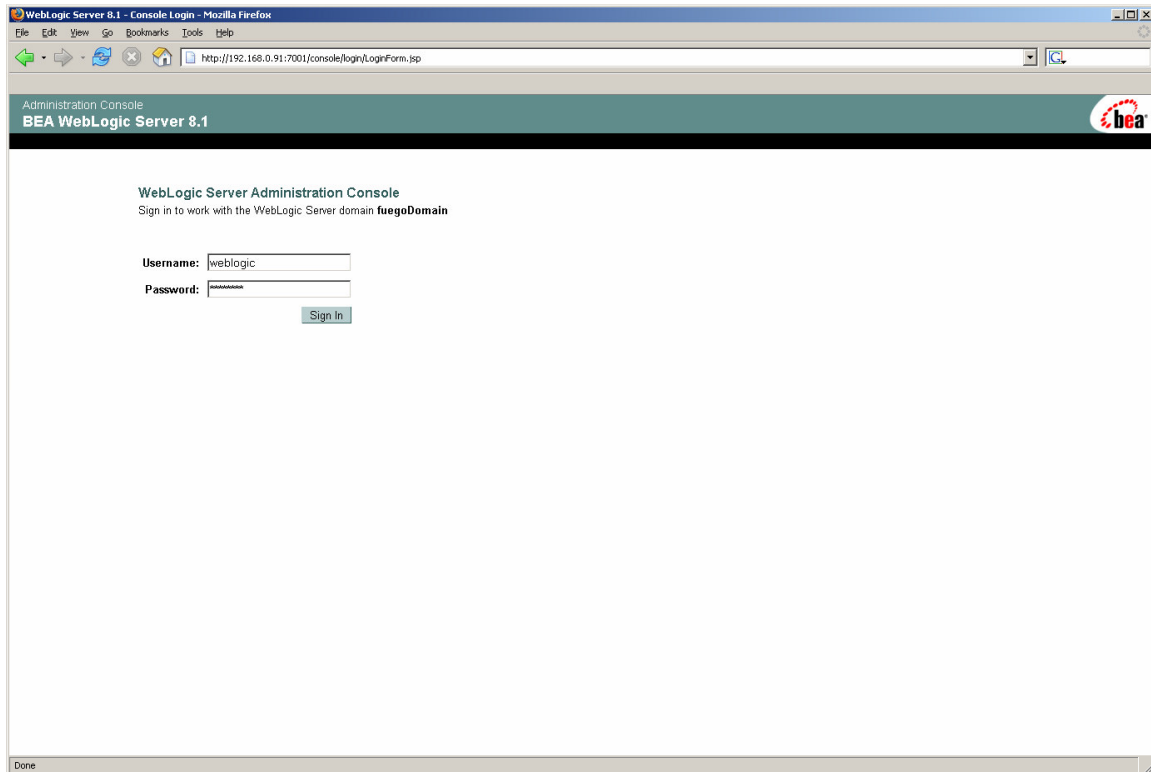
This section will outline how to create the J2EE Resources needed by the Fuego Server to be deployed in BEA Web Logic Application Server. The objective is to use all existing services provided by BEA Web Logic Application Server for centralized resource administration. These resources include creating JDBC Data Sources as well as Topics and Queues for Fuego Server internal asynchronous communication.

It is important to note that if the WebLogic Application Server is running on a Unix environment, the following property should be added to the startWebLogic.sh and startManagedWebLogic.sh scripts, for a correct startup of the Fuego deployed J2EE Applications: -Djava.awt.headless=true

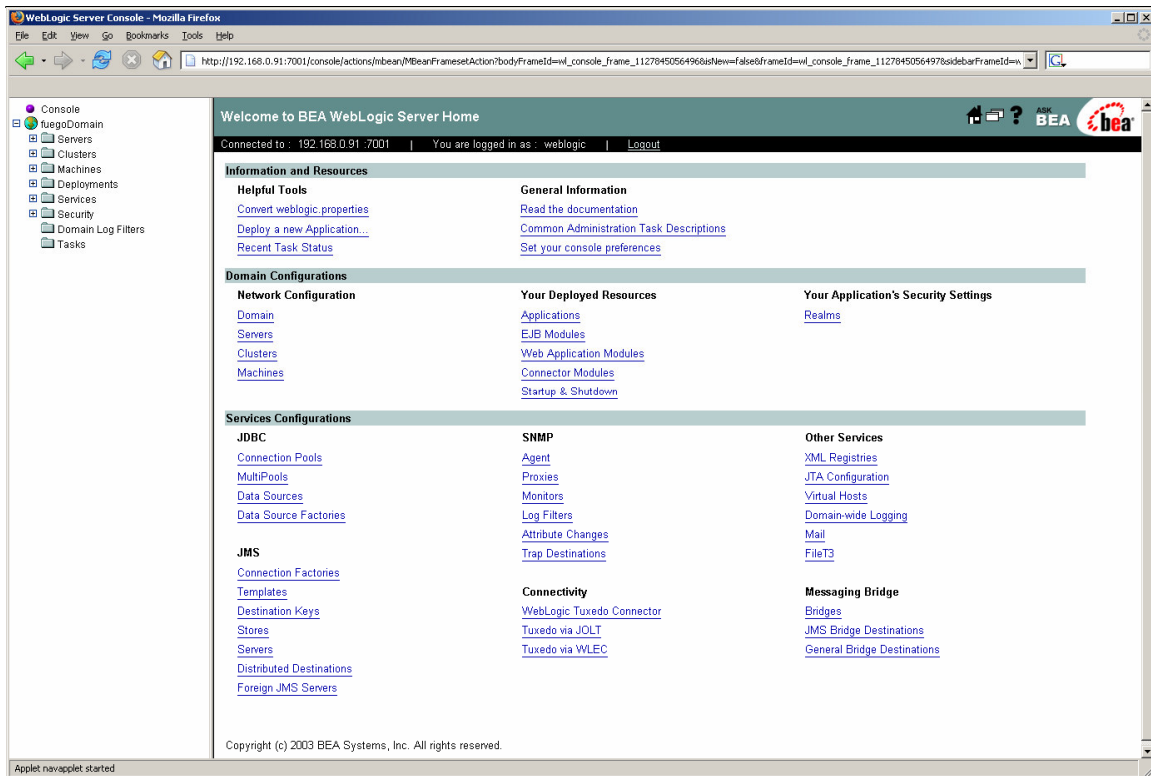
	2005 Fuego Inc. - Company Proprietary and Confidential.	
	All Rights Reserved.	Page 24 of 92

Creating Fuego Directory Service Connection Pool and Data Source

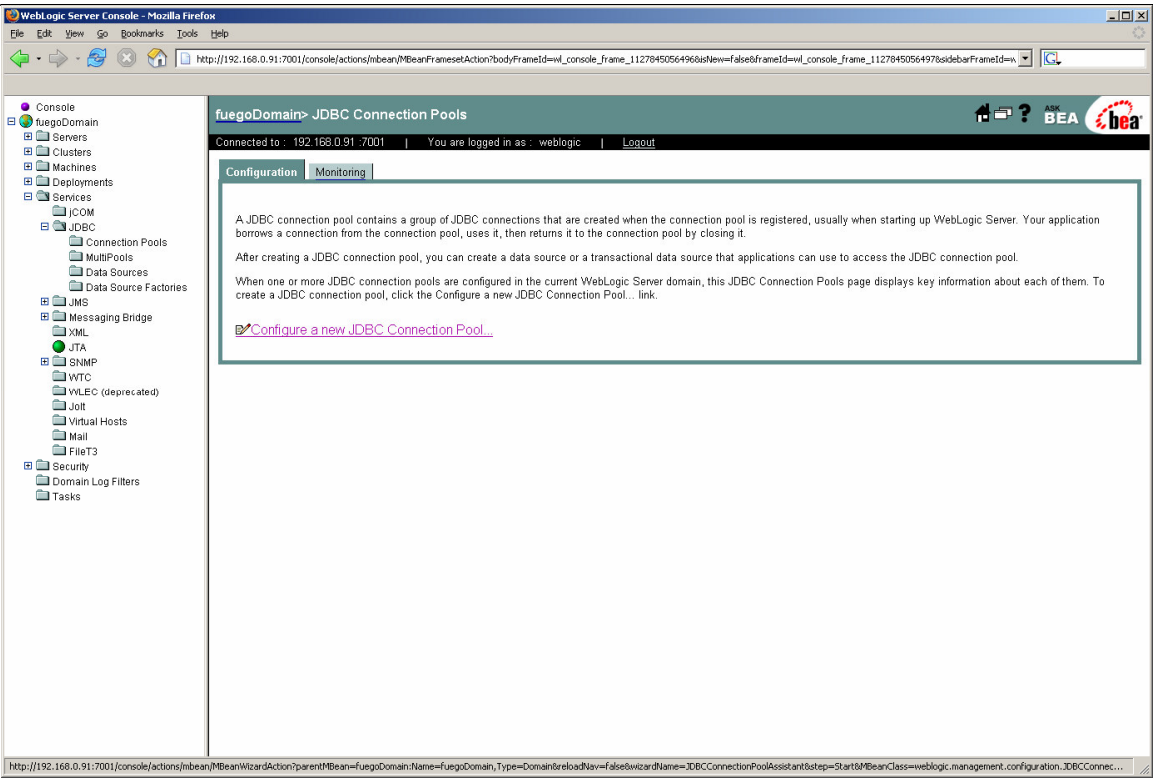
Login into BEA Web Logic Application Server Console using WebLogic Administrator user. Usually, this Administrator id is “weblogic”. By default, BEA Web Logic Console runs on the following URL <http://localhost:7001/console>.



Click on the “Sign In” button to proceed. Once successfully logged in, you will be presented with BEA Web Logic Server Configuration as depicted below.



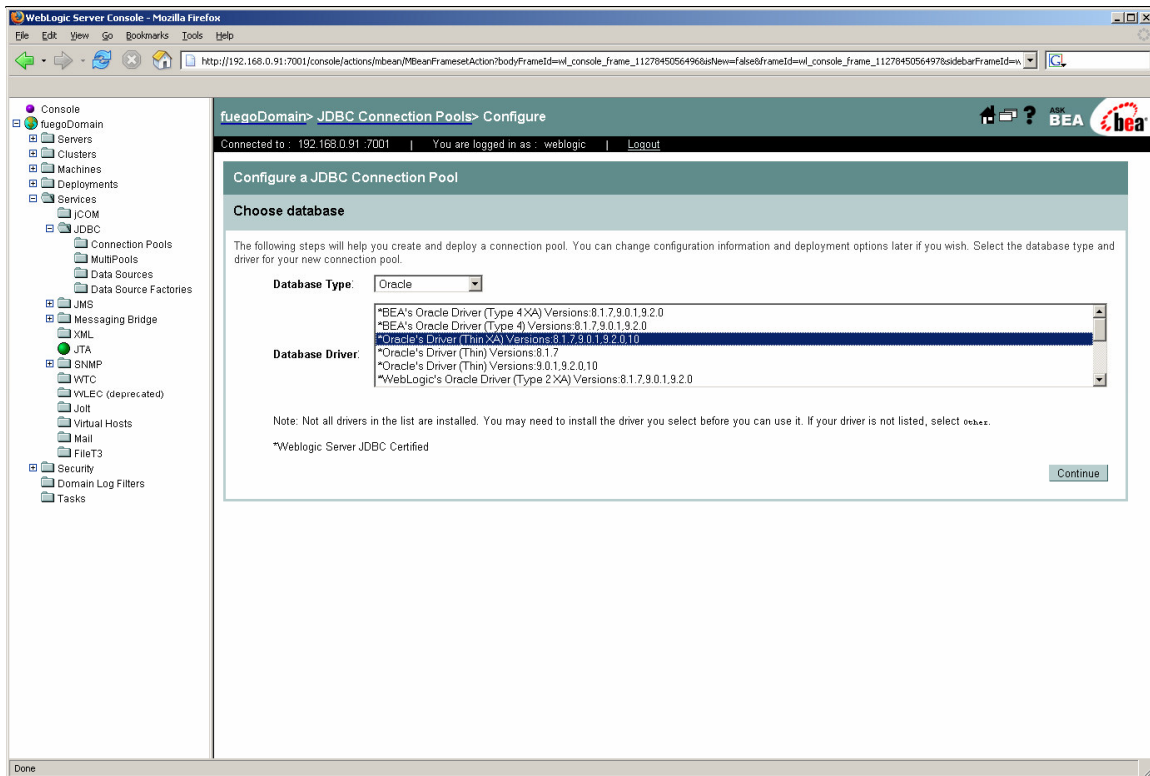
BEA Administrator should create now a JDBC Data Source for the already created Directory Service Database of Fuego, deployed on Oracle RDBMS. First, the Administrator should create a Connection Pool. This can be achieved by accessing the following path in WebLogic Console: Domain/Services/JDBC/Connection Pools and selecting the option “Configure a new JDBCConnectionPool”. This is depicted in the figure below.



As Fuego Directory Service is deployed on Oracle and Fuego needs Oracle Thin XA, the following selection in the new panel is required.

- Database Type: Oracle.
- Database Driver: Oracle Driver (Thin XA) Version: 8.1.7, 9.0.1, 9.2.0, 10.

These selections are also shown in the figure below.



Click on “Continue” to proceed with the Connection Pool creation. The following Panel requires the bindings to connect to the Fuego Directory Service Database. The following values should be provided based on the configuration provided before.

Name: This is the logical name for Fuego Directory Service JDBC Connection Pool. (e.g: Fuego FDI JDBC Connection Pool).

Database Name: This is the name of the Oracle SID where the Fuego Directory Service Database was created in the “Creation Fuego Directory Service” section. In our case, the name is: “DOCULABS”.

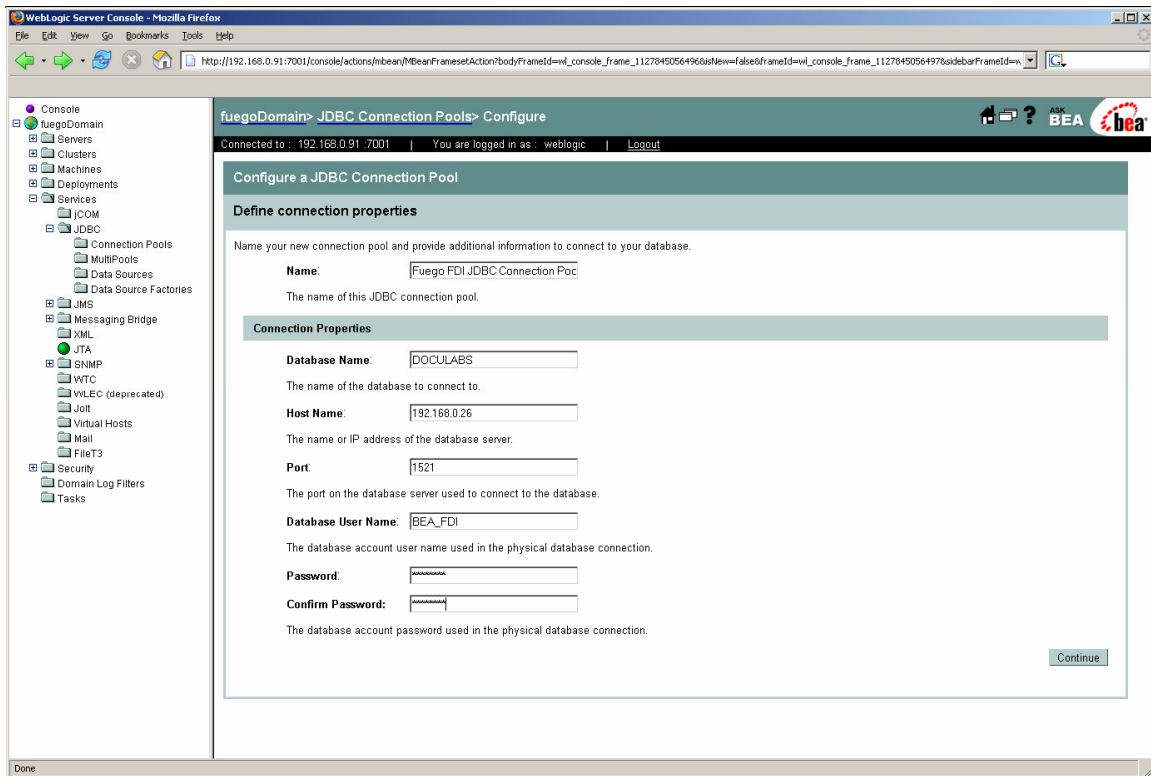
Host Name: This is the host where Fuego Directory Service Database is deployed. (e.g: 192.168.0.26)

Port: This is the TCP/IP Port defined in the MS SQL Server Database for incoming connection.

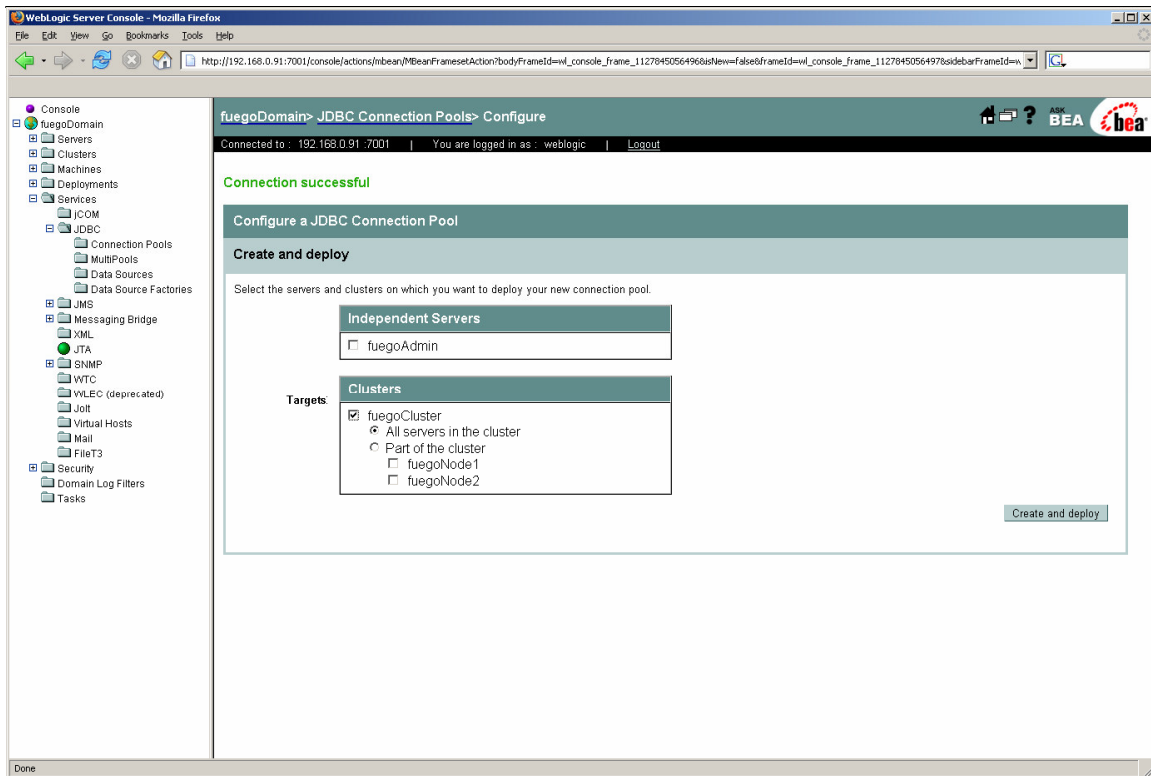
Database User Name: Per Fuego Convention, this is the same as the Database name. In our case, it is BEA_FDI

Password: This is the password for the MS SQL Server Login entity provided in the previous field.

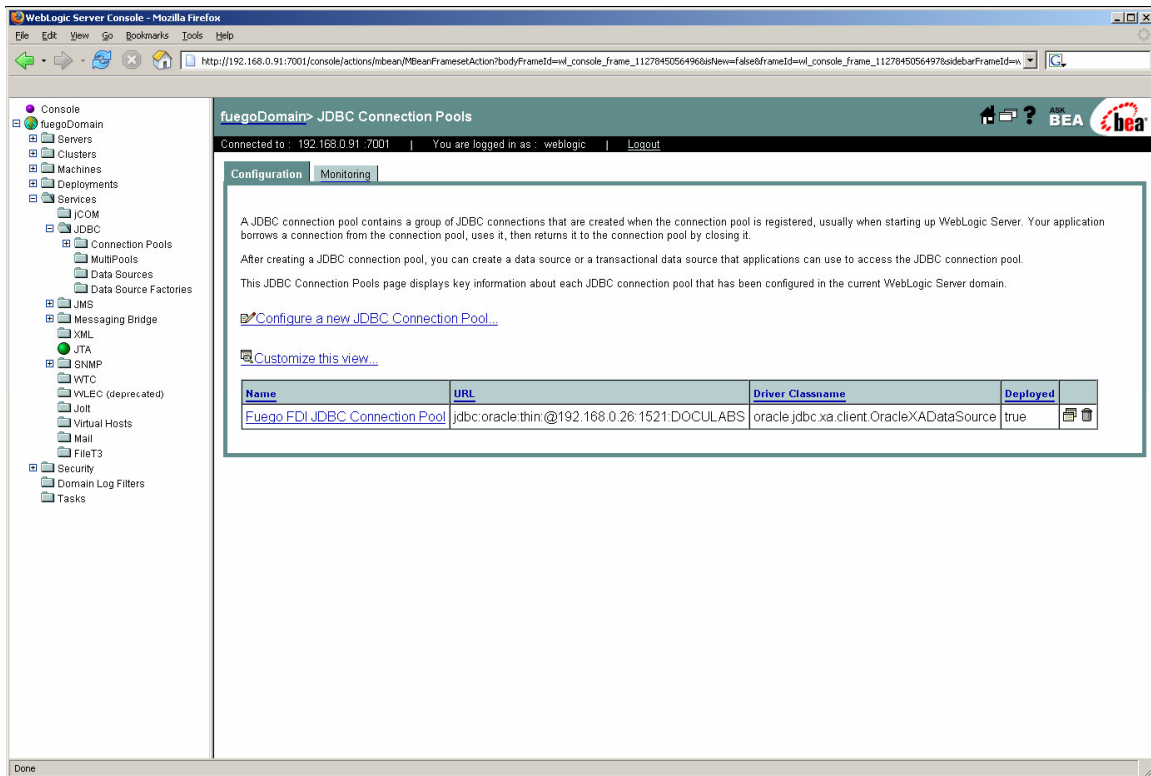
Confirm Password: Same value as provided for the “Password” field.



Once all these fields have been provided, click on the “Continue” button to continue with the testing of the bindings specified for the Connection Pool. The panel shown below allows the testing of this configuration. If for some reason, the JDBC Driver is not found, BEA Administrator should include it in the Application Server CLASSPATH to make this resource available for a correct behavior.

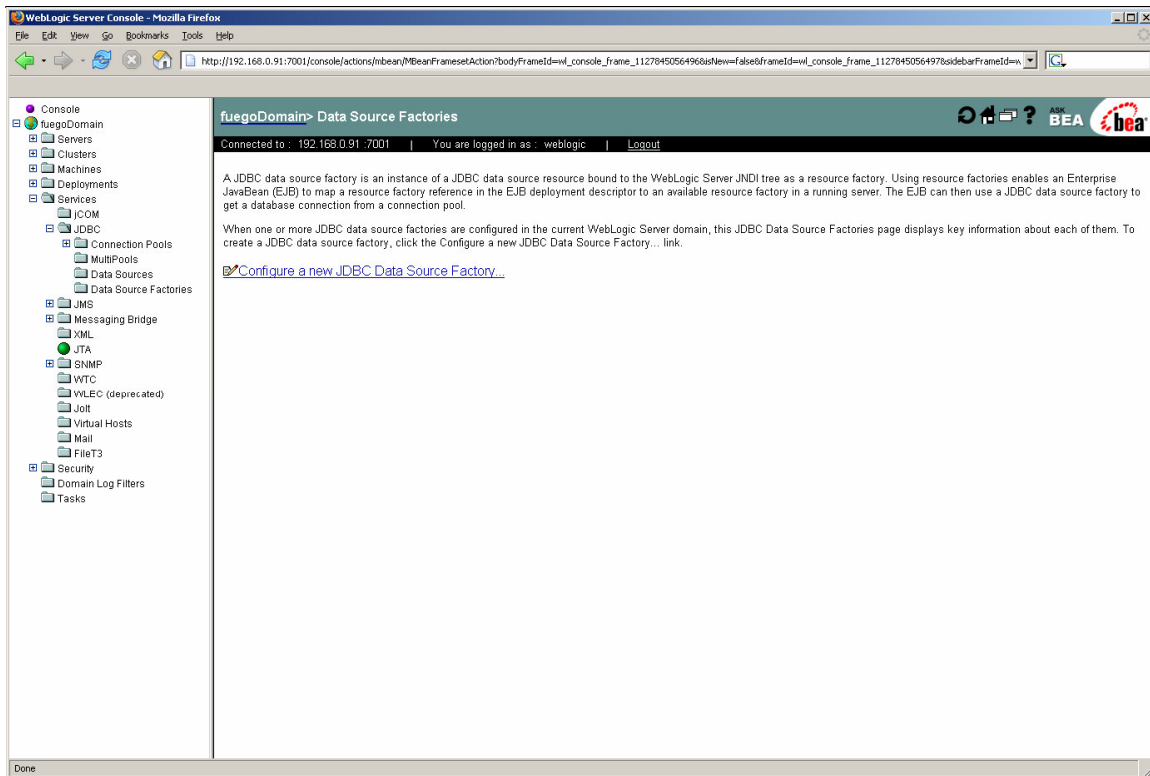


Select the deployment scope (in our case the whole cluster) and then click on the “Create and Deploy” button to finish the Connection Pool creation for Fuego Directory Service. The panel below shows the Connection Pool entry after a successful creation.



Now that the Connection Pool has been created, BEA Administrator should proceed to create the Data Source connected to this recently created Connection Pool. To create the Data Source, the following Path may be followed on the left menu: Domain/Services/JDBC/Data Sources.

The figure below shows how to trigger the creation of the Data Source for Fuego Directory Service.



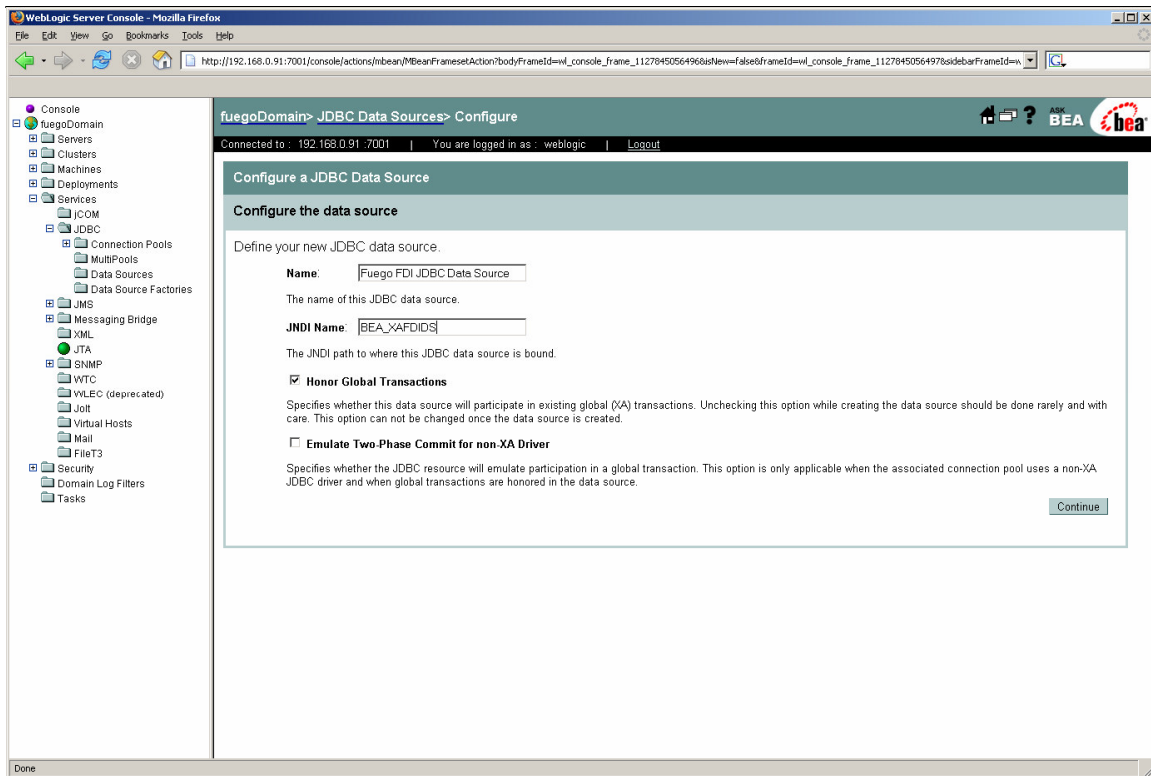
After clicking on the “Configure a new JDBC DataSource...” link, a new panel will be presented to define it. This panel requires the following field values to be provided:

Name: This is the logical name to be provided to Fuego Directory Service Data Source.

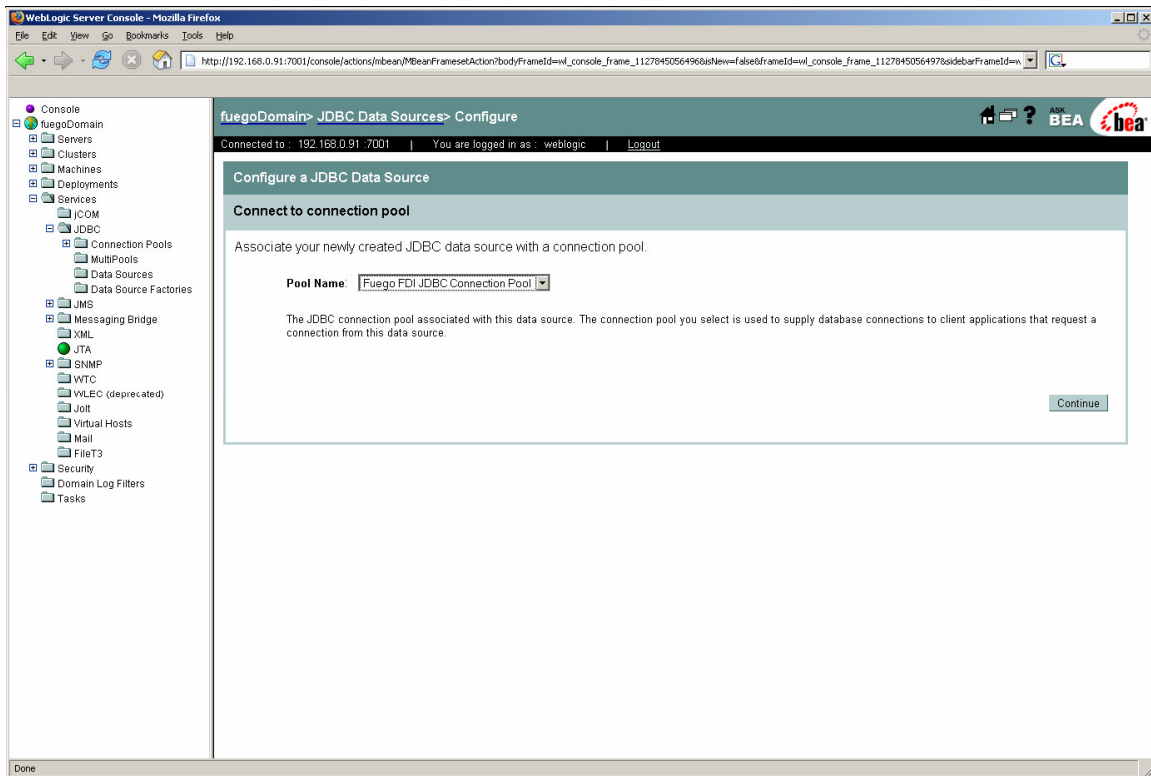
JNDI Name: This text field should be populated with the value presented when creating Fuego Server in Fuego Web Console for the “Engine FDI datasource lookup name” field. In our case, the value proposed by default when configuring Fuego Engine was “BEA_XAFDIDS”.

Honor Global Transactions: This checkbox should be selected.

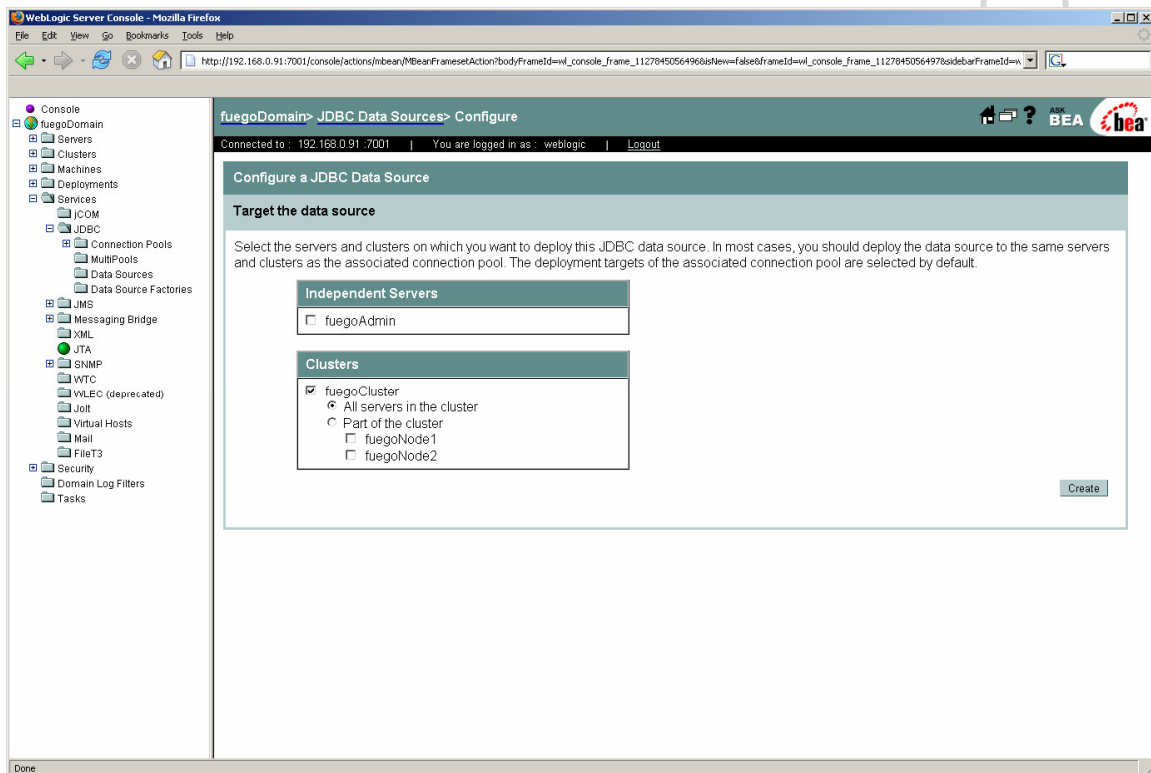
Emulate Two-Phase Commit for non-XA driver: This checkbox should not be selected.



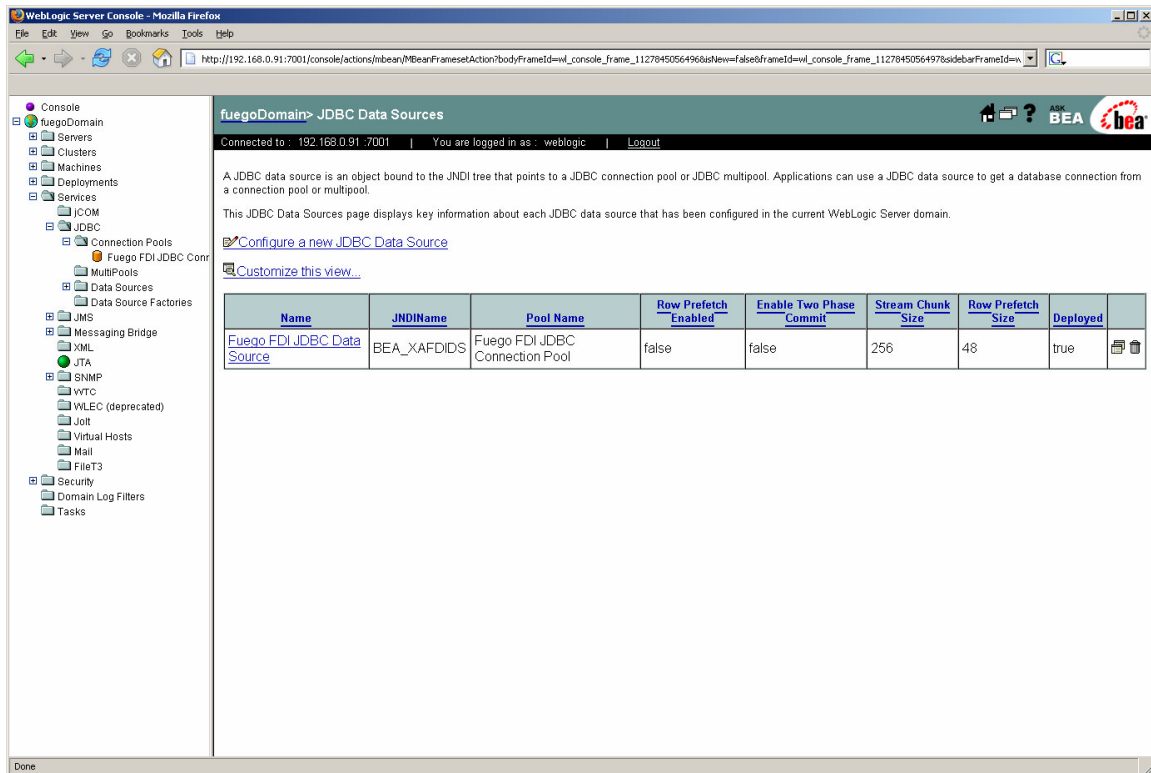
After completing the required fields, click on the “Continue” button to proceed with the selection of the previously created Connection Pool for Fuego Directory Service. The Connection Pool selection is shown in the figure below.



After clicking on the “Create” button, the Data Source creation will require you to specify the scope of the deployment.



The following panel shows the entry for the recently created and deployed Data Source.



Creating Fuego Server backend Connection Pool and Data Source

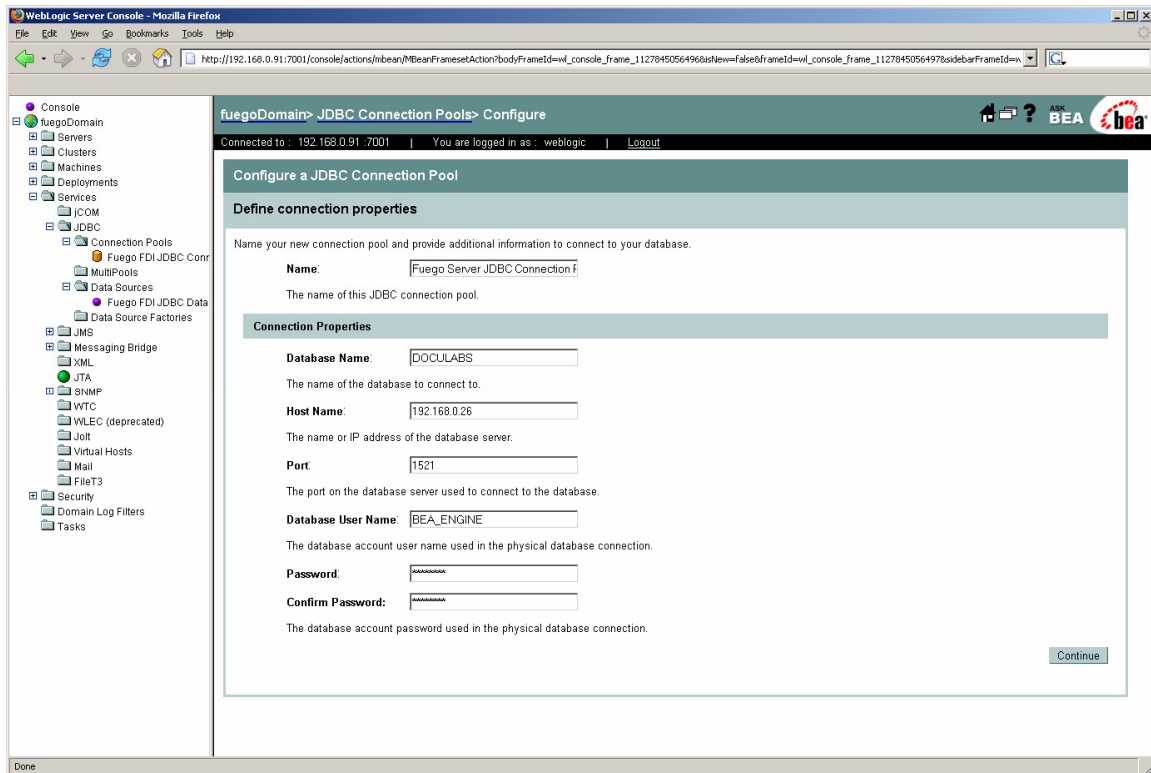
Similar actions should be taken to create Fuego Server Connection Pool and Data Source.

First, BEA Administrator should create a Connection Pool for Fuego Server. This can be achieved by accessing the following path in WebLogic Console: Domain/Services/JDBC/Connection Pools and selecting the option “Configure a new JDBCConnectionPool”. This is depicted in the figure below.

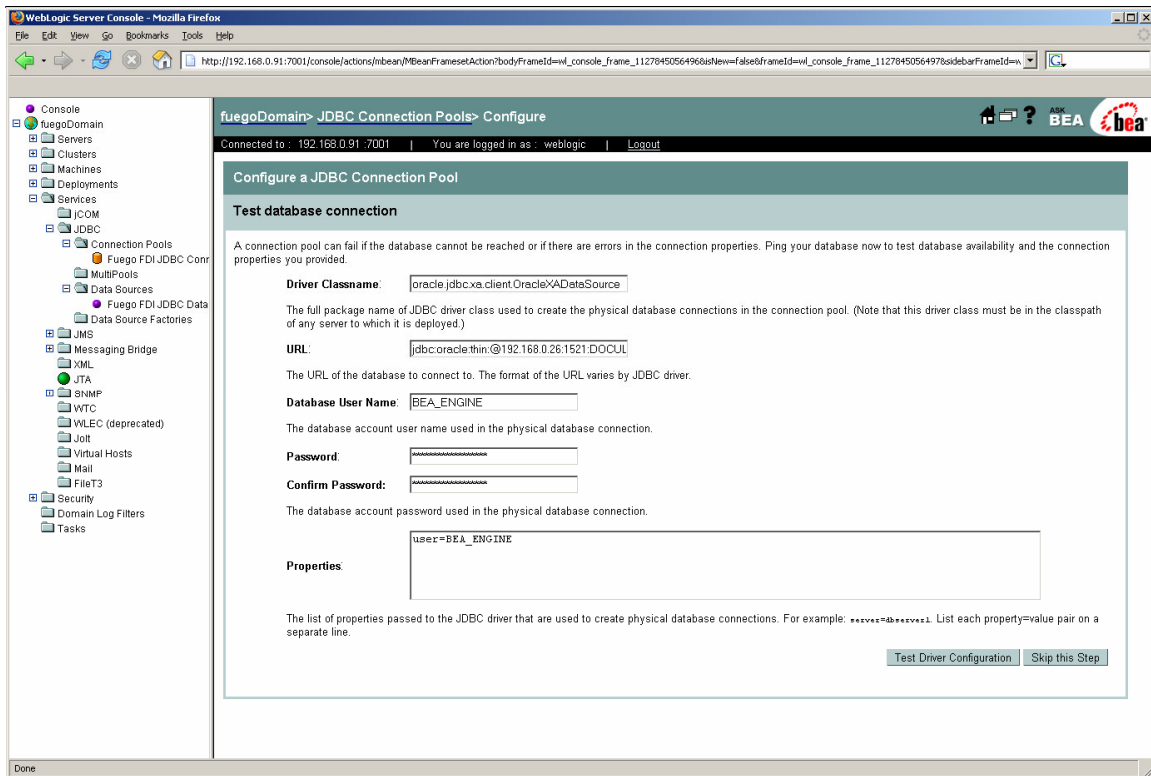
FUEGO



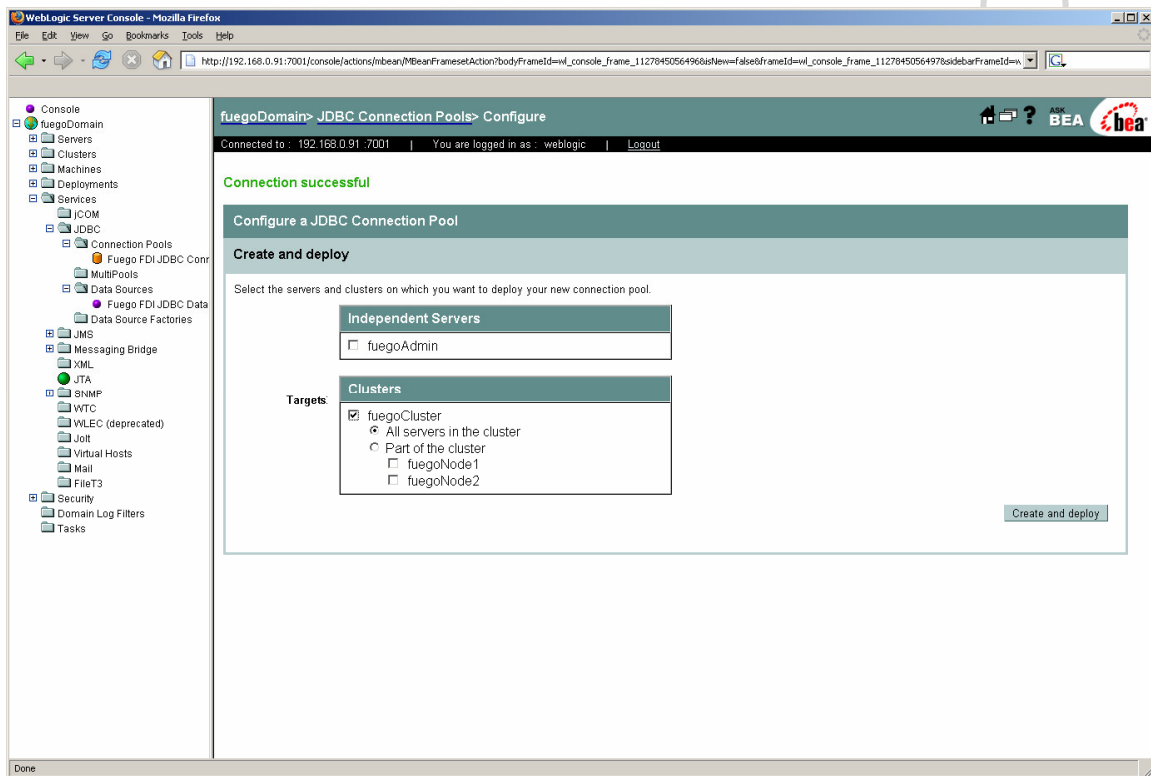
Click “Continue” to proceed. The new panel will request BEA Administrator to provide the binding parameters to connect to Fuego Server Database previously created.



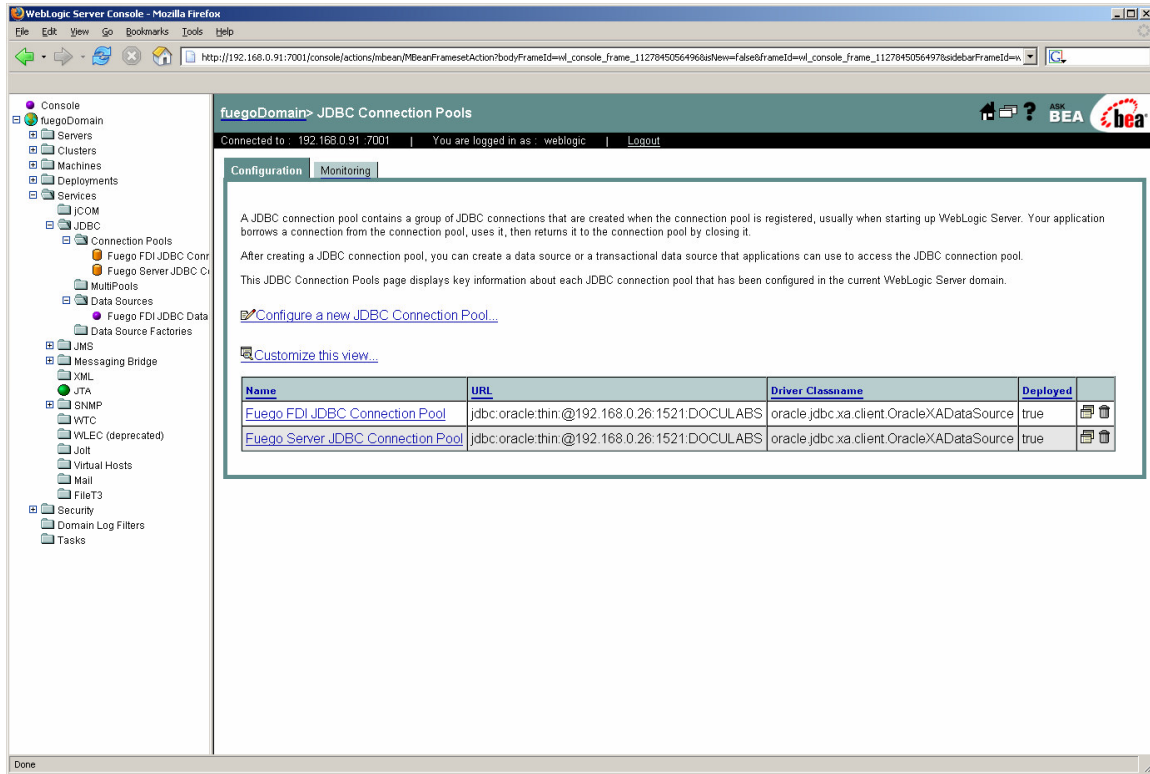
Click “Continue” to proceed. This new panel will enable BEA Administrator to test the recently created Connection Pool. Click on the “Test Driver Configuration” button to check the connectivity with Fuego Server Database.



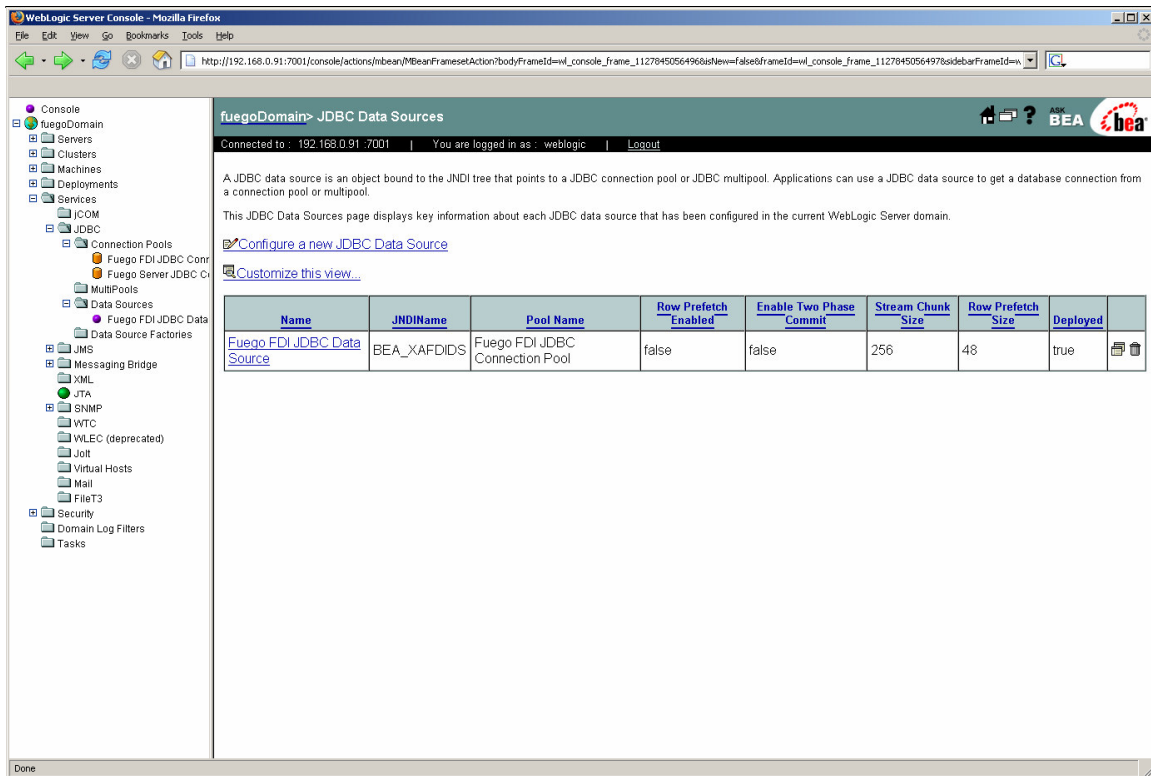
After a successful connectivity test with Fuego Server Database, BEA Administrator should see a panel like the one shown below.



Specify the scope of the deployment and then click on the “Deploy and Create” button to finally create the configured Connection Pool. After successful creation, BEA Administrator should see the panel with the recently created Connection Pool for Fuego Server Database.



Next, BEA Administrator should create the Data Source for the recently created Connection Pool. Right click on the Data Source node and select “Configure a new JDBC DataSource...” link on the prompted menu.



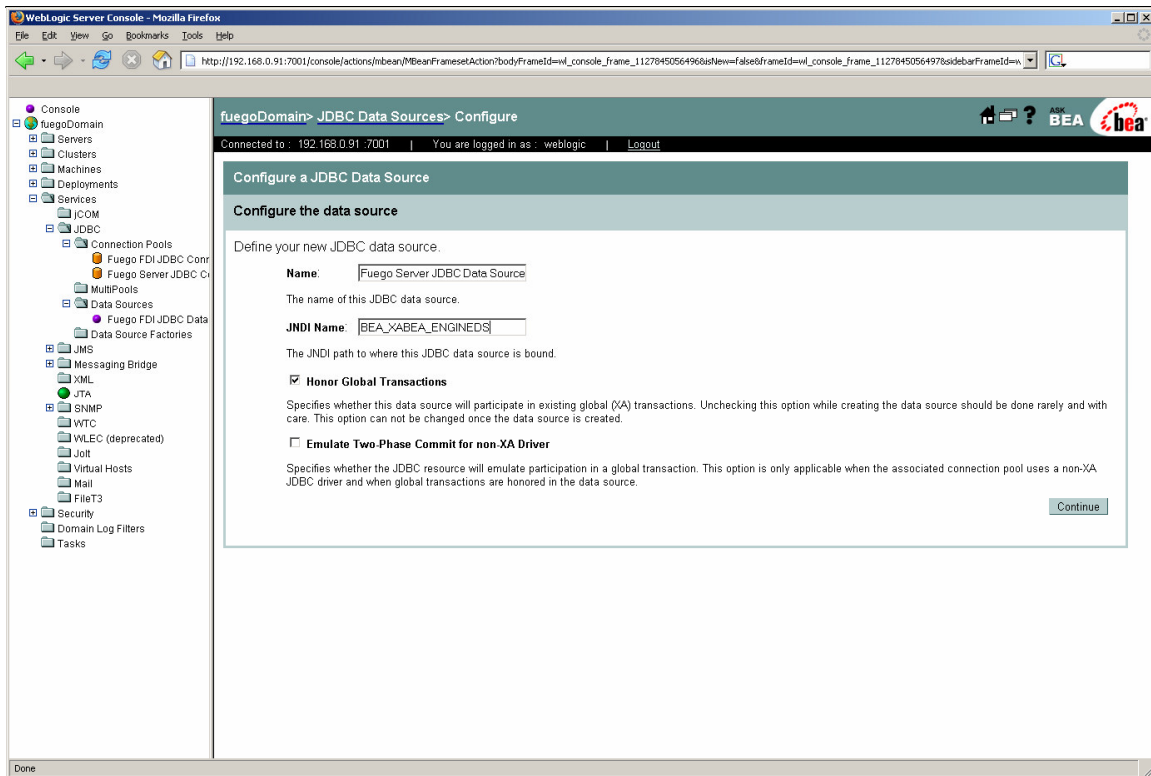
The new opened panel (shown in the figure below) will require the following fields.

Name: This is the logical name to be provided to Fuego Server Data Source.

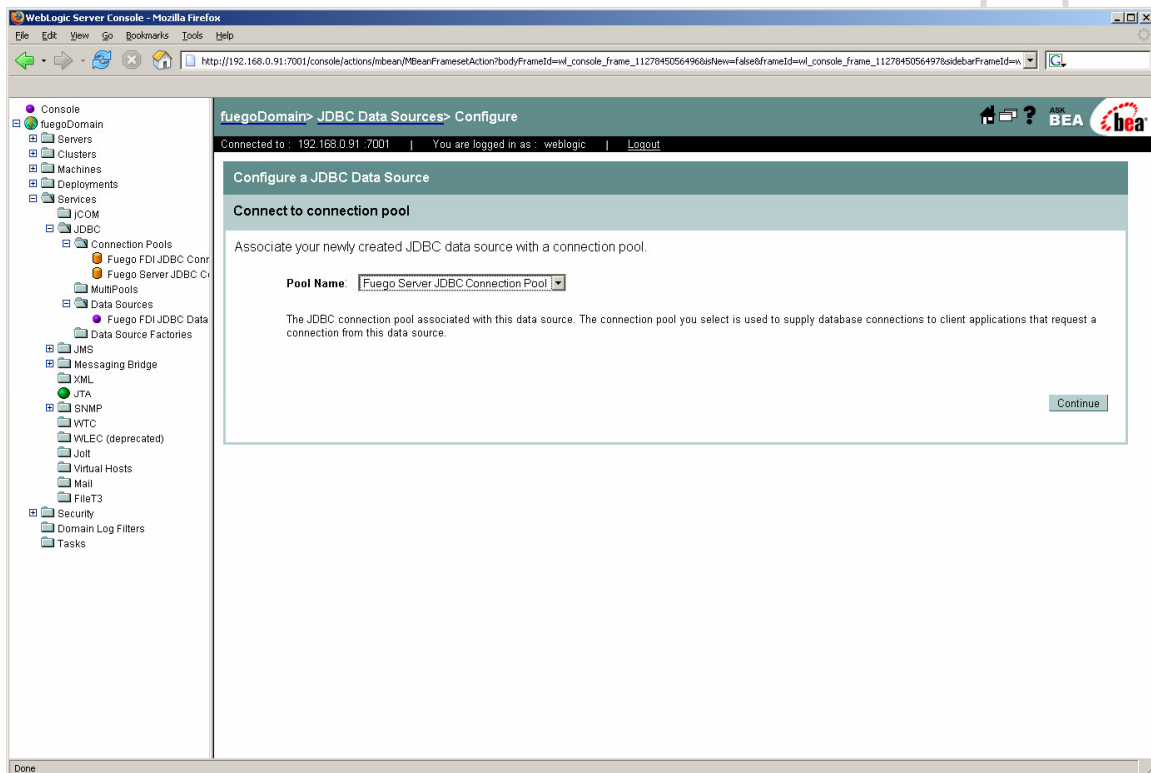
JNDI Name: This text field should be populated with the value presented when creating Fuego Server in Fuego Web Console for the “Engine runtime datasource lookup name” field. In our case, the value proposed by default when configuring Fuego Engine was “BEA_XABEA_ENGINEDS”.

Honor Global Transactions: This checkbox should be selected.

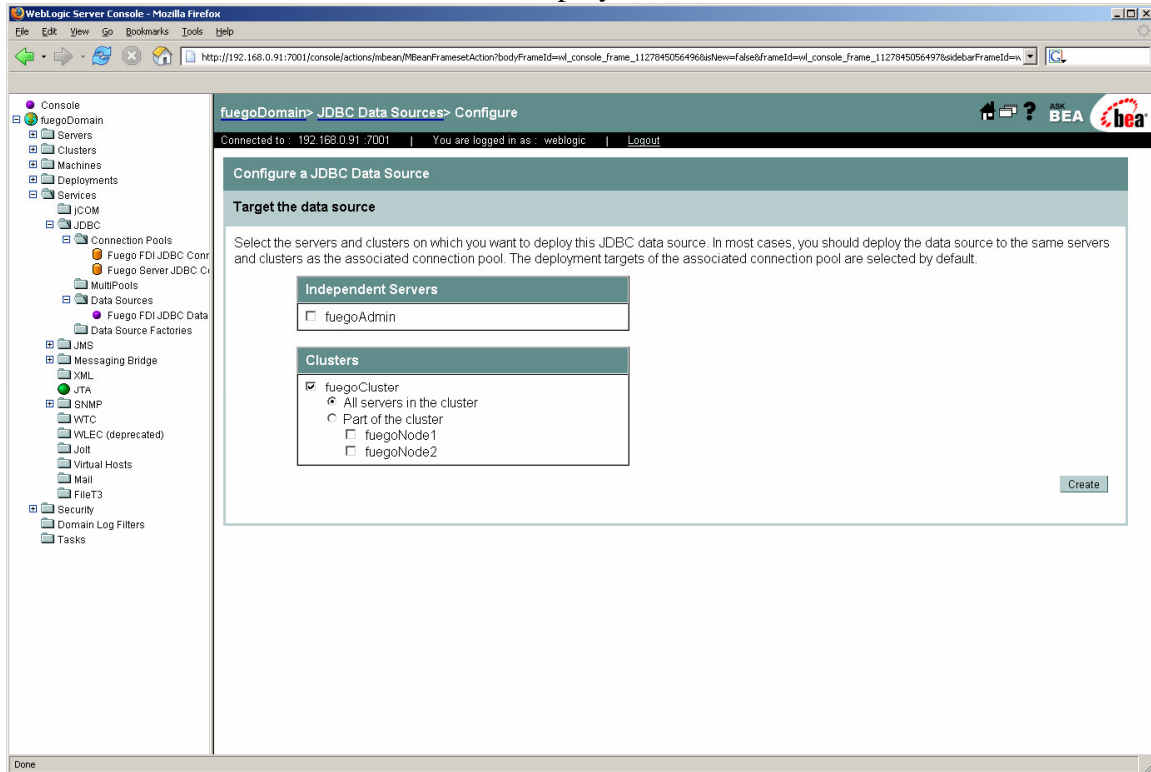
Emulate Two-Phase Commit for non-XA driver: This checkbox should not be selected.



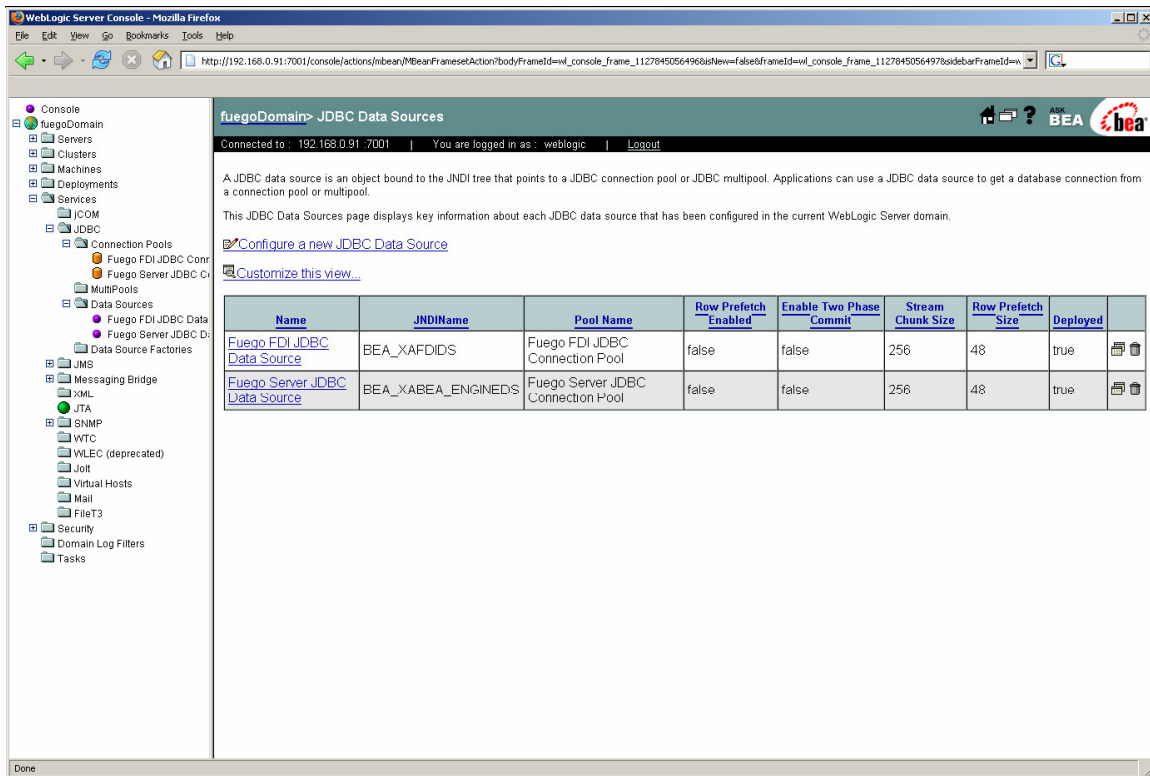
After completing the fields, click “Continue” to proceed. BEA Administrator should select recently created Fuego Server Connection Pool.



Click “Continue” to proceed. The next panel will require selecting the BEA Server where the Data Source should be deployed.



Select the deployment scope and click “Create” button to proceed. After successful creation of Fuego Server Data Source, it should be shown in the Data Source Configuration panel as depicted in the figure below.



Creating Topics and Queues for Fuego Server for Automatic Activity execution and News

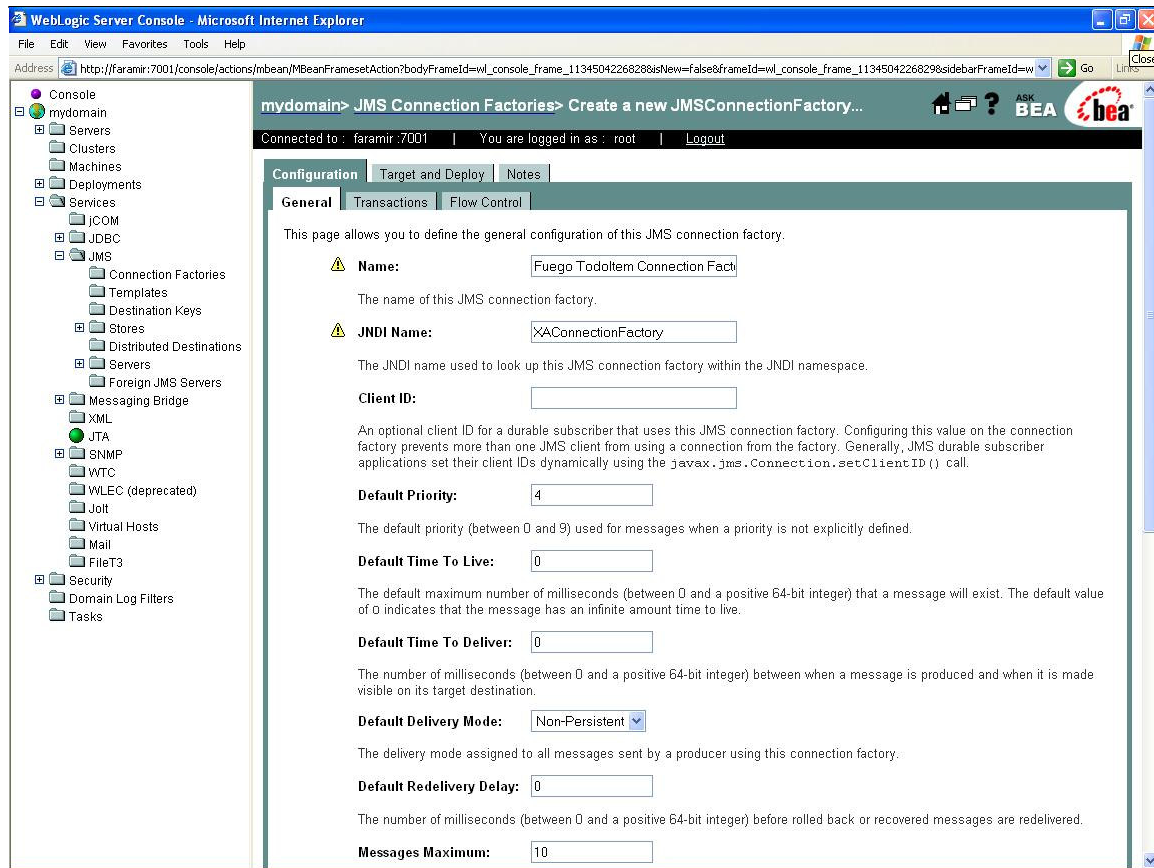
Fuego can use the embedded Web Logic messaging infrastructure, or can rely on a third party one, such as TIBCO EMS. This document explains both configuration procedures.

Topics and Queues for Fuego server (Web Logic infrastructure)

The following steps describe how to add the necessary topics and queues using Web Logic JMS infrastructure.

Creating a Connection Factory for ToDoItems Queue (Web Logic infrastructure)

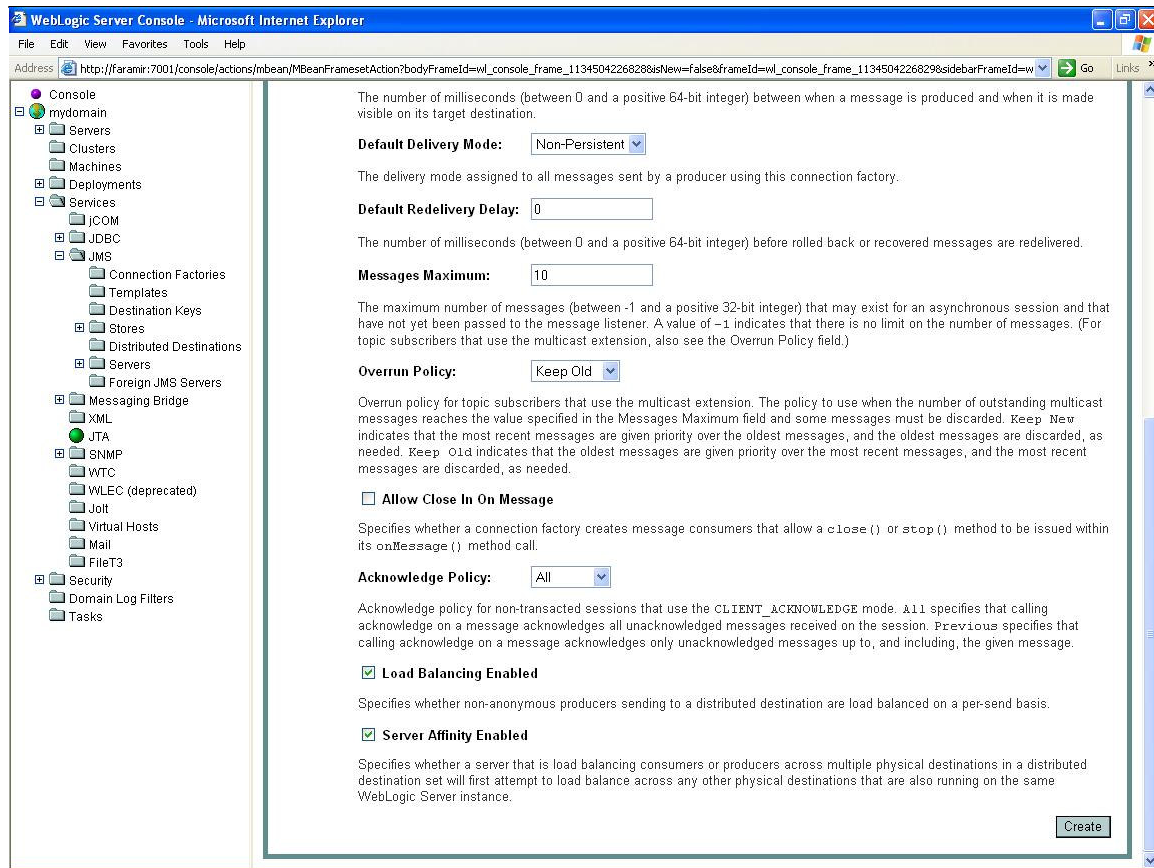
Navigate on the left menu panel through the following hierarchy “Domain/Services/JMS/Foreign JMS Servers/Connection Factories”. Once in the “Connection Factories” node, right click and select “Configure a new Connection Factory...” from the opened menu. This is depicted in the figure below.



The first panel will require BEA Administrator to configure at least 3 parameters that are listed below with a brief description for each one of these:

- **Name:** This is a logical name which should identify Fuego Server ToDoItems Resource.
- **JNDI Name:** This field should contain the value assigned to the “JMS Queue connection factory” configuration parameter specified in Fuego Server “Application Server” Tab. In this case, the default proposed value is: “XAConnectionFactory”.
- **Default Delivery Mode:** Non-persistent

The rest of the fields can accept the default values. Press “Create” to add the new connection factory.



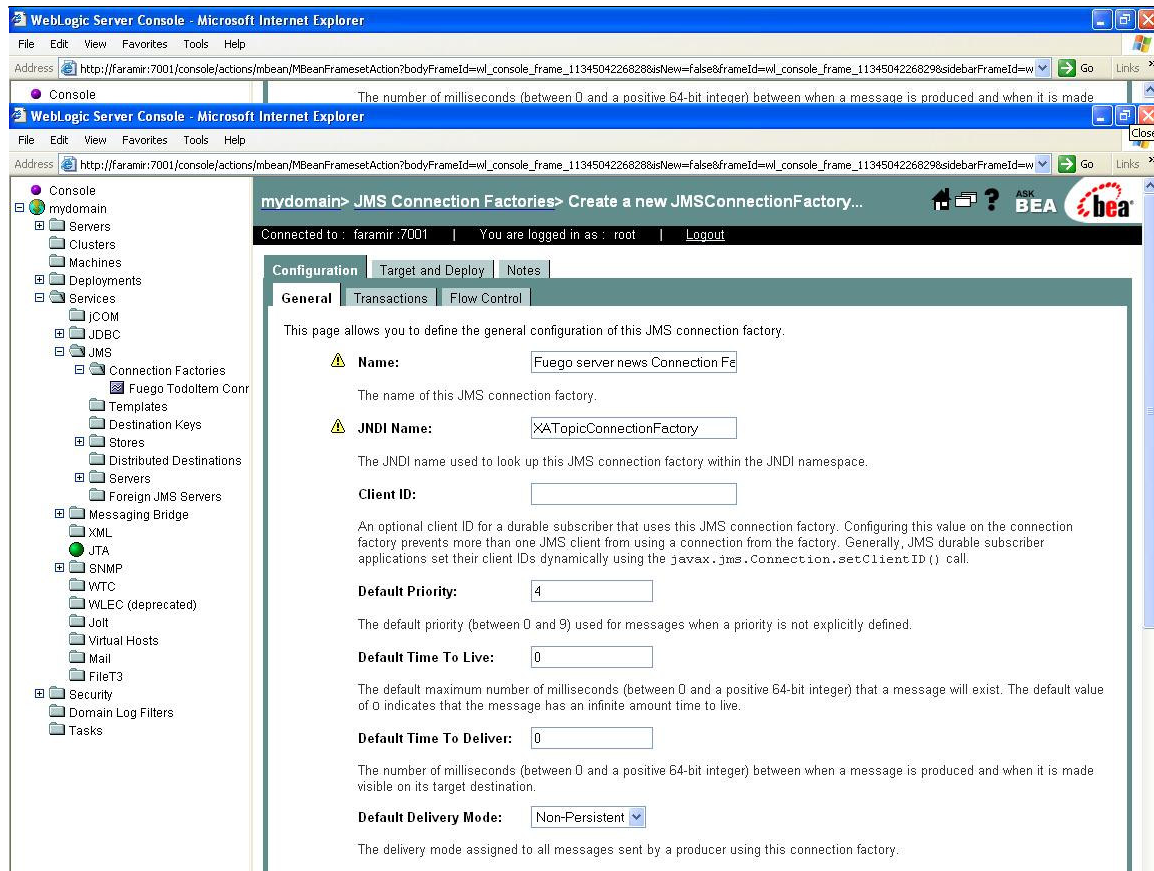
Creating a Connection Factory for Fuego Server News Topic (Web Logic infrastructure)

BEA Administrator should also create a JMS Connection Factory for Fuego Server News Service. The figure below shows the JMS Connection Factory Configuration panel.

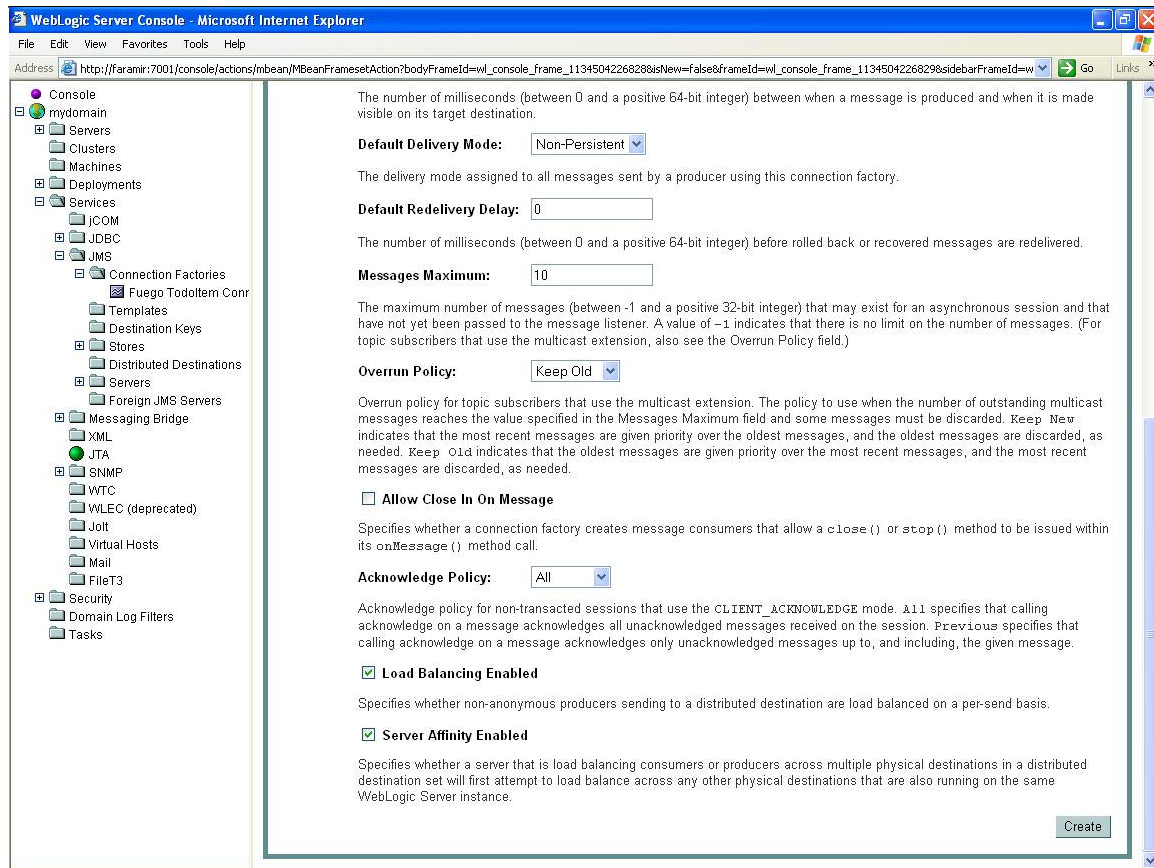
This panel will require BEA Administrator to configure at least 3 parameters that are listed below with a brief description for each one of these:

- **Name:** This is a logical name which should identify Fuego Server News Resource.
- **JNDI Name:** This field should contain the value assigned to the “JMS Topic connection factory” configuration parameter specified in Fuego Server “Application Server” Tab. In this case, the default proposed value is: “XATopicConnectionFactory”.
- **Default Delivery Mode:** Non-persistent

Fuego Enterprise 5.5 for Web Logic 8.1 Installation and Configuration Instructions



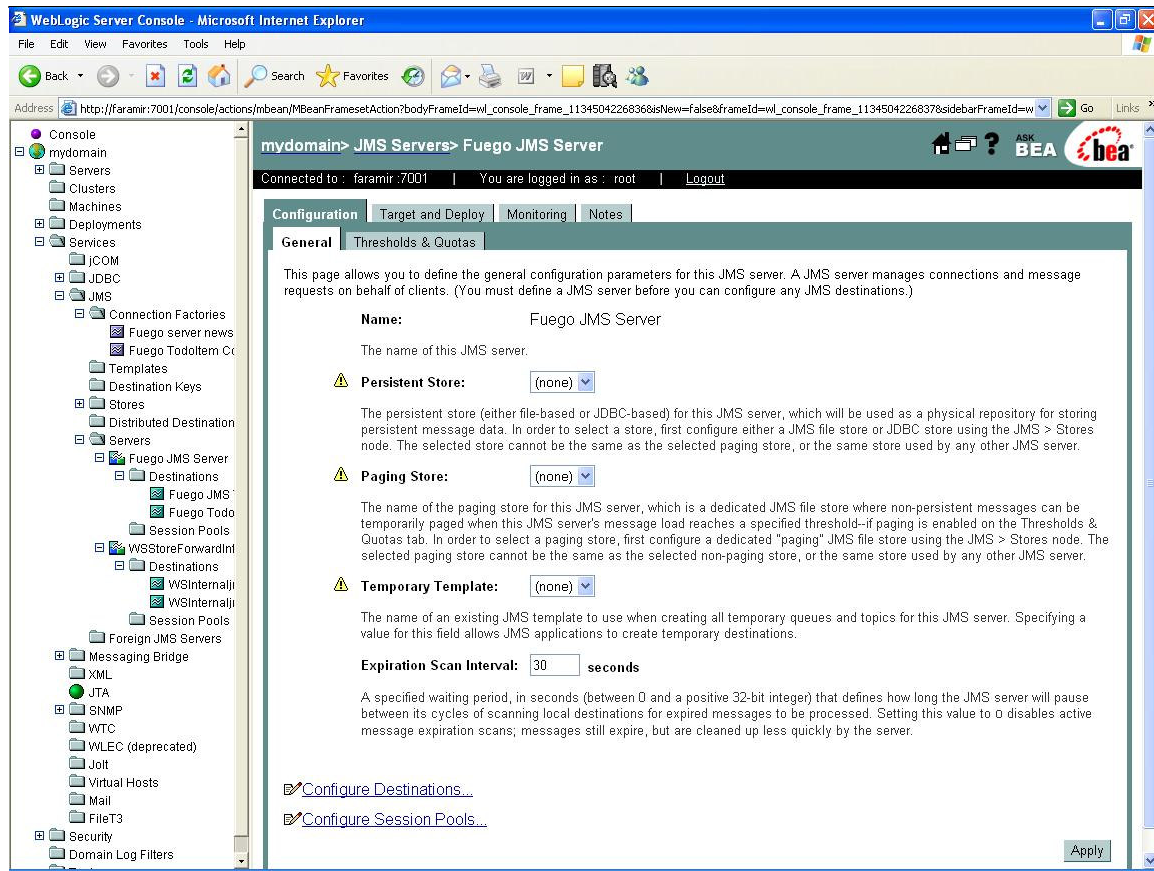
The rest of the fields can accept the default values. Press “Create” to add the new connection factory.



Creating JMS server (if necessary)

If BEA installation doesn't have a configured JMS server, it should be created as it will contain both ToDoItems Queue and Server News Topic.

To do that, go to the following left menu path: "Domain/Services/JMS/Servers". Select the option "Configure a new JMS Server...". In this example, we named it "Fuego JMS Server". The rest of the options can accept the default values.

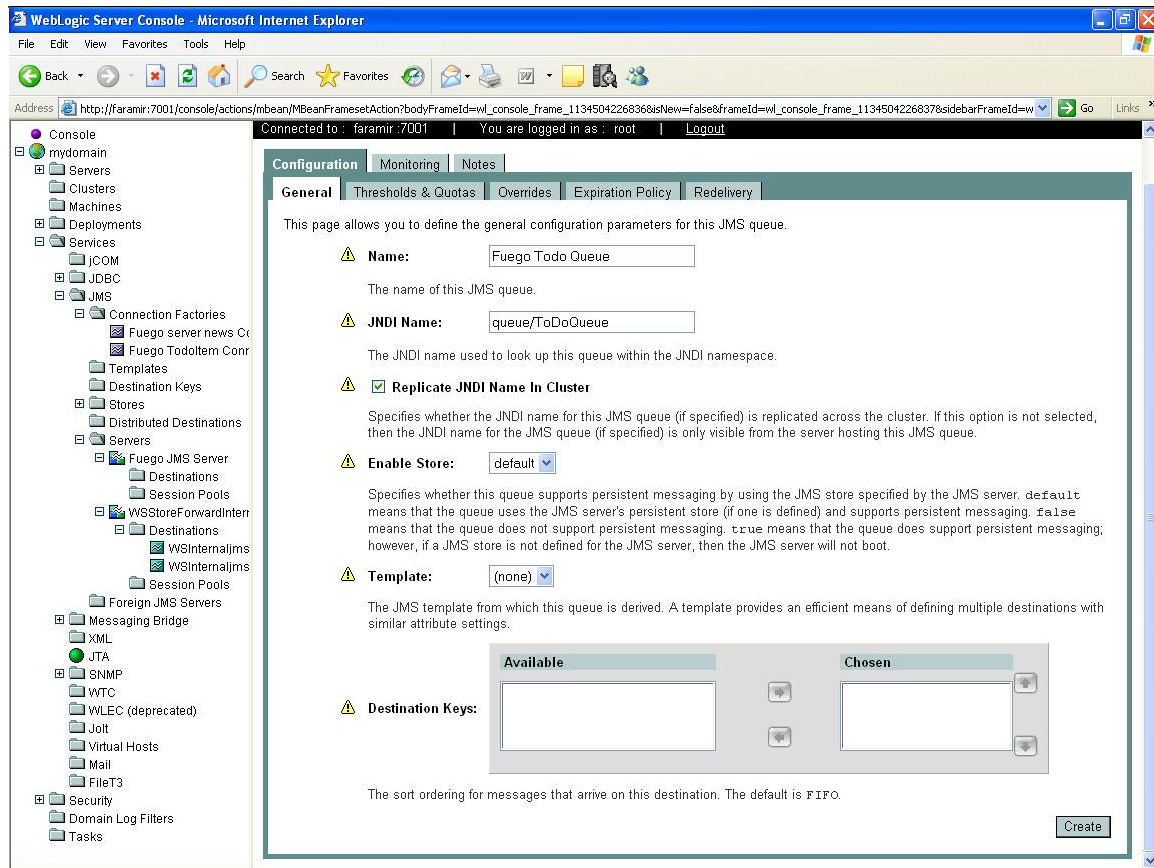


Creating Fuego ToDoItems Queue (Web Logic infrastructure)

After successfully creating the JMS server and Fuego Server ToDoItems JMS Connection Factory, BEA Administrator should create ToDoItems Queue. The Queue resource should be created under the JMS Destinations node.

BEA Destinations node may be found under the following left menu path: “Domain/Services/JMS/Servers/ Fuego JMS Server/Destinations”.

Right click on this node and select “Configure a new JMS Queue...” from the opened menu.



The dialog will request BEA Administrator to specify 2 mandatory fields. The rest of the fields may accept default values. The mandatory fields are:

- **Name:** This is a logical name used to identify the configuration of ToDoItems JMS Queue.
- **JNDI Name:** This field should contain the value specified for the configuration parameter “JMS Queue name” in Fuego Server “Application Server” Configuration Tab located in Fuego Web Console. The specified value was: “queue/ToDoQueue”.

Press “Create” button to add the queue.

Creating Fuego Server News Topic (Web Logic infrastructure)

BEA Administrator should also create a JMS Connection Factory for Fuego Server News Service.

BEA Destinations node may be found under the following left menu path: “Domain/Services/JMS/Servers/ Fuego JMS Server/Destinations”.

Right click on this node and select “Configure a new JMS Topic...” from the opened menu.

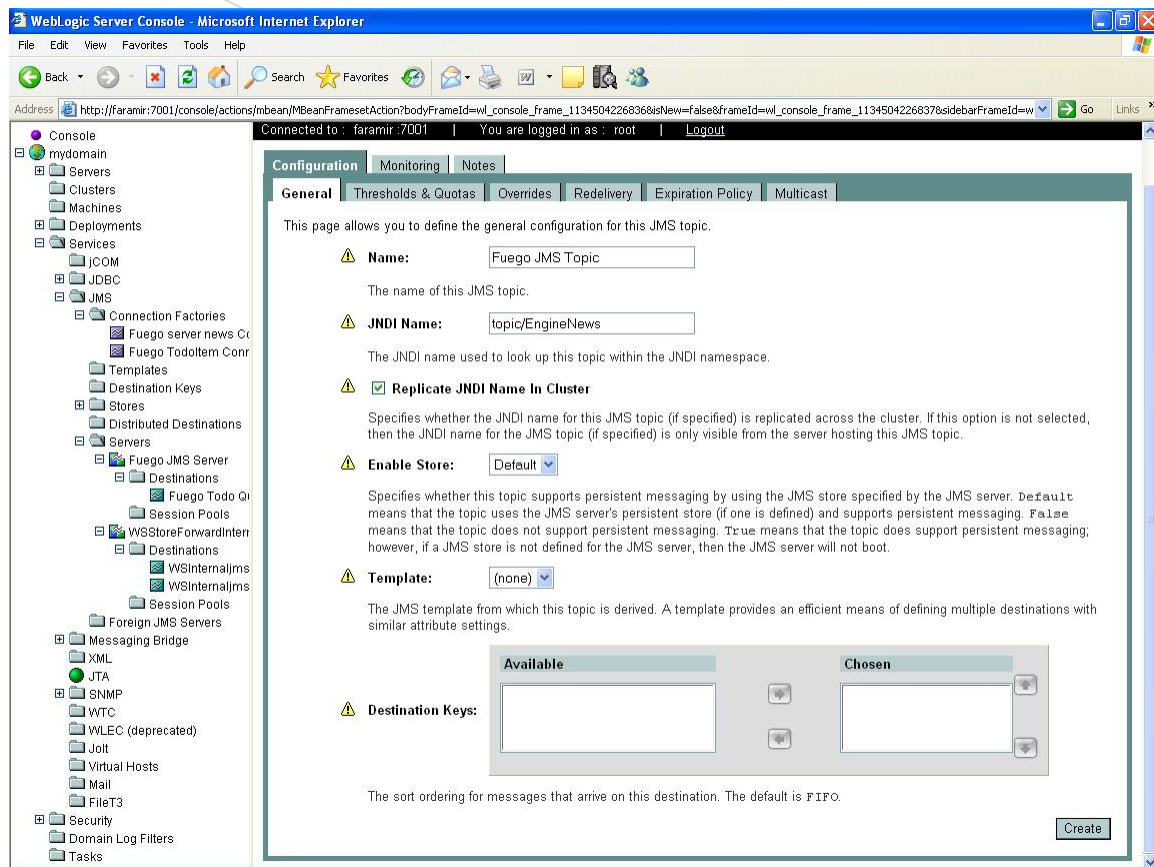
The figure below shows the JMS Connection Factory Configuration panel.

	2005 Fuego Inc. - Company Proprietary and Confidential.	
	All Rights Reserved.	Page 49 of 92

This panel will require BEA Administrator to configure at least 2 parameters that are listed below with a brief description for each one of these:

- **Name:** This is a logical name which should identify Fuego Server News Resource.
- **JNDI Name:** This field should contain the value assigned to the “JMS Topic connection factory” configuration parameter specified in Fuego Server “Application Server” Tab. In this case, the default proposed value is: “topic/EngineNews”.

The rest of the fields can accept the default values.



Topics and Queues for Fuego server (TIBCO EMS 4.1)

As it was mentioned before, BEA Administrator can use an external messaging infrastructure, such as TIBCO EMS. In that case, the following steps should be implemented.

We assume that BEA Administrator has already created and configured a JMS Foreign Provider for TIBCO. We will assume for this document that the TIBCO JMS Provider is named “TIBCO JMS Provider”.

This guide also assumes that the following resources have been created in the TIBCO EMS 4.1. The following is the list of commands to create the resources in TIBCO EMS.

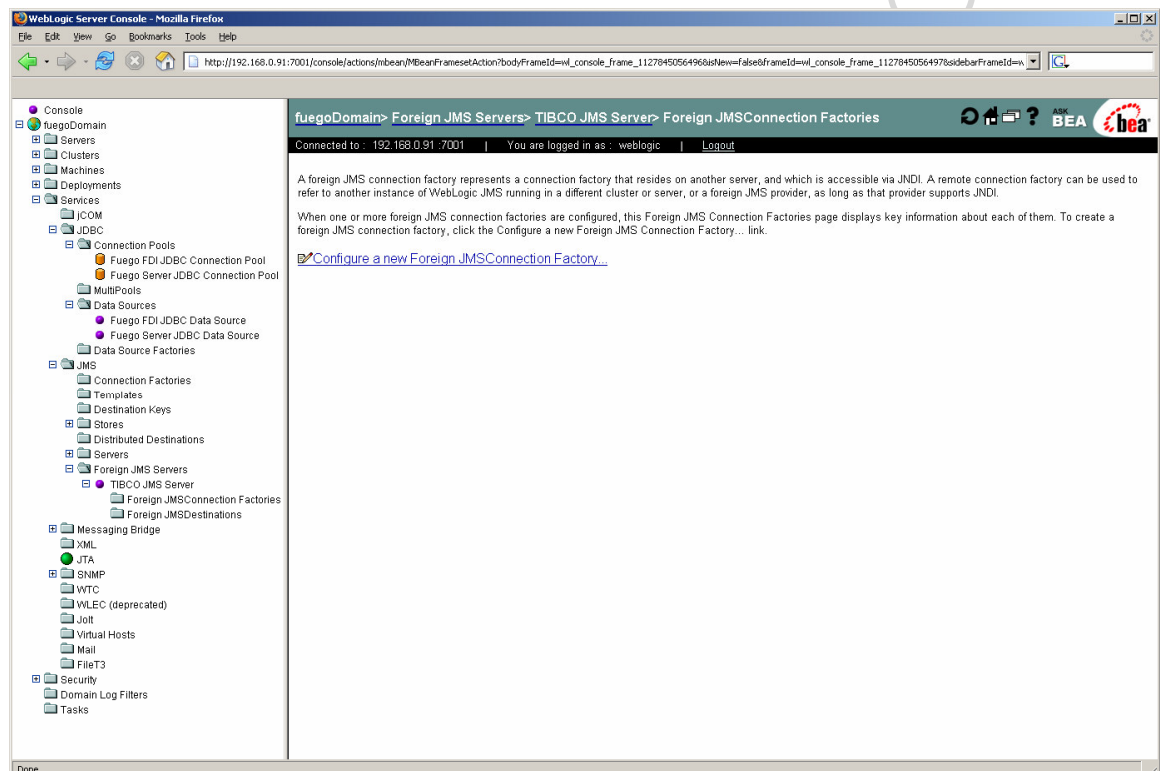
```

tcp://localhost:7222> create factory BEA_XAConnectionFactory
Error: Invalid create factory command
tcp://localhost:7222> create factory BEA_XAConnectionFactory xaqueue
url=tcp://192.168.0.91:7222
QueueConnectionFactory 'BEA_XAConnectionFactory' has been created
tcp://localhost:7222> create factory BEA_XATopicConnectionFactory xatopic
url=tcp://192.168.0.91:7222
TopicConnectionFactory 'BEA_XATopicConnectionFactory' has been created
tcp://localhost:7222> create queue BEA_ToDoQueue
Queue 'BEA_ToDoQueue' has been created
tcp://localhost:7222> create topic BEA_EngineNews
Topic 'BEA_EngineNews' has been created
tcp://localhost:7222> create jndiname BEA_ToDoQueue queue BEA_ToDoQueue
JNDI name 'BEA_ToDoQueue' has been created
tcp://localhost:7222> create jndiname BEA_EngineNews topic BEA_EngineNews
JNDI name 'BEA_EngineNews' has been created

```

Creating a Connection Factory for ToDoItems Queue (TIBCO EMS 4.1)

Navigate on the left menu panel through the following hierarchy “Domain/Services/JMS/Foreign JMS Servers/Foreign JMSConnection Factories”. Once in the “Connection Factories” node, right click and select “Configure a new Foreign JMSConnection Factory ...” from the opened menu. This is depicted in the figure below.



The first panel will require BEA Administrator to configure at least 3 parameters that are listed below with a brief description for each one of these:

- **Name:** This is a logical name which should identify Fuego Server ToDoItems Resource.
- **Local JNDI Name:** This field should contain the value assigned to the “JMS Queue connection factory” configuration parameter specified in Fuego Server “Application Server” Tab. In this case, the default proposed value is: “BEA_XAConnectionFactory”.
- **Remote JNDI Name:** This field should contain the name of the ToDoQueue JNDI Connection Factory defined in TIBCO EMS In this case, the default proposed value is: “BEA_XAConnectionFactory”.

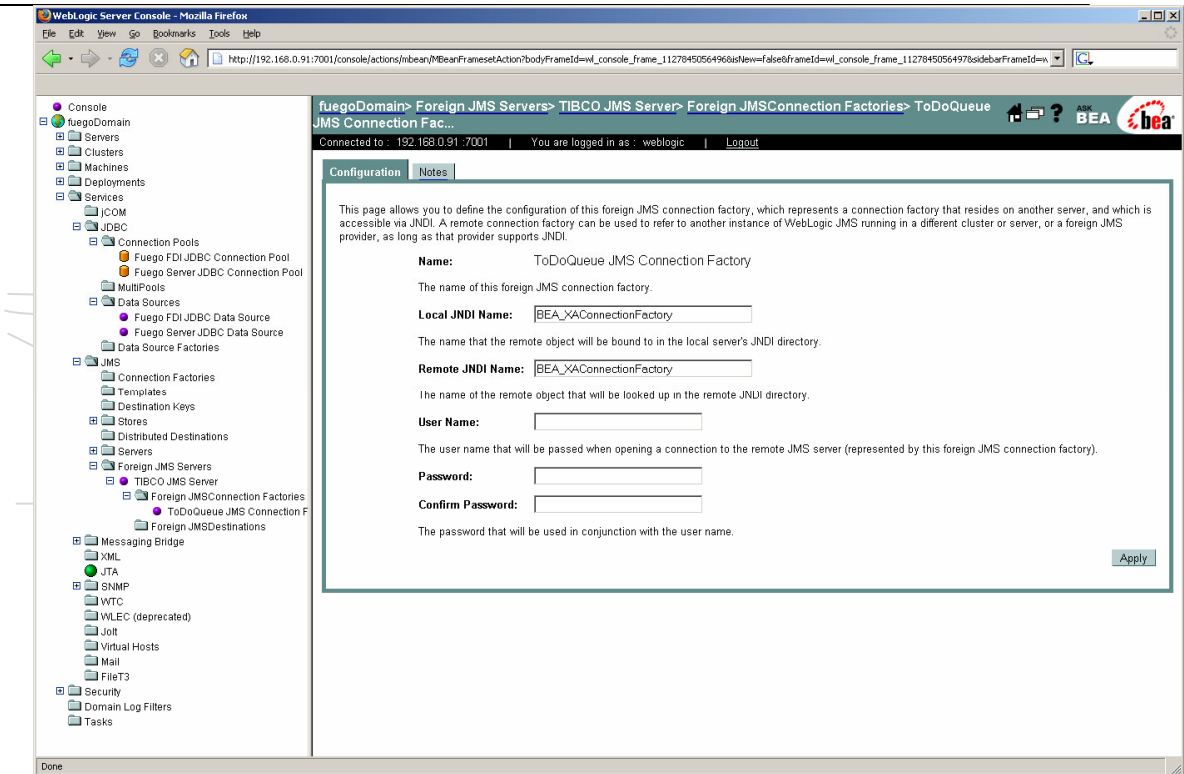
The rest of the fields can accept the default values.

The screenshot shows the WebLogic Server Console in Mozilla Firefox. The left sidebar displays a tree view of the console structure, with 'Foreign JMS Servers' selected. The main panel shows the 'Configuration' tab for 'TIBCO JMS Server' with the following fields:

- Name:** ToDoQueue JMS Connection Factory
- Local JNDI Name:** BEA_XAConnectionFactory
- Remote JNDI Name:** BEA_XAConnectionFactory
- User Name:** (empty)
- Password:** (empty)
- Confirm Password:** (empty)

The 'Create' button is located at the bottom right of the form.

Click on the “Create” button at the end of the form.



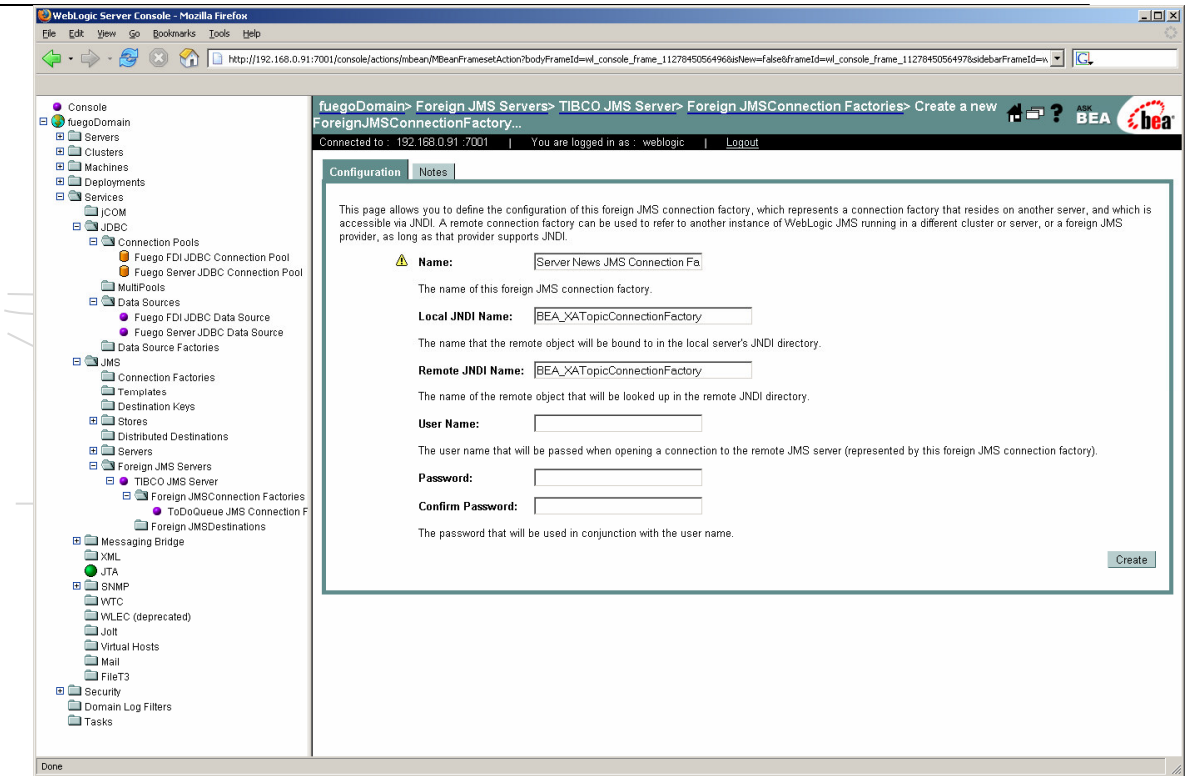
Creating a Connection Factory for Fuego Server News Topic (TIBCO EMS 4.1)

BEA Administrator should also create a JMS Connection Factory for Fuego Server News Service. The figure below shows the JMS Connection Factory Configuration panel.

This panel will require BEA Administrator to configure at least 3 parameters that are listed below with a brief description for each one of these:

- **Name:** This is a logical name which should identify Fuego Server News Resource.
- **JNDI Name:** This field should contain the value assigned to the “JMS Topic connection factory” configuration parameter specified in Fuego Server “Application Server” Tab. In this case, the default proposed value is: “BEA_XATopicConnectionFactory”.
- **Remote JNDI Name:** This field should contain the name of the Server News JNDI Connection Factory defined in TIBCO EMS In this case, the default proposed value is: “BEA_XATopicConnectionFactory”.

The rest of the fields can accept the default values.

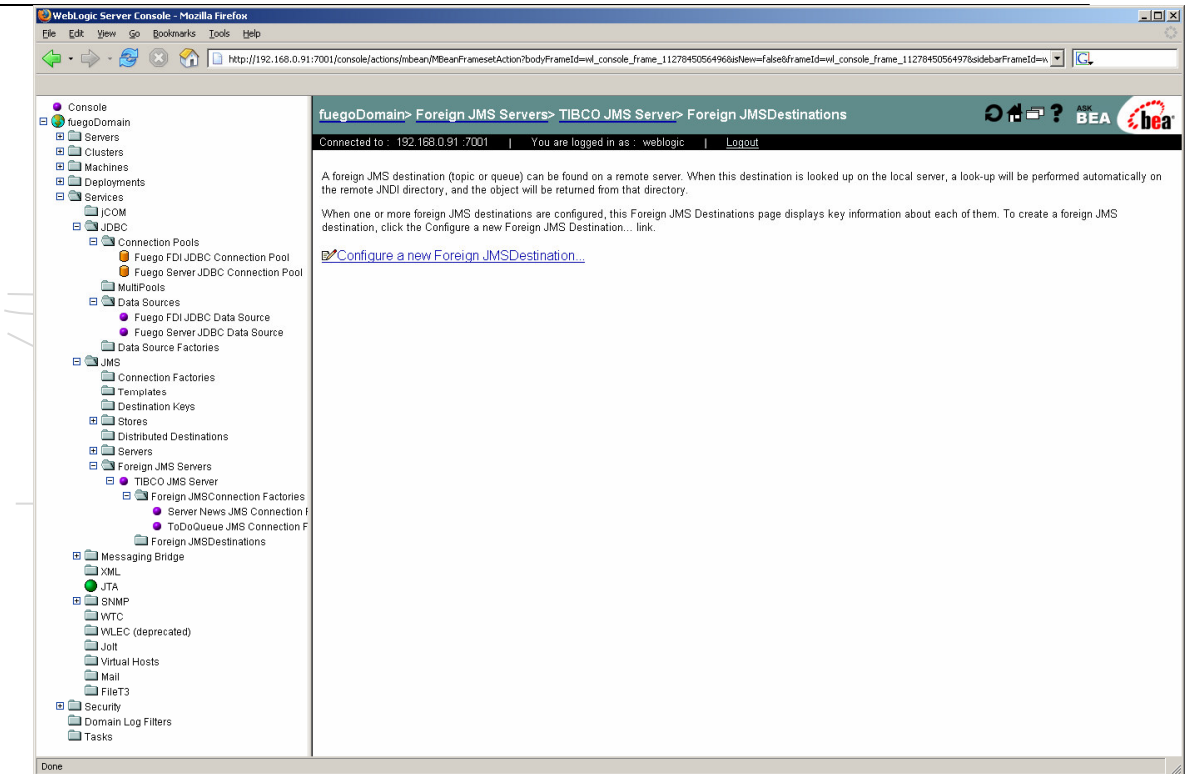


Click on the “Create” button to proceed.

Creating Fuego ToDoItems Queue (TIBCO EMS 4.1)

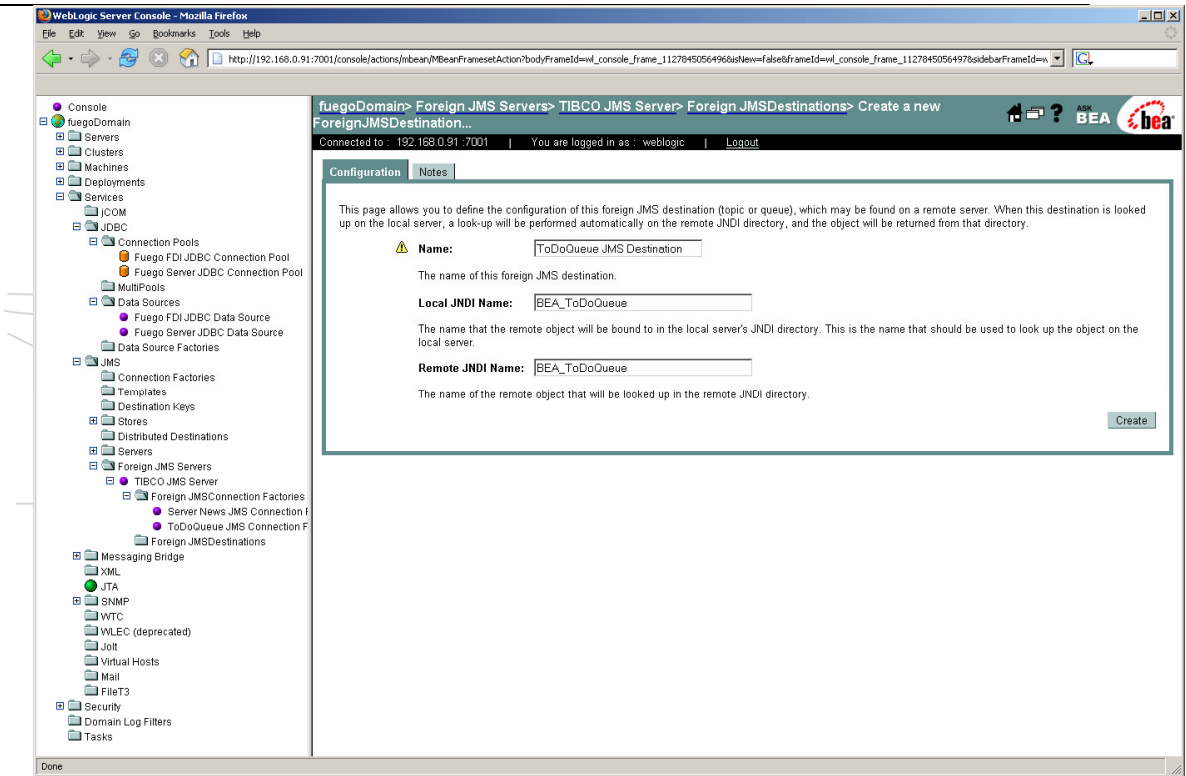
After successfully creating Fuego Server ToDoItems JMS Connection Factory, BEA Administrator should create ToDoItems Queue. The Queue resource should be created under the JMS Destinations node. By default, BEA Destinations node may be found under the following left menu path: “Domain/Services/JMS/Foreign JMS Server/TIBCO JMS Server/Foreign JMSDestinations.”

Right click on this node and select “Configure a new Foreign JMSDestination ...” from the opened menu.



The following dialog will request BEA Administrator to specify 3 mandatory fields. The rest of the fields may accept default values. The mandatory fields are:

- **Name:** This is a logical name used to identify the configuration of ToDoItems JMS Queue.
- **Local JNDI Name:** This field should contain the value specified for the configuration parameter “JMS Queue name” in Fuego Server “Application Server” Configuration Tab located in Fuego Web Console. The specified value was: “BEA_ToDoQueue”.
- **Remote JNDI Name:** This field should the JNDI Name given in TIBCO EMS to the ToDoQueue Queue. The specified value was: “BEA_ToDoQueue”.

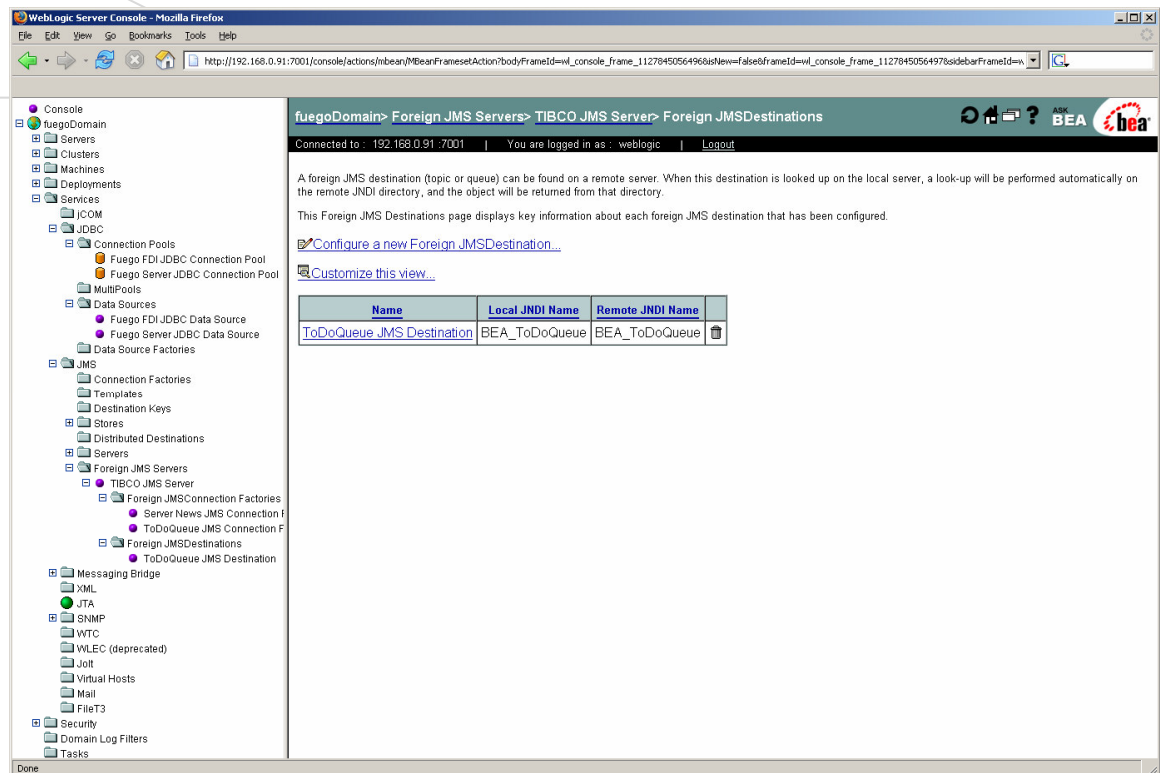


Once these values have been specified, click on the “Create” button to finish the resource creation.

Creating Fuego Server News Topic (TIBCO EMS 4.1)

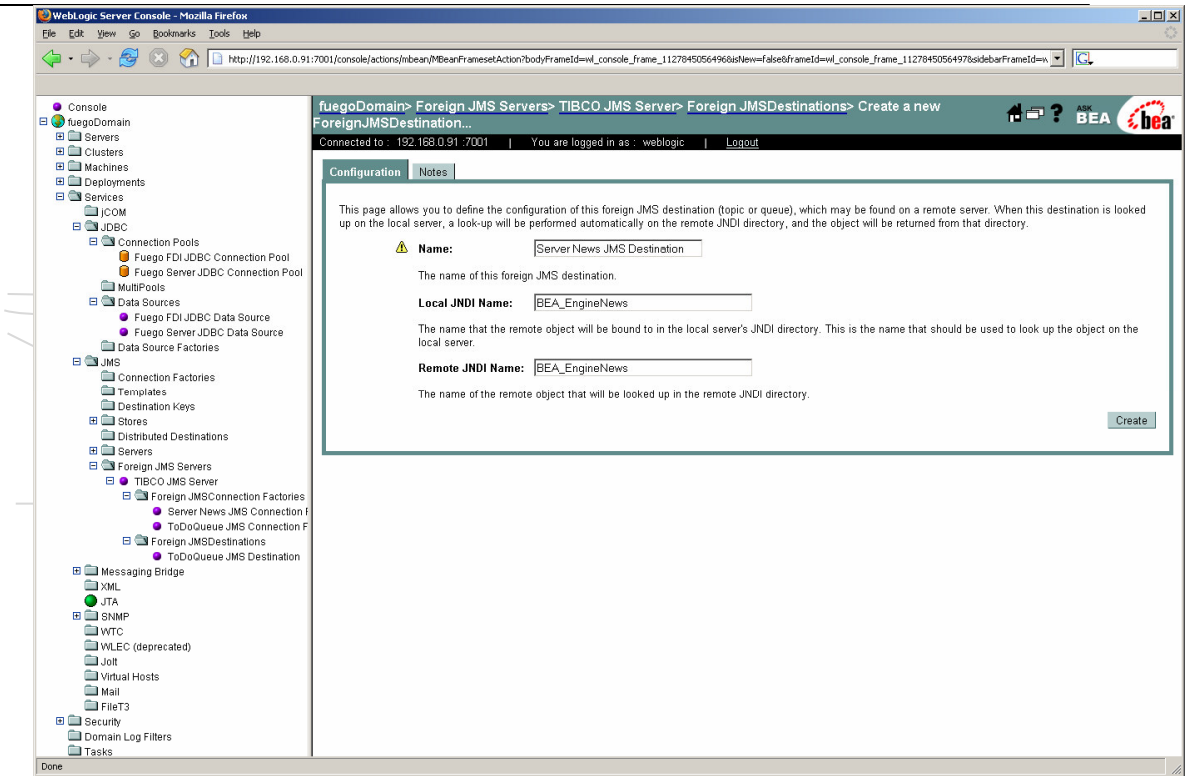
After successfully creating Fuego Server News JMS Connection Factory, BEA Administrator should create News Topic. The Queue resource should be created under the JMS Destinations node. By default, BEA Destinations node may be found under the following left menu path: “Domain/Services/JMS/Foreign JMS Server/TIBCO JMS Server/Foreign JMSDestinations.”

Right click on this node and select “Configure a new Foreign JMSDestination ...” from the opened menu.



The following dialog will request BEA Administrator to specify 2 mandatory fields. The rest of the fields may accept default values. The mandatory fields are:

- **Name:** This is a logical name used to identify the configuration of Server News' JMS Topic.
- **Local JNDI Name:** This field should contain the value specified for the configuration parameter “JMS Topic name” in Fuego Server “Application Server” Configuration Tab located in Fuego Web Console. The specified value was: “BEA_EngineNews”.
- **Remote JNDI Name:** This field should contain the JNDI Name given in TIBCO EMS to the Server News Topic. The specified value was: “BEA_EngineNews”.



Once these values have been specified, click on the “Create” button to finish the resource creation.

Deploying Fuego J2EE Deployer, Fuego Server and Fuego Work Portal

Now that the J2EE resources have been created in WebLogic Application Server, Fuego Administrator should provide the EAR (Enterprise Application Files) files to the WebLogic Administrator for deployment.

Fuego provides the J2EE Deployer application, which can be used by Fuego Administrator to deploy server and project applications from Fuego Web Console. That way, manual deployment in WebLogic can be avoided for those applications.

Fuego J2EE Deployer will also simplify the administration (start, stop, uninstall, etc) of server and project applications, by allowing Fuego Administrator to execute those tasks directly from Fuego Web Console.

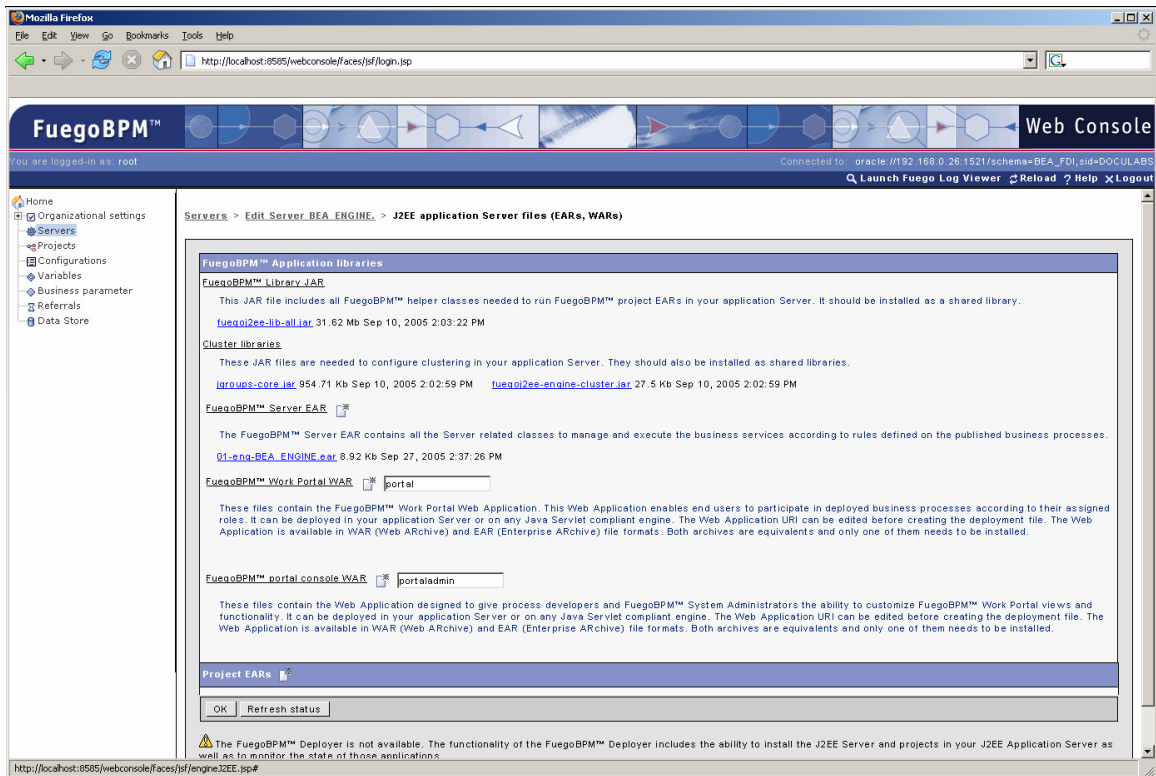
Deploying Fuego Library JAR

Fuego Library JAR needs to be made available in BEA Application Server and have this JAR file available in the Application Server CLASSPATH. Check BEA reference documentation as to how to install the JAR file to be included in BEA CLASSPATH. You can include the reference to this jar in the startWebLogic.cmd (startWebLogic.sh) or startManagedWebLogic.cmd (startManagedWebLogic.sh) start up scripts.

In addition, you may need to include the “jgroups-core.jar” and “fuegoj2ee-engine-cluster.jar” JARs in these scripts as well if you are deploying a cluster configuration. These files can be found inside \$FUEGO\$/j2ee/common/lib/cluster directory or can also be download from Fuego Web Console.

To obtain Fuego Library JAR, Fuego Administrator should click on the link labeled “fuegoj2ee-lib-all.jar” (inside the server ‘J2EE application server files’ page from web console). After downloading the JAR file, provide it to BEA Administrator.

The following picture shows the screen in which the library jar can be obtained.



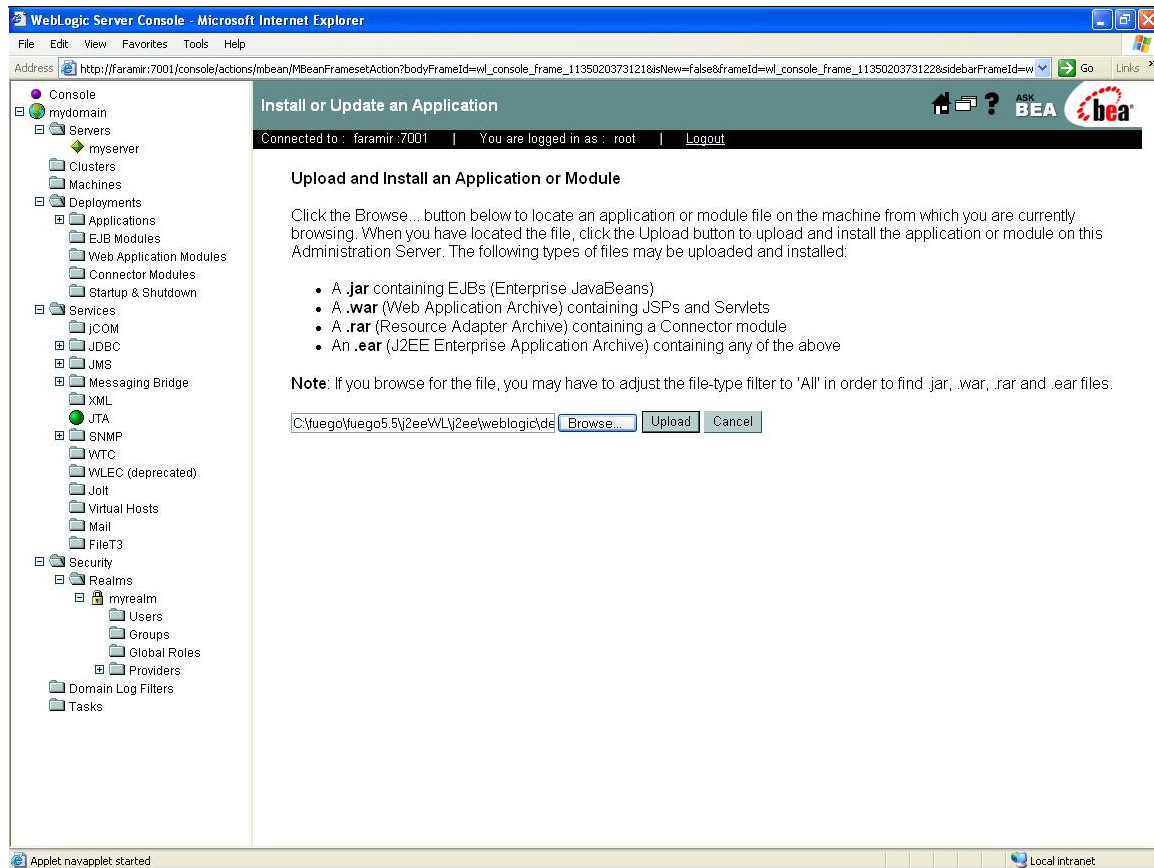
Deploying Fuego J2EE Deployer

Fuego Administrator should provide the WebLogic J2EE Deployer EAR file `wlj2eedeployer.ear` (located under `$FUEGO_J2EE_ENTERPRISE/j2ee/weblogic/deployer`).

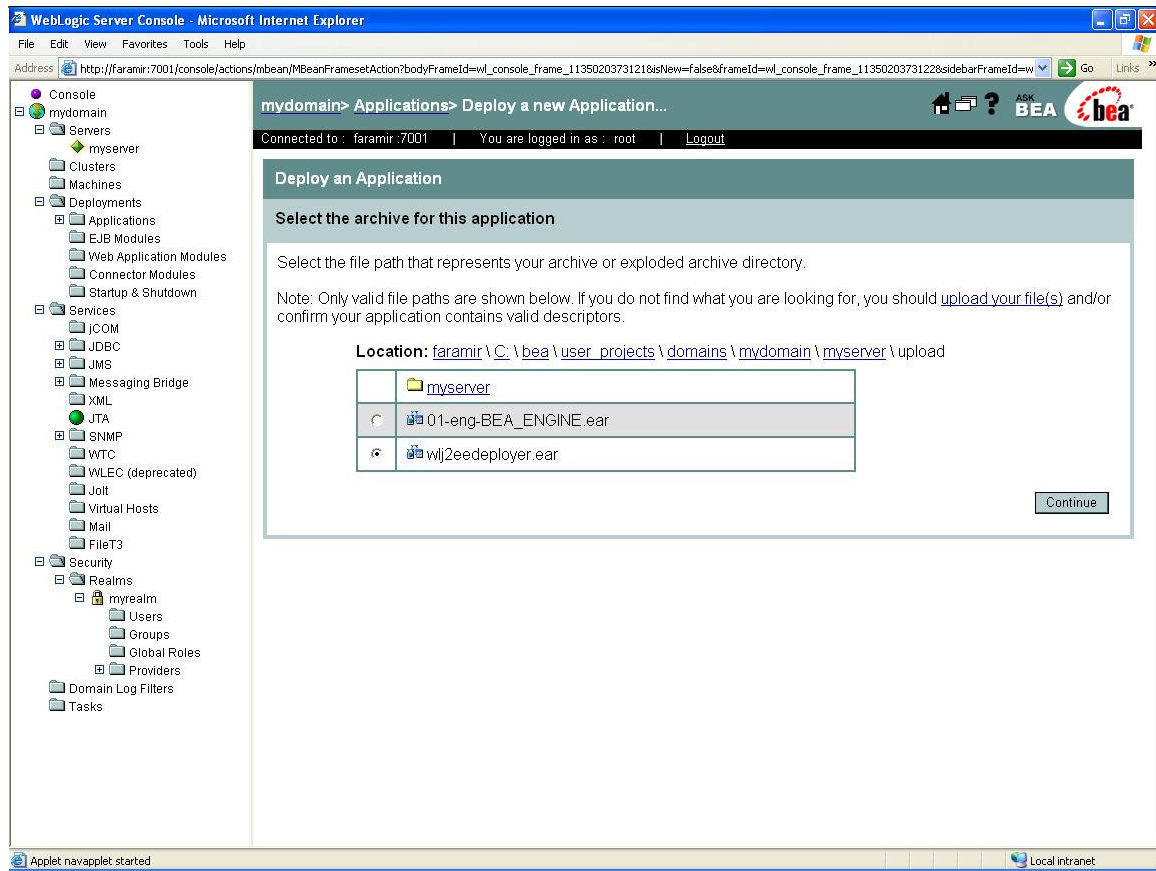
Before installing the ear file, the user "FuegoWebLogicDeployer" must be created in the Weblogic Application Server and it must have administrator permission (typically the user must participate in the "Administrators" group, that has the "Admin" role assigned).

WebLogic Administrator should then deploy this EAR through the following page (that can be found in the WebLogic Console navigational tree path: `$DOMAIN/Deployments/Applications`). After clicking on the "Deploy a new Application", BEA Administrator should complete the panel shown below.

Fuego Enterprise 5.5 for Web Logic 8.1 Installation and Configuration Instructions



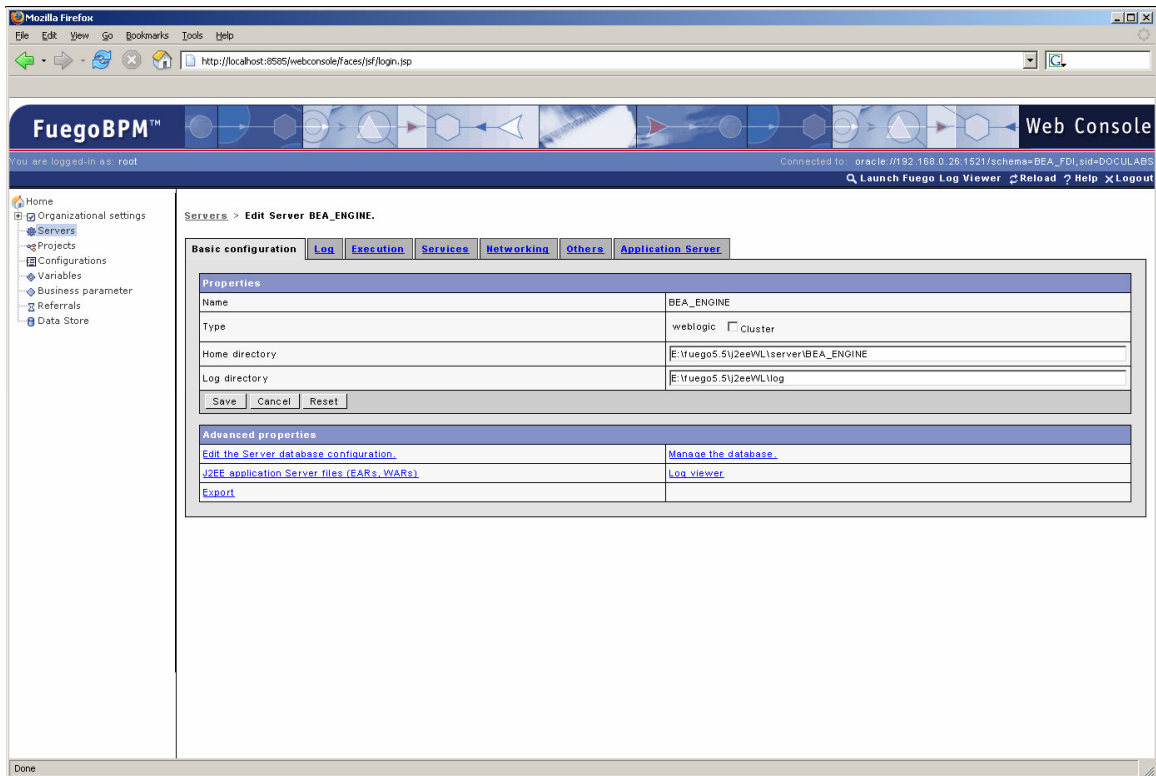
WebLogic Administrator should browse and upload the ear file that was previously generated.



After that, the recently uploaded ear should be selected to continue installation. Default options can be selected in the rest of the procedure.

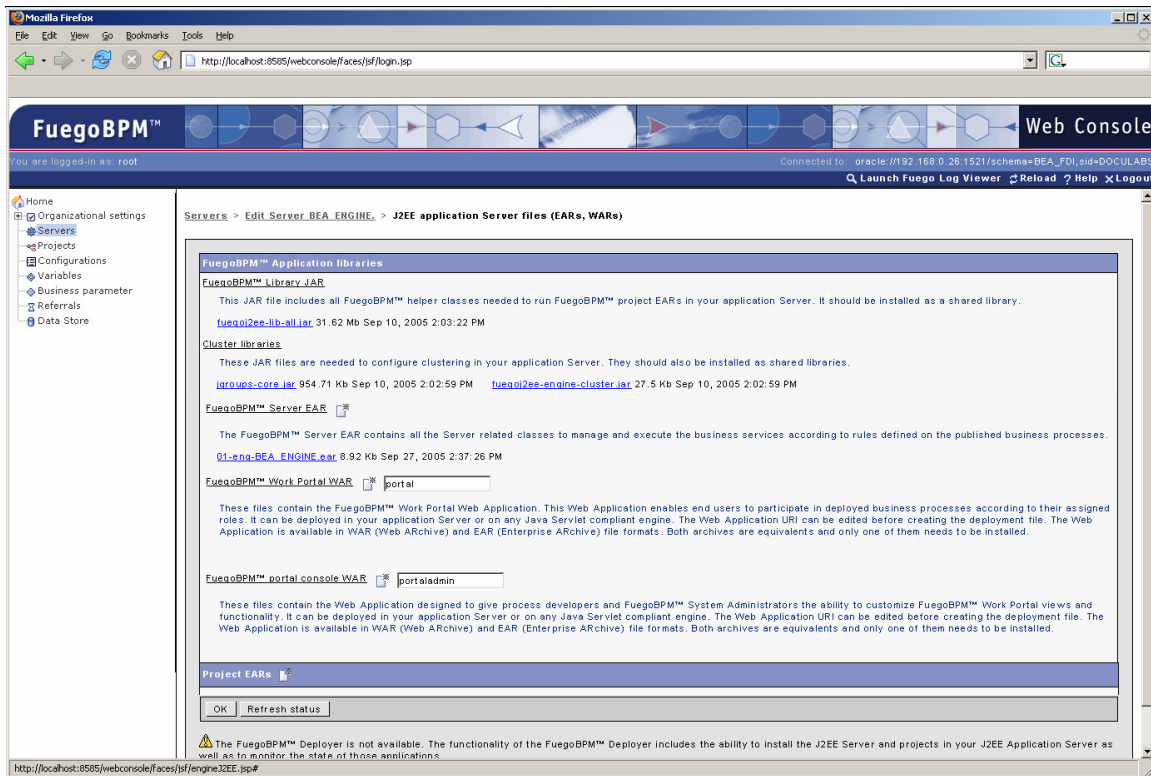
Deploying Fuego Server

Fuego Administrator should log into Fuego Web Console and click on the “Servers” node on the left panel and then click on the “Fuego Server Name” on the right area of the browser. The Fuego Server Configuration panel should look similar to the one shown below.



Once in the Fuego Server “Basic Configuration” Tab, click on the “J2EE application server files (EARS, WARs)” link at the bottom of the panel.

A new panel will provide information as well as resources to be deployed on BEA Application Server. The figure below shows what this panel should look like.



If Fuego Deployer has been already installed, Fuego Administrator can add the Server application directly from Fuego web console.

From 'J2EE application Server files' page, first create the ear file by pressing the creation icon, and then click the 'Install Server ear in the Application Server' button that is placed at the right of the server ear link.

After a successful installation, the server ear administration area should look like this:



In that area you will find buttons to start and stop server, uninstall o re-install it.

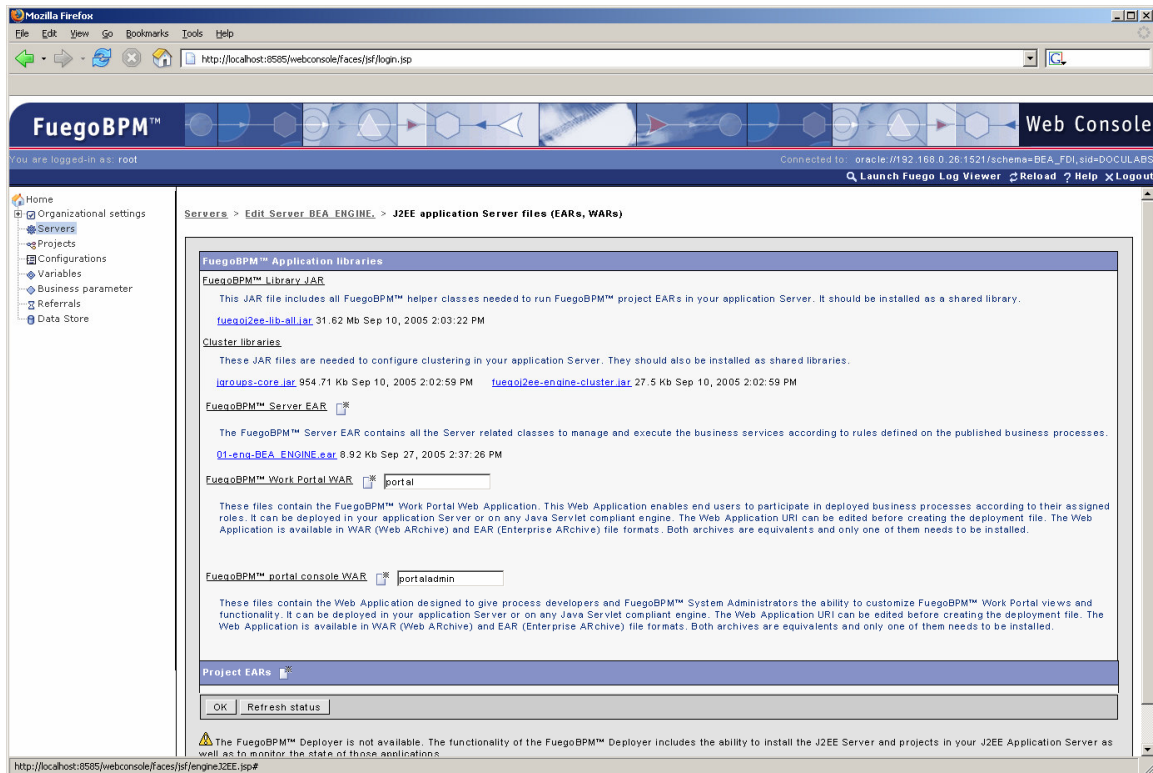
WebLogic Administrator may want to install Fuego Server without the Fuego Deployer application. To proceed with the manual Fuego Server installation, Fuego Administrator should provide the following resources to BEA Administrator to be deployed:

- Fuego Library JAR (that has been already mentioned)
- Cluster libraries (if installing on a clustered BEA Web Logic infrastructure. They have also been mentioned)
- Server EAR

To obtain Fuego Server EAR, click on the icon next to the “FuegoBPM Server EAR” section. After clicking on the icon, Fuego Web Console will create or re-create the

	2005 Fuego Inc. - Company Proprietary and Confidential.	
	All Rights Reserved.	Page 64 of 92

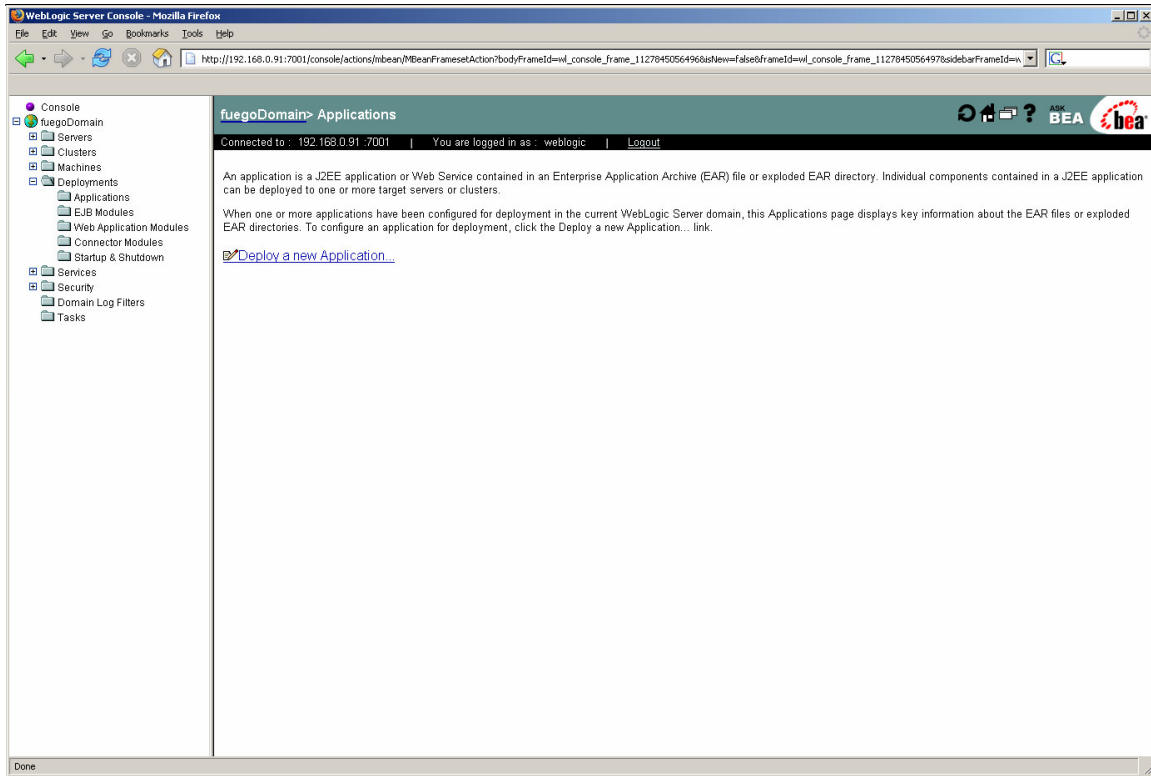
Fuego Server EAR for deployment in BEA Application Server. A link to the recently created Fuego Server EAR will be provided in this same panel as shown below. Once the link is enabled, download the EAR file and provide it to BEA Administrator.



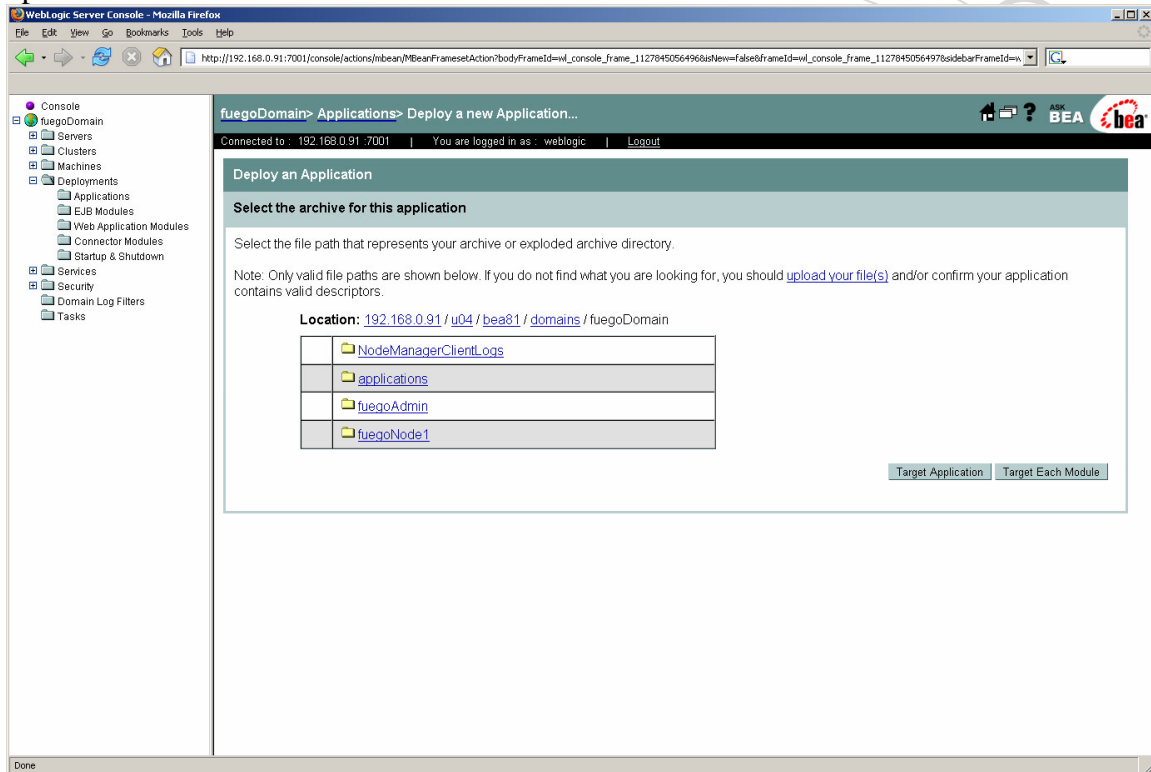
Deploying Server EAR

If WebLogic Administrator chooses to manually install Fuego server ear, he can use the following steps as a basic guide.

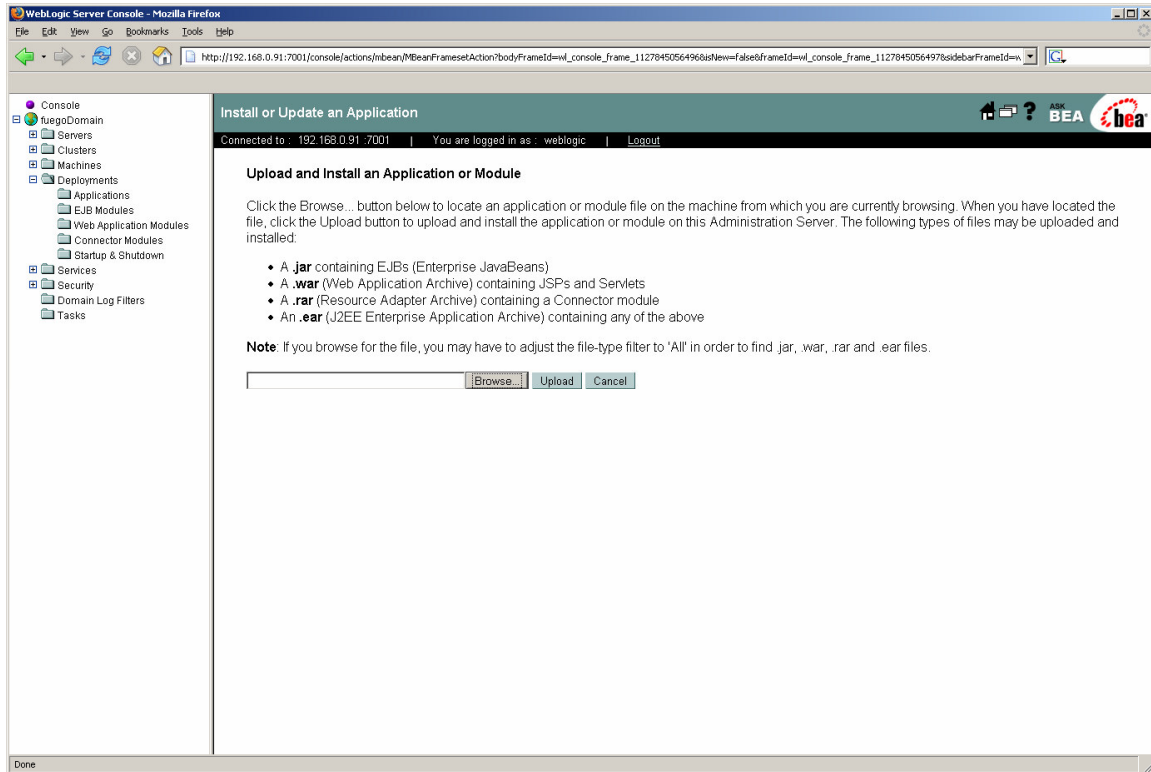
First, log into WebLogic Console and click on “Domain/Deployments/Applications” menu item. The following screen depicts this action.



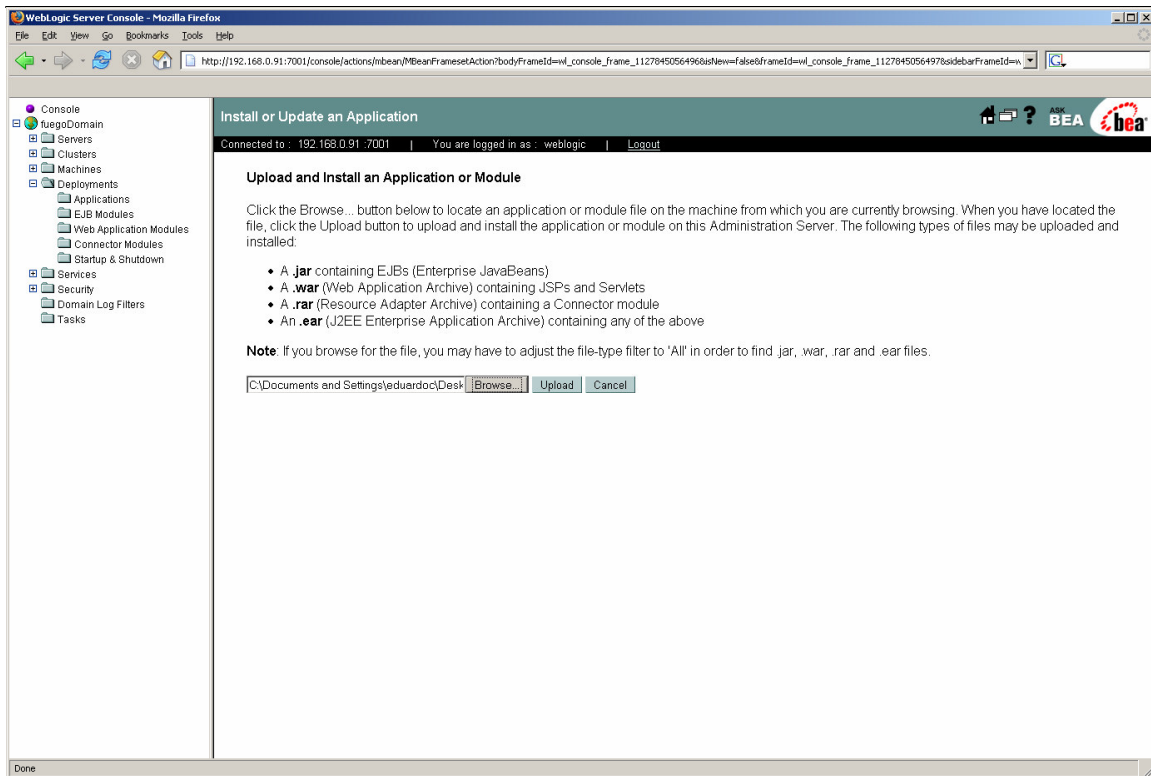
Click on the “Deploy a new Application” link. BEA Administrator should then upload the Server EAR file.



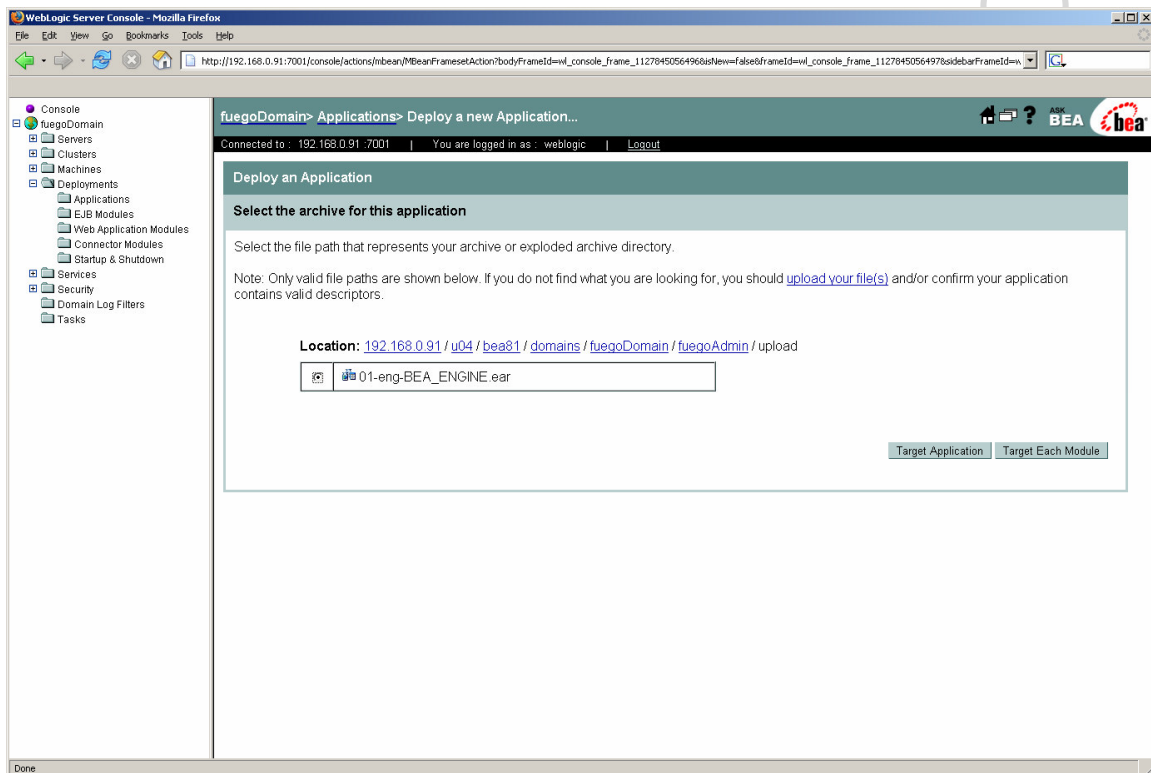
After clicking the link, the following panel will be presented and BEA Administrator should click on the “Browse” button to select Fuego Server EAR.



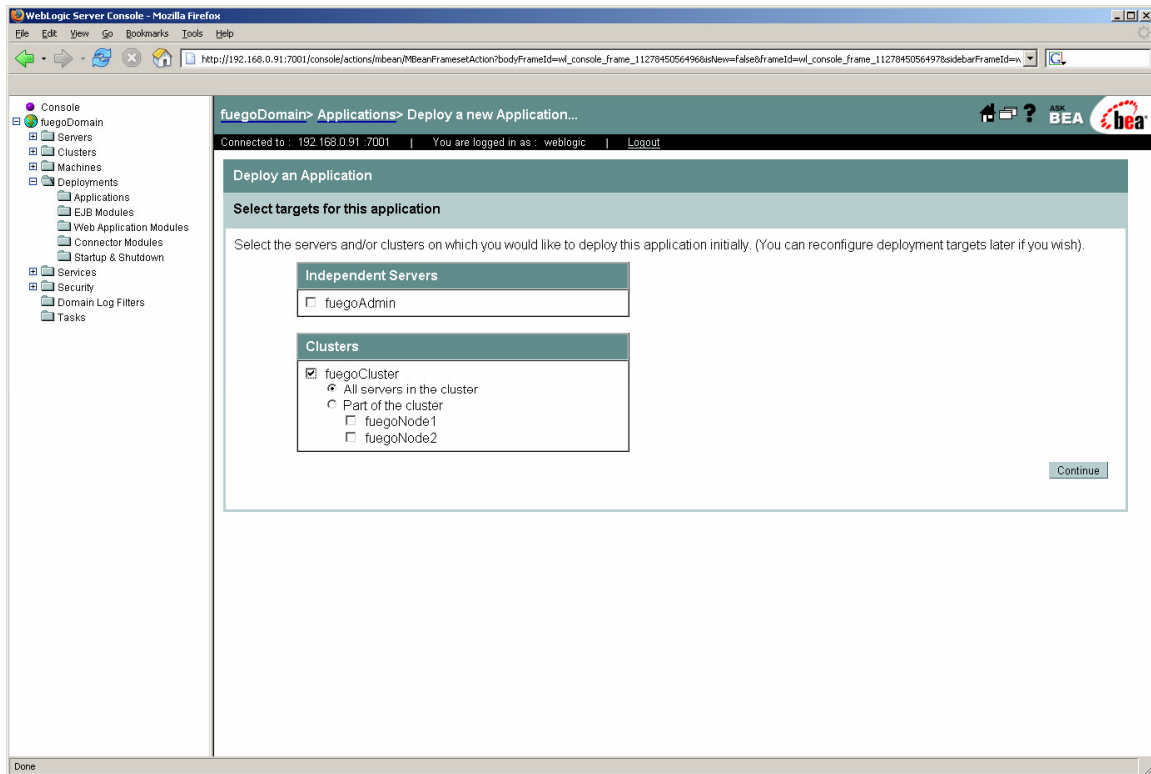
Once the EAR file has been selected, click on the “Upload” button to proceed.



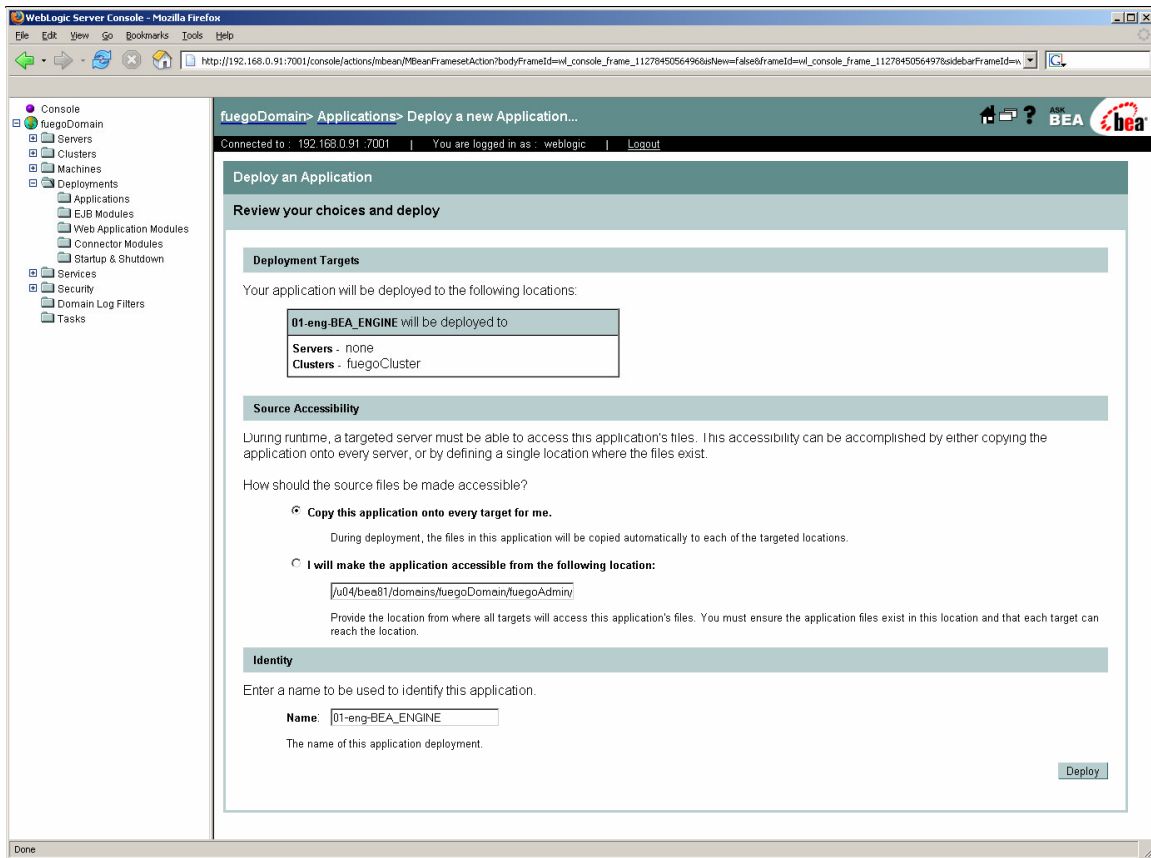
After the EAR file is uploaded, select the EAR file and click on the “Continue” button.



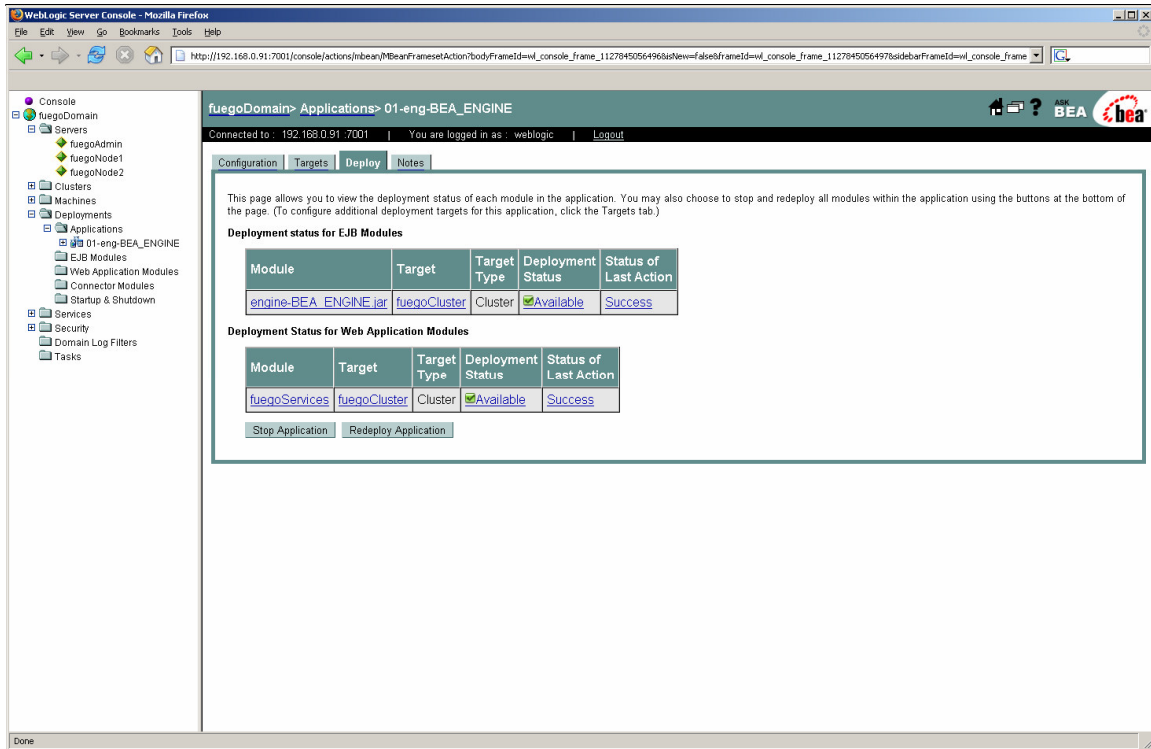
A new screen appears when you continue with the deployment.



In this new panel, BEA Administrator should define the scope for the Fuego Engine deployment. In our case, we will proceed with the cluster scope deployment. Click “Continue” to proceed with the deployment.



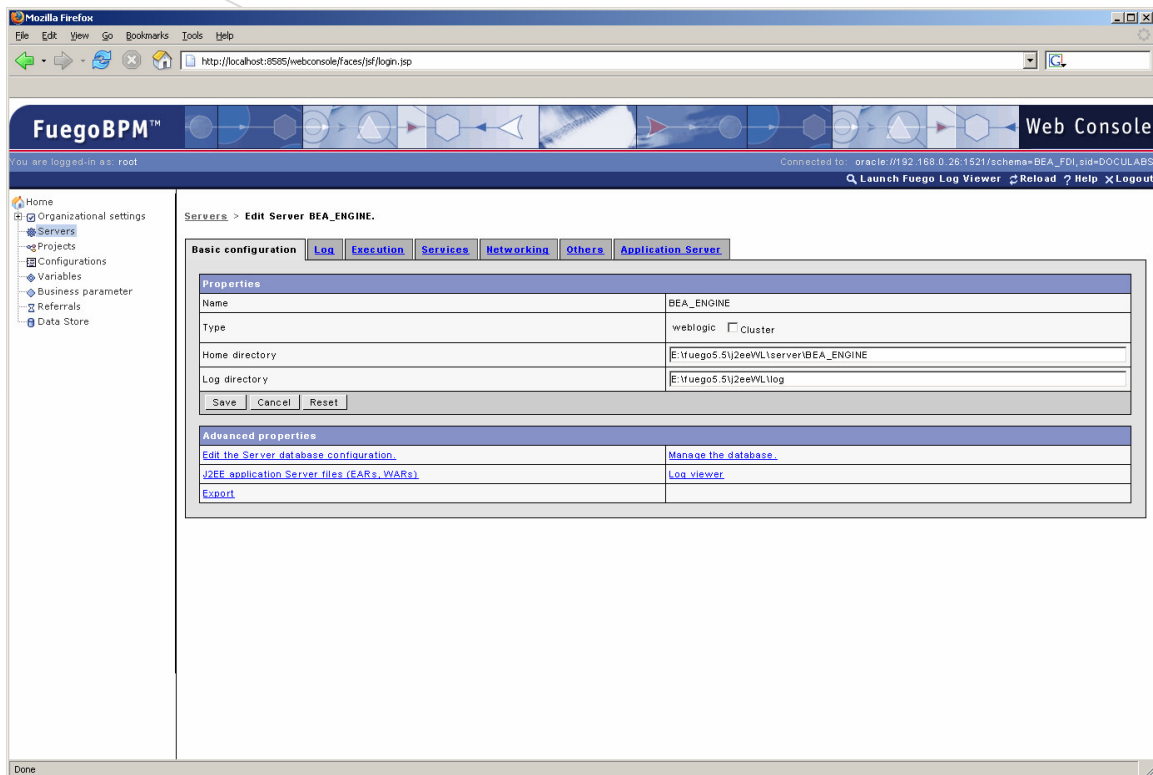
In this new panel, click “Deploy” to start the Fuego Engine EAR deployment. Upon successful Fuego Server EAR deployment, the status information about the EAR resources should be presented. It should look similar to the one provided in the figure below.

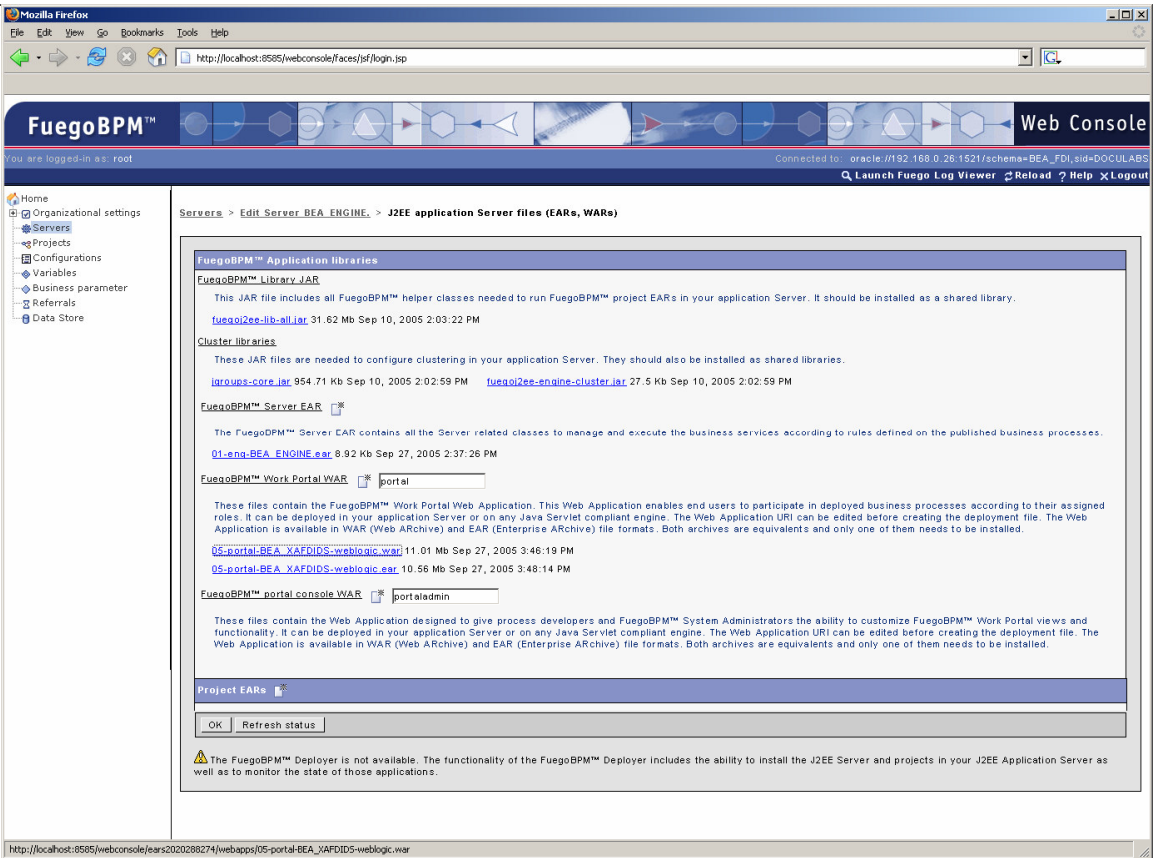


Deploying Fuego Portal

Fuego Administrator should first generate the Portal WAR file from the Fuego Web Console Panel. Once the WAR file has been generated, Fuego Administrator should download the WAR and give it to WebLogic Administrator for its deployment using WebLogic Console.

To obtain the WAR file, Fuego Administrator should open Fuego Web Console and open “J2EE application server files (EARs, WARs)” page, placed inside the server properties link.

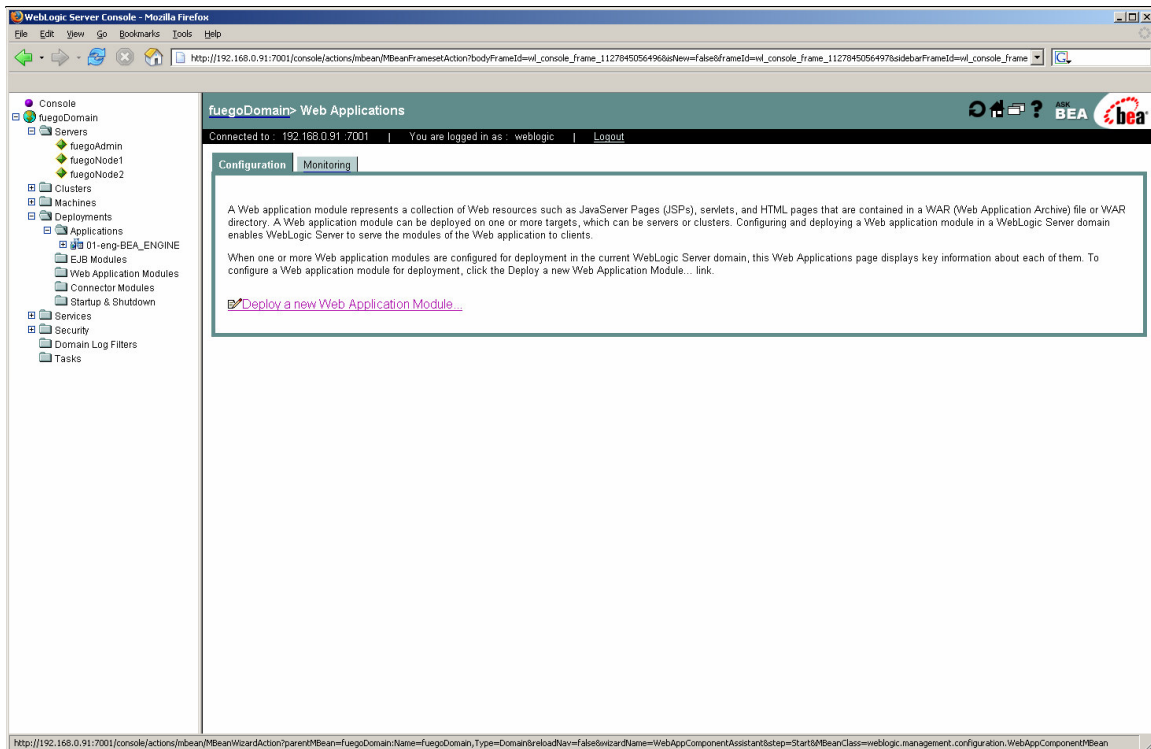




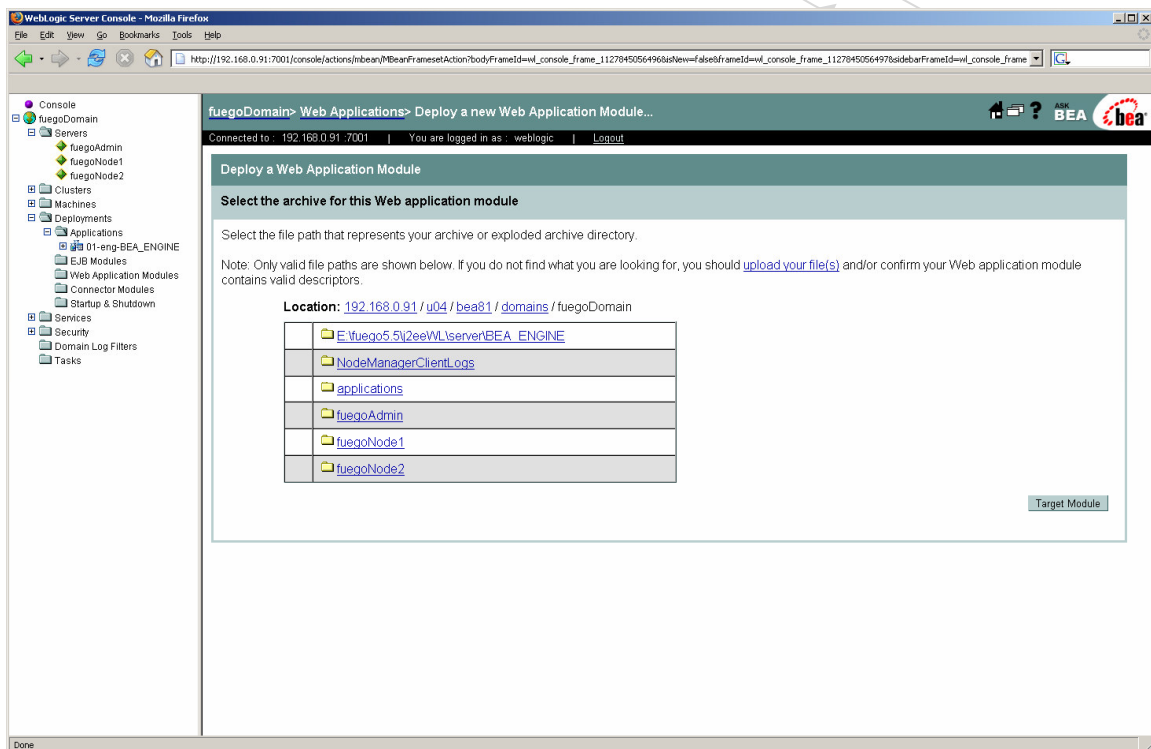
There, Fuego Administrator will be able to generate (or regenerate) the portal war file. After war has been created or updated, it should be sent to WebLogic Administrator, who will install it as a new Web Application.

To do that, from WebLogic console, the administrator must go to the page “\$Domain\$/Deployments/Web Applications Modules”.

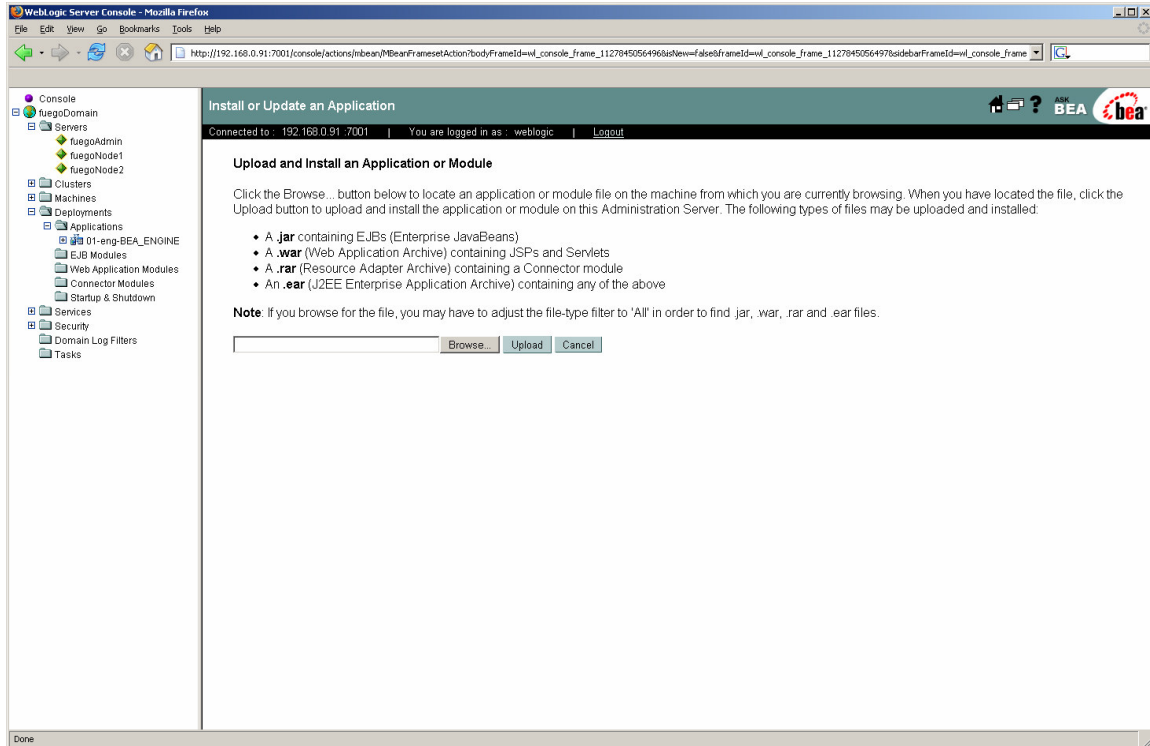
Fuego Enterprise 5.5 for Web Logic 8.1 Installation and Configuration Instructions



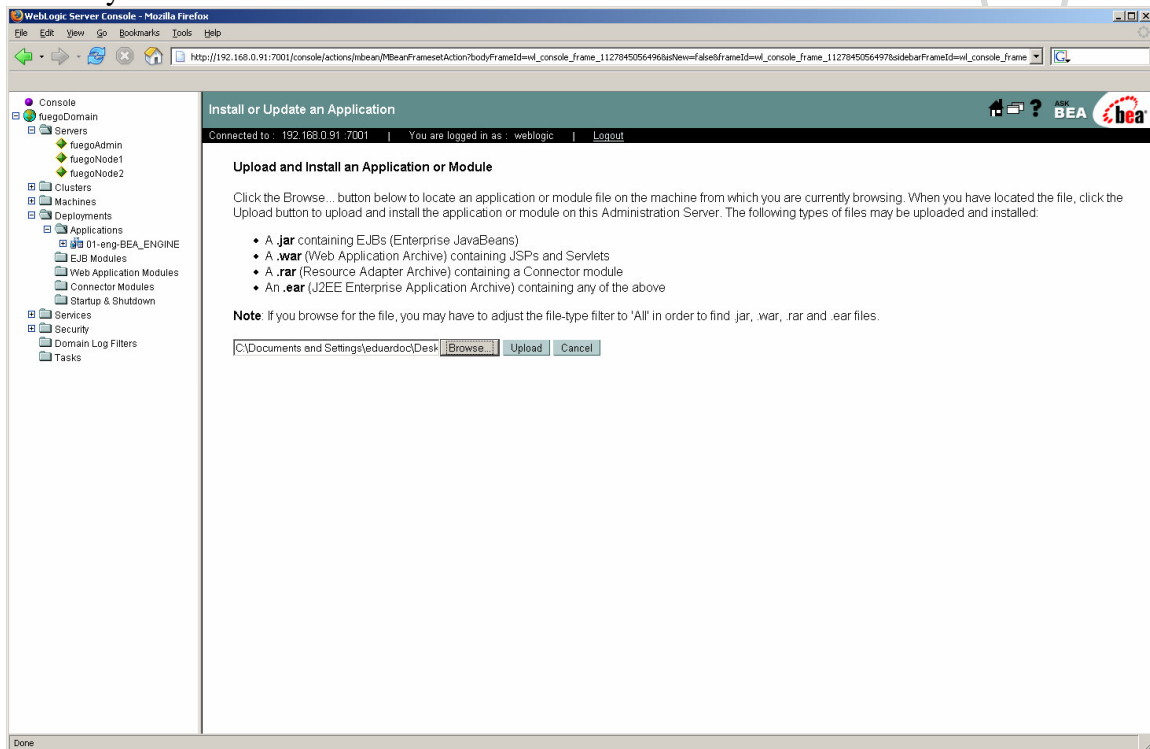
Select the option “Deploy a new Web Application Module...”.



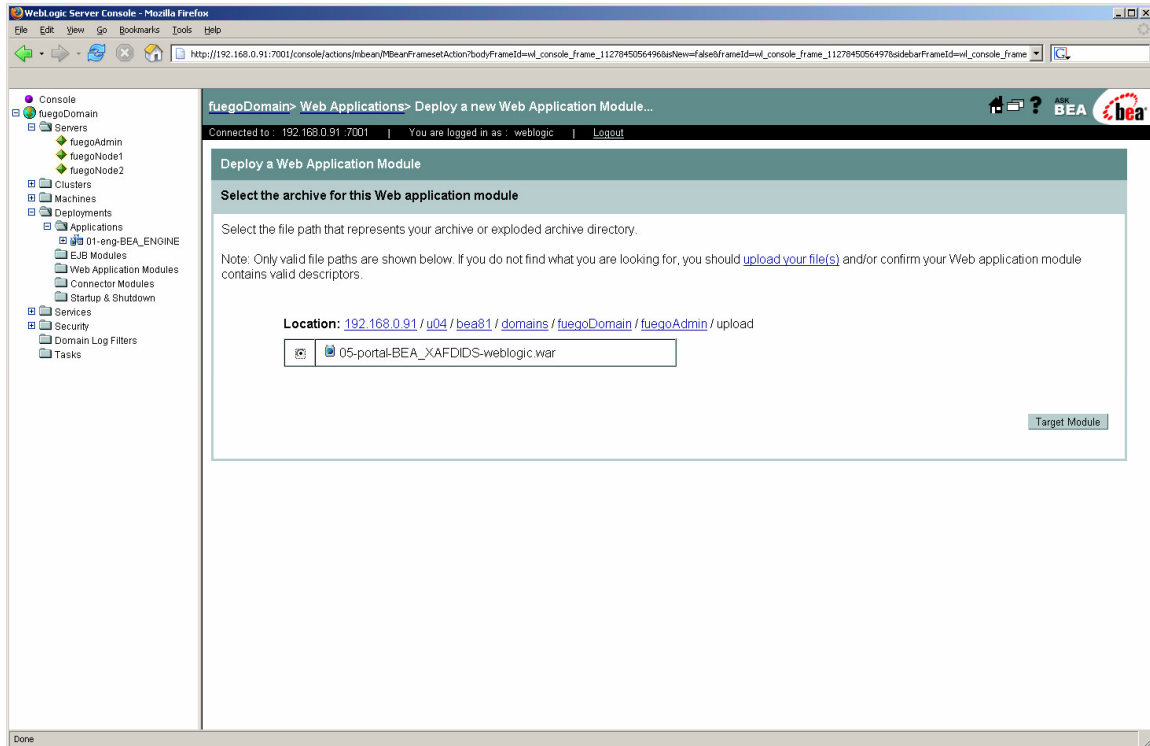
Go to “upload your file” to access the page in which you will be able to get the war file.



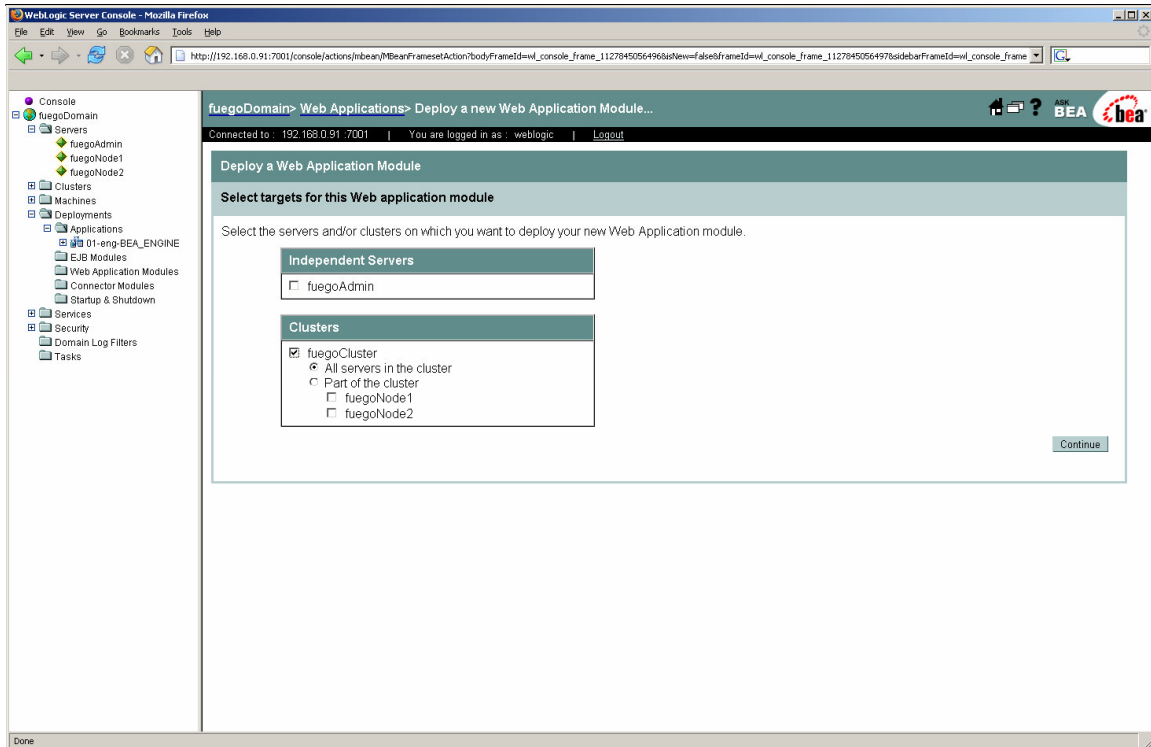
Expand your war in a directory, press the “Browse” button and select you war directory.



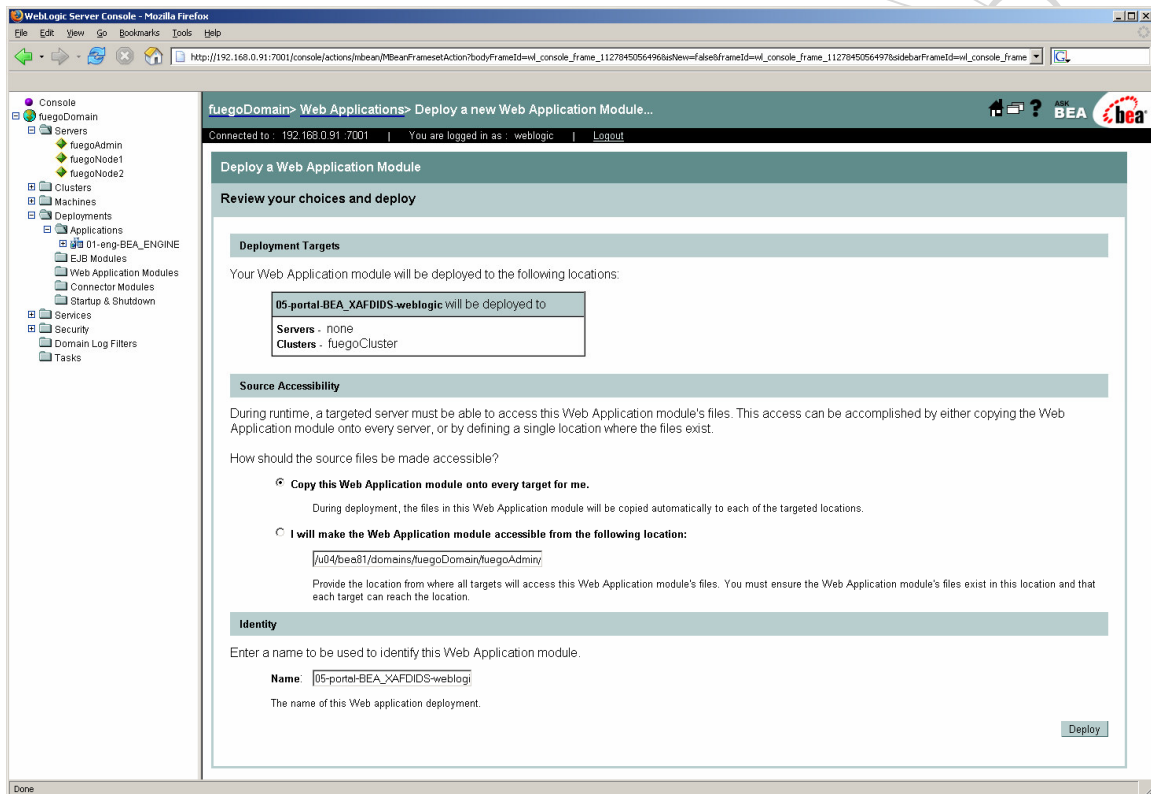
Press “Upload” to put the war inside a WebLogic directory.



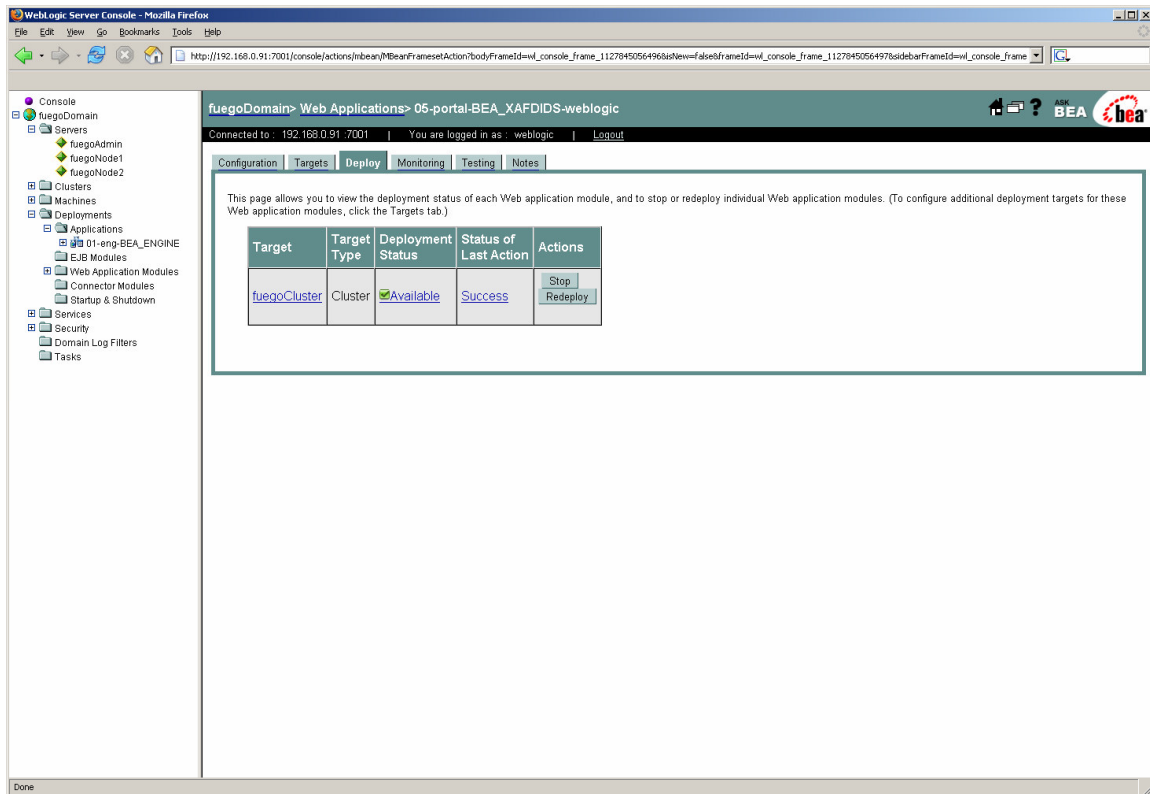
Now, the webLogic console should find the war application and WebLogic Administrator will be able to select it.



Now, the target server or cluster in which the WAR application is going to be deployed must be selected.



Press “Deploy” to install your portal application. After that, the WAR should appear as an installed application.

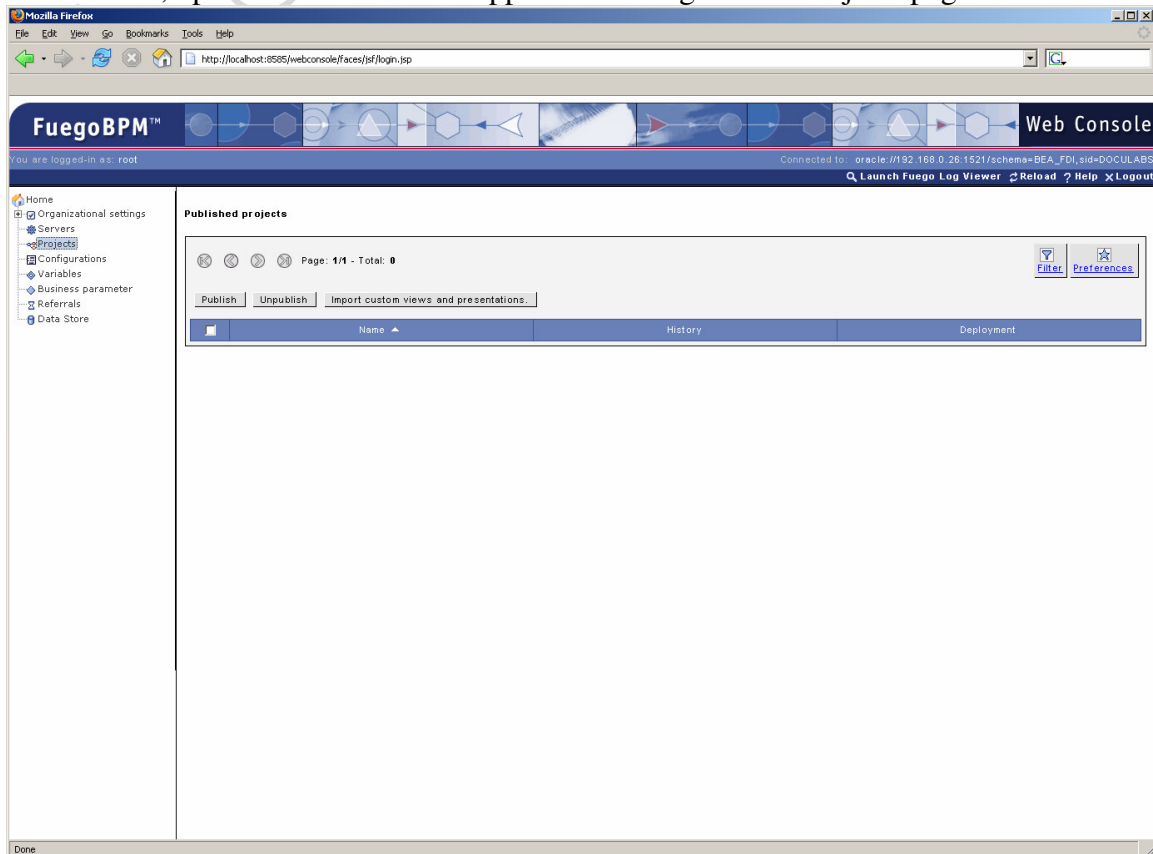


Fuego Portal Console application (WAR) can be deployed following the same procedure described for Fuego Portal.

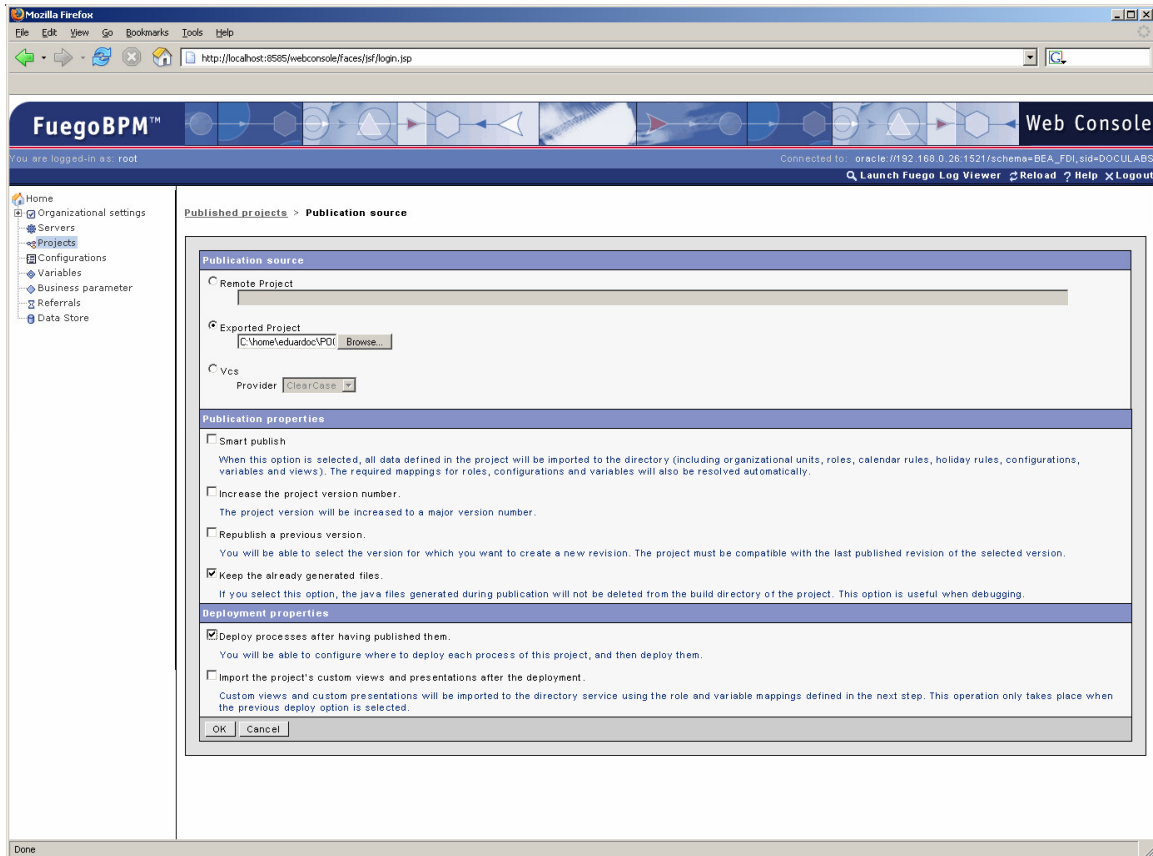
Publishing and deploying a project into a Fuego Engine

Fuego projects can be deployed into WebLogic using the Fuego Deployer application, but before that, they must be activated in Fuego itself, using Web Console application. This guide describes a summary of the Fuego publish and deploy procedure. For more details and information, please see the Fuego Administrative Guide documentation.

First, open the Web Console application and go to the Projects page.

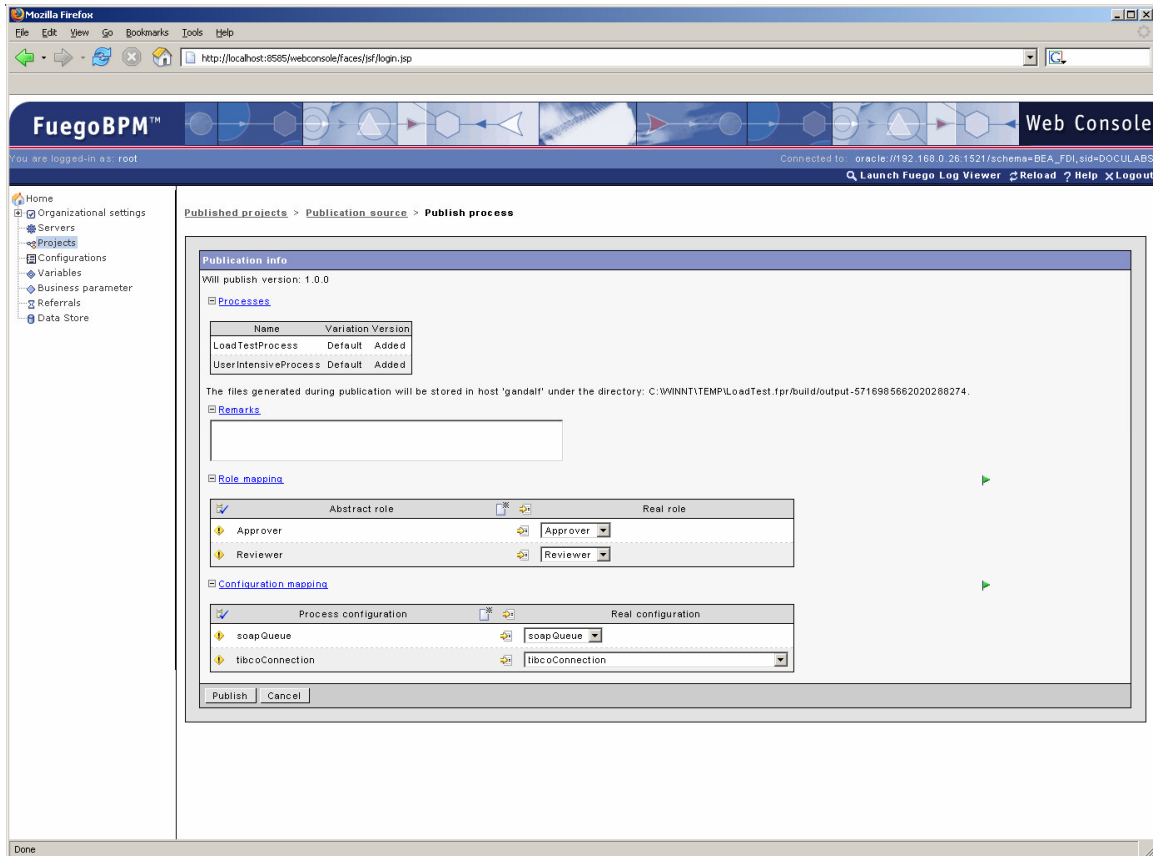


Press the “Publish” button to initiate the installation procedure.

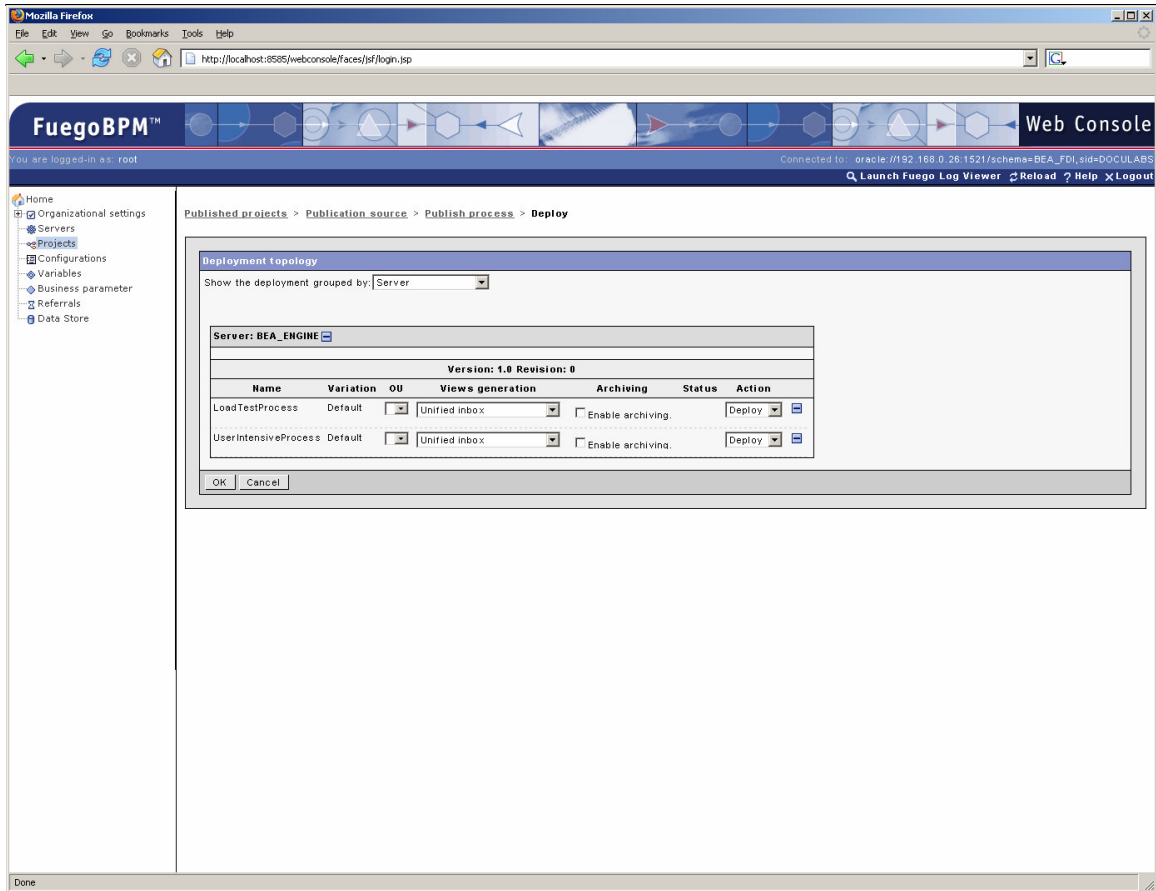


Select the project source (remote project, exported project or VCS). It is recommended to keep generated files (in case Fuego code must be inspected due to the appearance of an error) and to automatically active (deploy) the published processes.

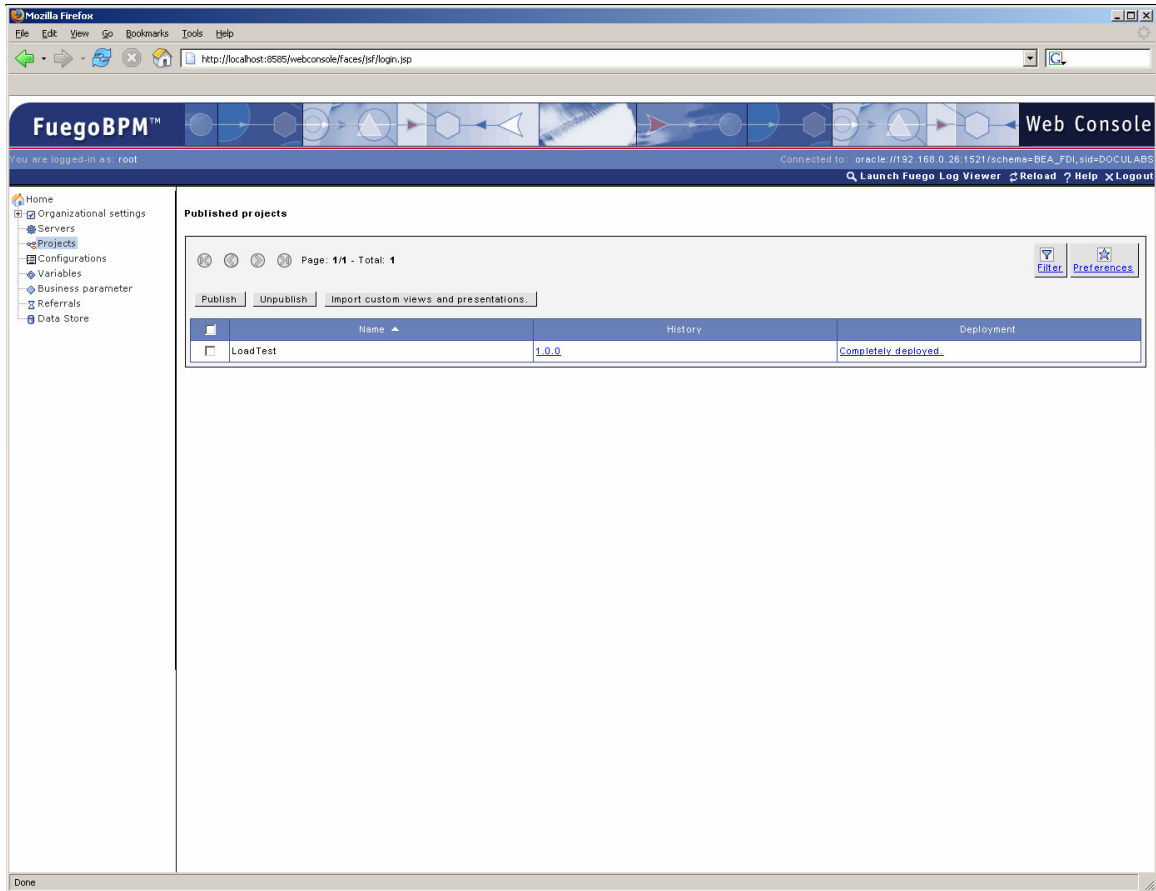
Have in mind that at this point, Fuego administrator should have previously created all necessary roles, configurations, variables, etc.



Map all needed project roles, configurations and variables. Now, an activation (deploy) screen appears in which Fuego Administrator should select the engine for each process of the project and the organizational unit in which it will be installed.



After that, the project will appear as installed, as depicted in the following screen.



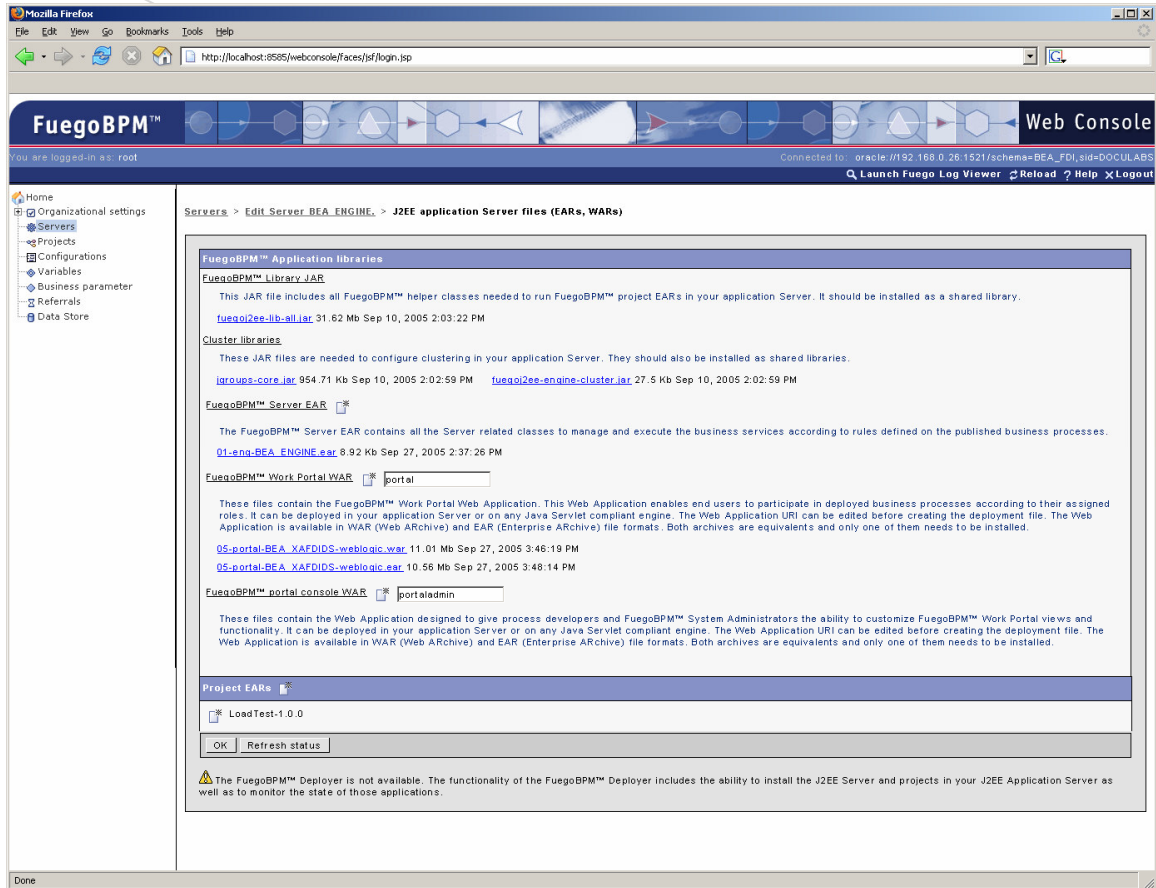
Now the project is active in Fuego installation. The last step is to deploy it inside WebLogic, so it can be accessed by all authorized users.

Deploying the Fuego Project EAR in WebLogic

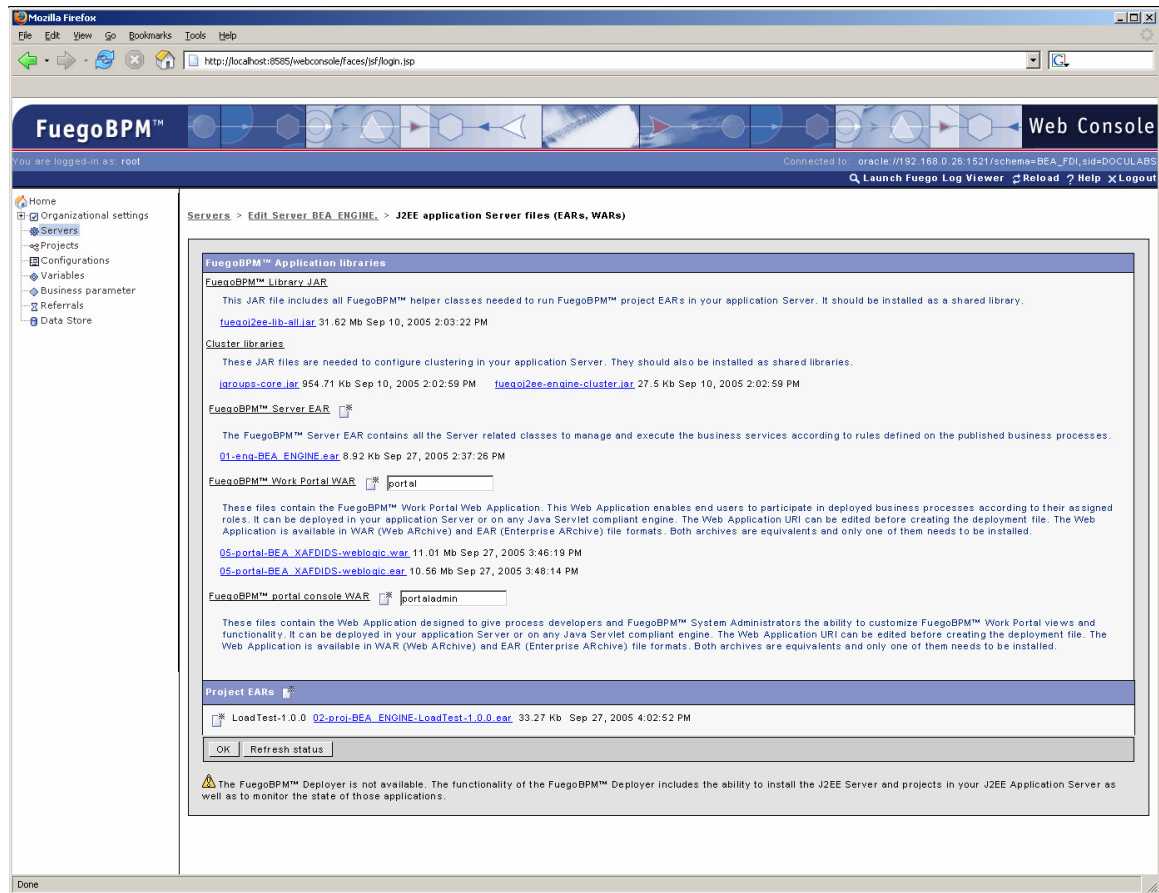
As it was done while deploying all main Fuego applications, a Fuego project file (ear) will also be needed in order to be deployed inside WebLogic.

To do that, Fuego Administrator should go to Fuego Web Console and open “J2EE application server files (EARs, WARs)” page, placed inside the server properties link.

There, he will see the recently installed project inside ‘Project EARs’ area.



Press the “create project ear file” option to generate the ear file.



If Fuego Deployer application has been installed, Fuego Administrator will be able to active the project by selecting the “Install the project ear in Application Server” button.

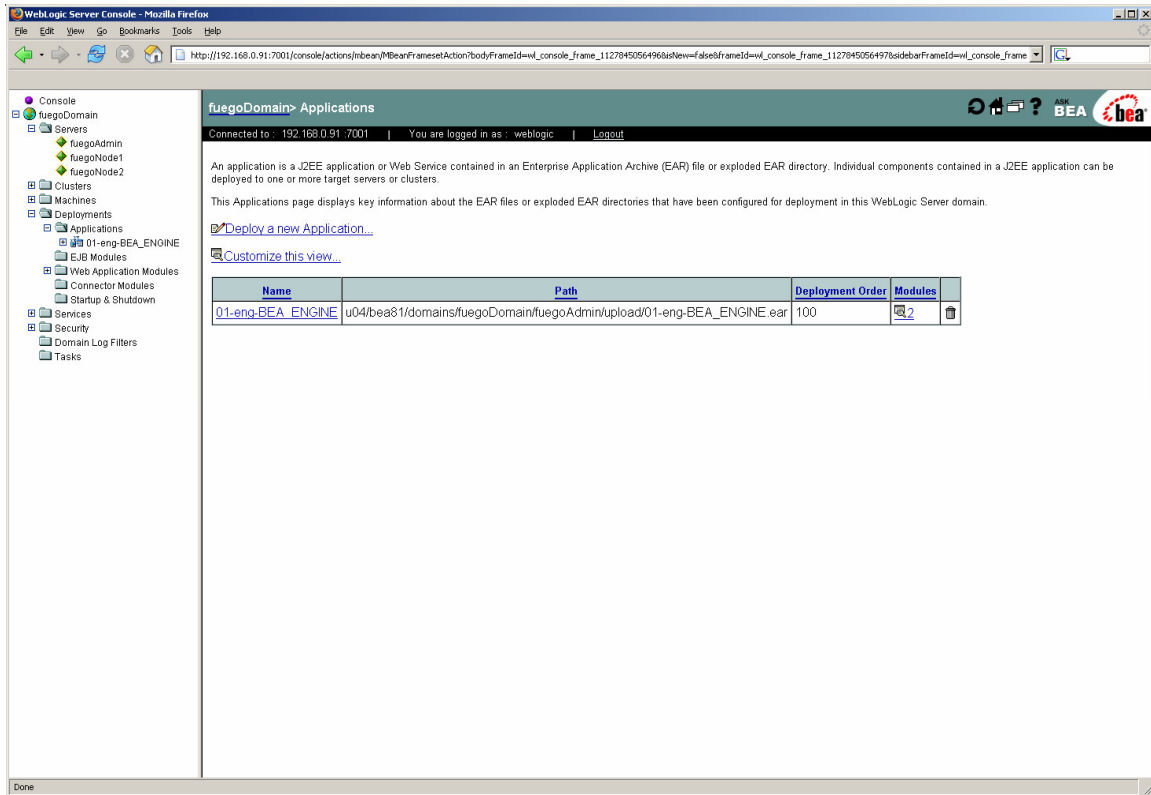
After that, the installed project should appear this way:



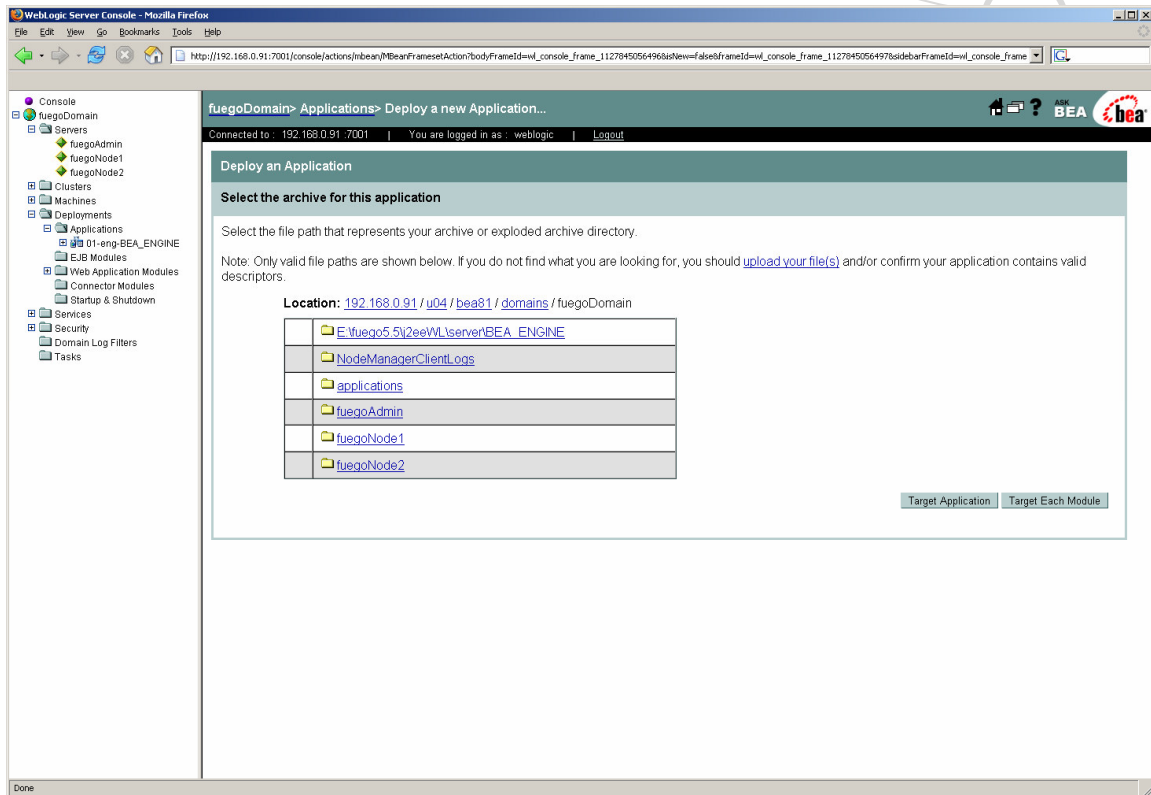
If Fuego deployer is not available, Fuego projects may be manually installed inside WebLogic server by following these steps:

First, download the ear file from Fuego Web Console.

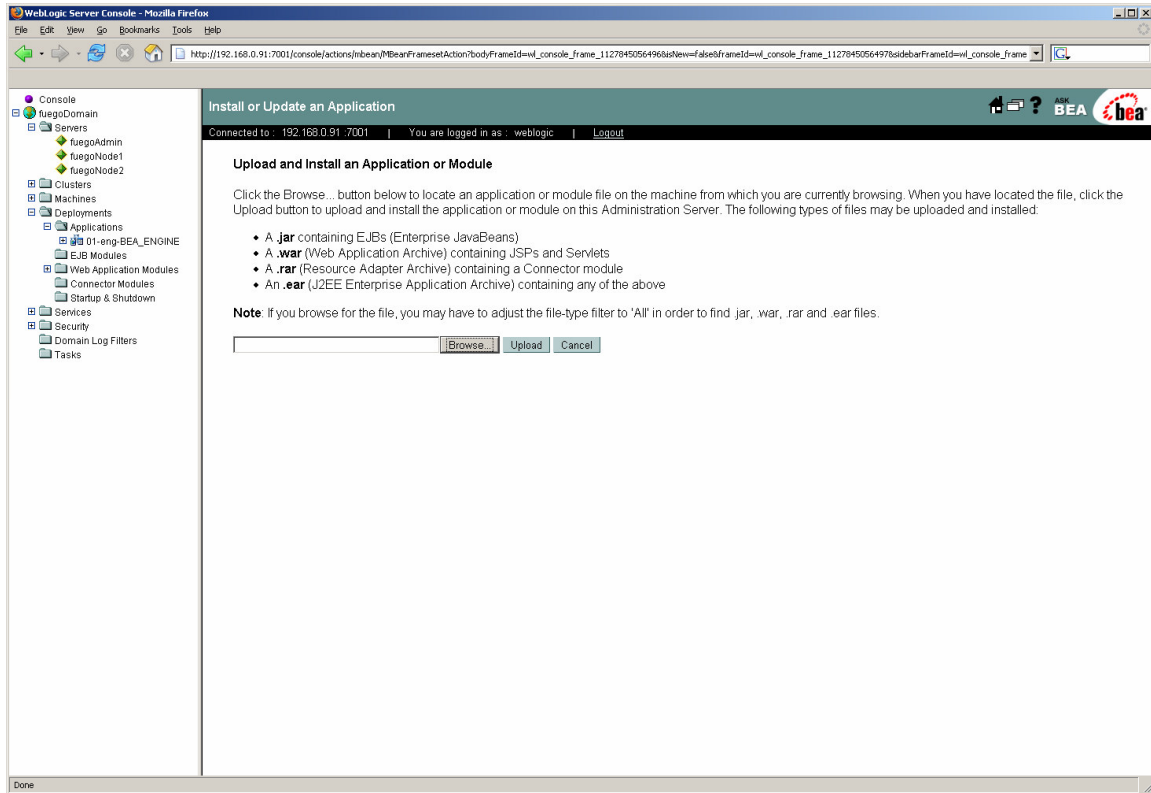
Then, go to WebLogic console and open the “Applications” page.



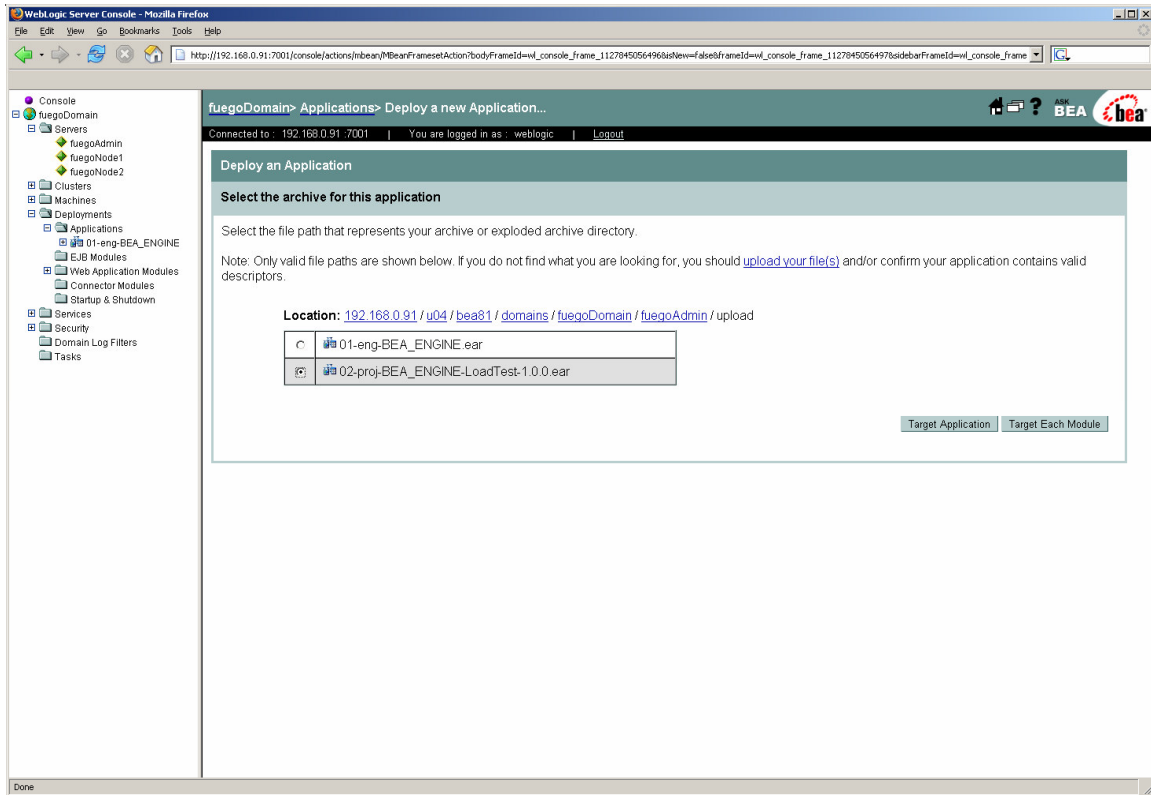
Select the “Deploy a new Application...” link.



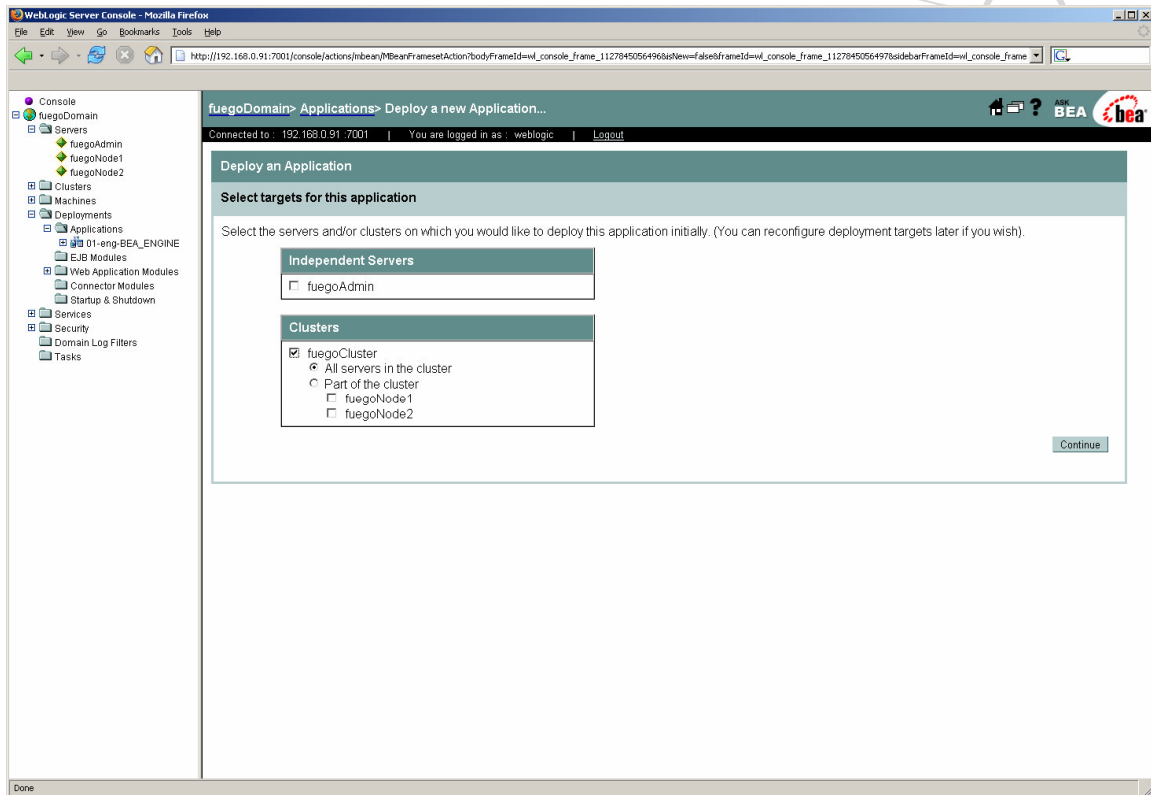
Browse you directory to find the project ear file that will be installed.



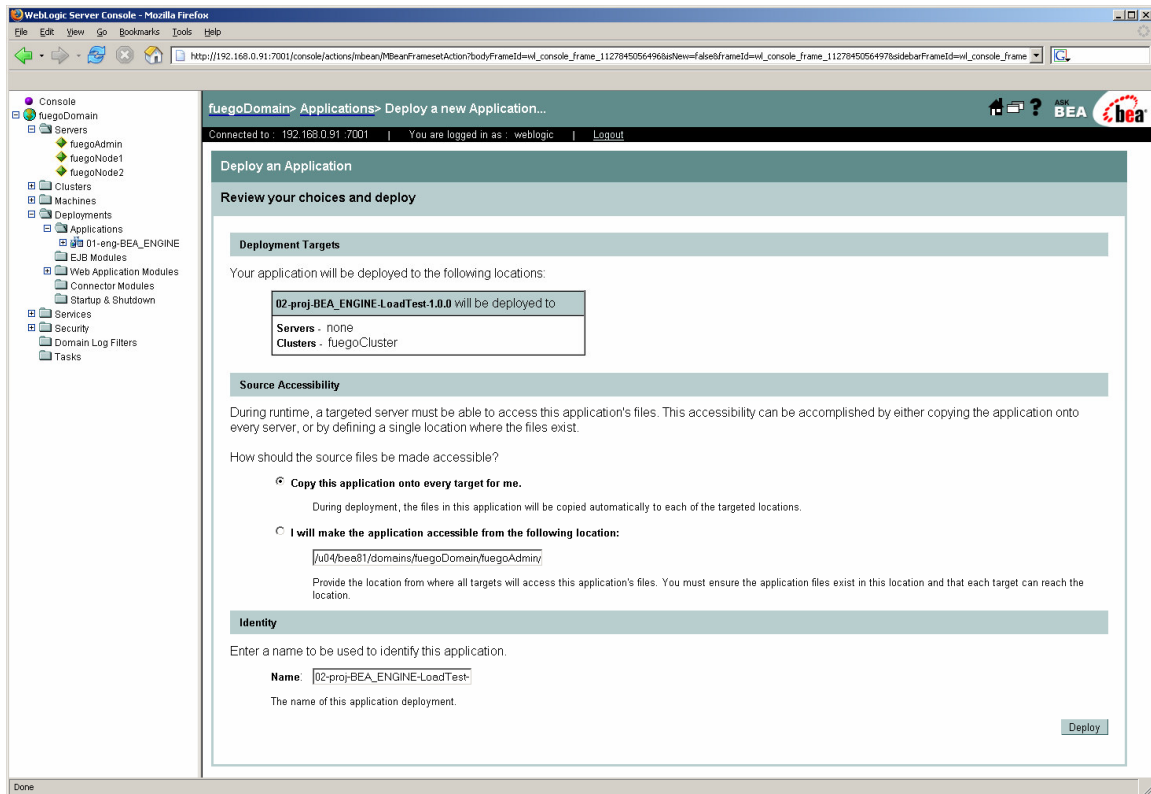
Upload the project file by choosing the appropriate button.



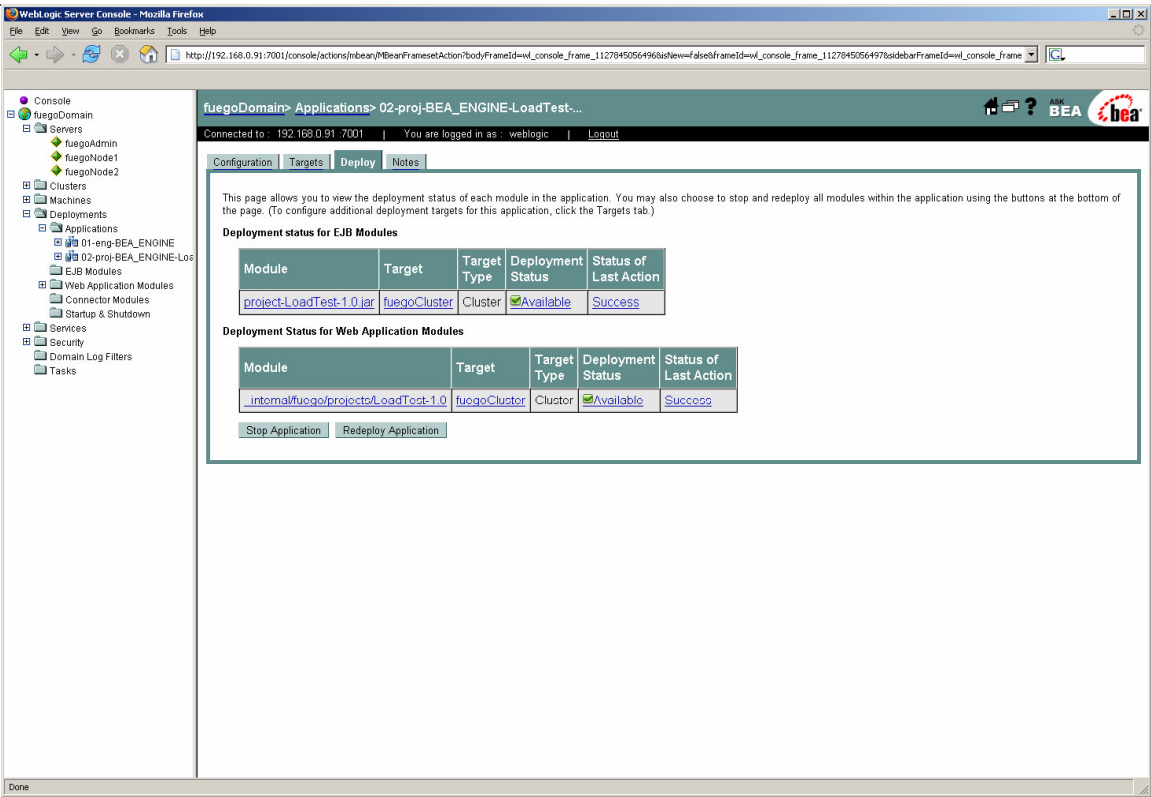
Select the Fuego project reference as the target application to be installed.



Now, the target server for this application should be selected. Press the “Deploy” button to install the project application.



After that, the project should appear as an active WebLogic application (as it's shown in the following picture).



The Fuego application project is now ready to be used with WebLogic Application Server.

Connecting to the Work Portal

The next step is to connect to the Work Portal to check that the deployment has been done successfully. You can connect to the Portal with a URL similar to this one: <http://host:port/portal>, where host is the host name where the Application Server is deployed (at least one of the nodes) and port is the port where the BEA Application Server is listening for incoming HTTP connection (this will be the BEA Embedded Web Server). “portal” is the out of the box Web Application name but if you have decided to change its name, you will need to adapt the URL mentioned above.

Troubleshooting

Example 1: Exception while trying to add Fuego server application.

weblogic.management.ApplicationException:

Exception:weblogic.management.ApplicationException: prepare failed for engine-BEA_ENGINE.jar

Module: engine-BEA_ENGINE.jar Error: Exception preparing module: EJBModule(engine-BEA_ENGINE.jar,status=NEW)

Unable to deploy EJB: C:\bea\user_projects\domains\mydomain\myserver\wlnotdelete\Fuego-BEA_ENGINE\engine-BEA_ENGINE.jar from engine-BEA_ENGINE.jar:

weblogic.ejb20.deployer.DeploymentDescriptorException: Unable to load a class specified in your ejb-jar.xml:

fuego.ejbengine.EJBSecureEngineAdapter

```

    at weblogic.ejb20.deployer.MBeanDeploymentInfoImpl.initializeBeanInfos(MBeanDeploymentInfoImpl.java:550)
    at weblogic.ejb20.deployer.MBeanDeploymentInfoImpl.<init>(MBeanDeploymentInfoImpl.java:232)
    at weblogic.ejb20.deployer.EJBDeployer.prepare(EJBDeployer.java:1262)
    at weblogic.ejb20.deployer.EJBModule.prepare(EJBModule.java:477)
    at weblogic.j2ee.J2EEApplicationContainer.prepareModule(J2EEApplicationContainer.java:2962)
    at weblogic.j2ee.J2EEApplicationContainer.prepareModules(J2EEApplicationContainer.java:1534)
    at weblogic.j2ee.J2EEApplicationContainer.prepare(J2EEApplicationContainer.java:1188)
    at weblogic.j2ee.J2EEApplicationContainer.prepare(J2EEApplicationContainer.java:1031)
    at
weblogic.management.deploy.slave.SlaveDeployer$ComponentActivateTask.prepareContainer(SlaveDeployer.java:2602)
    at weblogic.management.deploy.slave.SlaveDeployer$ActivateTask.createContainer(SlaveDeployer.java:2552)
    at weblogic.management.deploy.slave.SlaveDeployer$ActivateTask.prepare(SlaveDeployer.java:2474)
    at weblogic.management.deploy.slave.SlaveDeployer.processPrepareTask(SlaveDeployer.java:798)
    at weblogic.management.deploy.slave.SlaveDeployer.prepareDelta(SlaveDeployer.java:507)
    at weblogic.management.deploy.slave.SlaveDeployer.prepareUpdate(SlaveDeployer.java:465)
    at weblogic.drs.internal.SlaveCallbackHandler$1.execute(SlaveCallbackHandler.java:25)
    at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:197)
    at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:170)

```

Resolution: Fuego libraries had not been correctly added to WebLogic CLASSPATH.

Example 2: Exception while starting server application.

BEA-101126 Dec 19, 2005 1:05:25 PM ART Error HTTP

[ServletContext(id=27510707,name=/fuegoServices,context-path

=/fuegoServices)] Error casting servlet: "engineStartup" to
javax.servlet.Servlet

java.lang.ClassCastException

```

    at com.sun.rowset.CachedRowSetImpl.getTimestamp(CachedRowSetImpl.java:2204)
    at com.sun.rowset.CachedRowSetImpl.getTimestamp(CachedRowSetImpl.java:2653)
    at fuego.directory.provider.jdbc.oracle.OraclePersistenceManager.getTimeAttribute(OraclePersistenceManager.java:84)
    at fuego.directory.provider.jdbc.JDBCServiceAccessor.getAuditInfo(JDBCServiceAccessor.java:183)
    at fuego.directory.provider.jdbc.JDBCParticipantsAccessor.buildHumanParticipant(JDBCParticipantsAccessor.java:723)
    at fuego.directory.provider.jdbc.HumanParticipantAdaptor.createFrom(HumanParticipantAdaptor.java:47)
    at fuego.directory.provider.jdbc.JDBCPersistenceManager.fetch(JDBCPersistenceManager.java:351)
    at fuego.directory.provider.jdbc.JDBCPersistenceManager.fetch(JDBCPersistenceManager.java:310)
    at fuego.directory.provider.jdbc.JDBCParticipantsAccessor.fetchHumanParticipant(JDBCParticipantsAccessor.java:1059)
    at fuego.directory.provider.jdbc.JDBCParticipantsAccessor.fetchHumanParticipant(JDBCParticipantsAccessor.java:282)
    at
fuego.directory.provider.jdbc.JDBCAuthenticationAccessor.getSessionParticipant(JDBCAuthenticationAccessor.java:417)
    at
fuego.directory.provider.jdbc.JDBCAuthenticationAccessor.authenticateInternal(JDBCAuthenticationAccessor.java:309)
    at fuego.directory.provider.jdbc.JDBCAuthenticationAccessor.authenticate(JDBCAuthenticationAccessor.java:71)
    at fuego.directory.provider.jdbc.JDBCAuthenticationAccessor.connect(JDBCAuthenticationAccessor.java:115)
    at fuego.directory.provider.DirectorySessionImpl.connect(DirectorySessionImpl.java:169)

```

	2005 Fuego Inc. - Company Proprietary and Confidential.	
	All Rights Reserved.	Page 91 of 92

```

at fuego.directory.provider.Factory.startSession(Factory.java:461)
at fuego.directory.provider.jdbc.j2ee.RemoteJdbcDirectoryFactory.startSession(RemoteJdbcDirectoryFactory.java:197)
at fuego.directory.Directory.startSession(Directory.java:651)
at fuego.ejbengine.Engine.initDirectorySession(Engine.java:334)
at fuego.ejbengine.Engine.start(Engine.java:116)
at fuego.ejbengine.servlet.SchedulerServlet.init(SchedulerServlet.java:93)
at weblogic.servlet.internal.ServletStubImpl$ServletInitAction.run(ServletStubImpl.java:993)
at weblogic.security.acl.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:317)
at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:118)
at weblogic.servlet.internal.ServletStubImpl.createServlet(ServletStubImpl.java:869)
at weblogic.servlet.internal.ServletStubImpl.createInstances(ServletStubImpl.java:848)
at weblogic.servlet.internal.ServletStubImpl.prepareServlet(ServletStubImpl.java:787)
at weblogic.servlet.internal.WebAppServletContext.preloadServlet(WebAppServletContext.java:3252)
at weblogic.servlet.internal.WebAppServletContext.preloadServlets(WebAppServletContext.java:3197)
at weblogic.servlet.internal.WebAppServletContext.preloadResources(WebAppServletContext.java:3174)
at weblogic.servlet.internal.WebAppServletContext.setStarted(WebAppServletContext.java:5647)
at weblogic.servlet.internal.WebAppModule.start(WebAppModule.java:869)
at weblogic.j2ee.J2EEApplicationContainer.start(J2EEApplicationContainer.java:2022)
at weblogic.j2ee.J2EEApplicationContainer.activate(J2EEApplicationContainer.java:2063)
at
weblogic.management.deploy.slave.SlaveDeployer$ComponentActivateTask.activateContainer(SlaveDeployer.java:2592)
at weblogic.management.deploy.slave.SlaveDeployer$ActivateTask.doCommit(SlaveDeployer.java:2515)
at weblogic.management.deploy.slave.SlaveDeployer$Task.commit(SlaveDeployer.java:2317)
at weblogic.management.deploy.slave.SlaveDeployer$Task.checkAutoCommit(SlaveDeployer.java:2399)
at weblogic.management.deploy.slave.SlaveDeployer$Task.prepare(SlaveDeployer.java:2311)
at weblogic.management.deploy.slave.SlaveDeployer$ActivateTask.prepare(SlaveDeployer.java:2479)
at weblogic.management.deploy.slave.SlaveDeployer.processPrepareTask(SlaveDeployer.java:798)
at weblogic.management.deploy.slave.SlaveDeployer.prepareDelta(SlaveDeployer.java:507)
at weblogic.management.deploy.slave.SlaveDeployer.prepareUpdate(SlaveDeployer.java:465)
at weblogic.drs.internal.SlaveCallbackHandler$1.execute(SlaveCallbackHandler.java:25)
at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:197)
at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:170)

```

Resolution: An incorrect version of jdbc driver had been used. After configuring the right one, Fuego server was started without any problem.