16L06

IBM Integration Bus

Web Administration Interface
File-based Administration Security
LDAP Authentication

Featuring:

- Browser and remote admin access with HTTPS
- Web browser administration without role security
- Authentication with LDAP
- Configuring Web UI with role-based security
- Web browser admin with role security
- BAR deploy and BAR override
- Toolkit authorization
- Operational Policy administration

June 2016

Hands-on lab built at product
Version 10.0.0.5

1. INTRODUCTION 3

1.1 RESET TESTNODE_IIBUSER SECURITY 3
1.2 OUTLINE OF LAB 4

2. CONFIGURE REMOTE ADMINISTRATION USING HTTPS 5

2.1 SET THE PORT FOR THE WEBADMIN LISTENER FOR HTTPS 6
2.2 SET THE KEYSTORE FILENAME FOR THE WEBADMIN LISTENER 6
2.3 SET THE KEYSTORE PASSWORD FOR THE WEBADMIN LISTENER 6
2.4 ENABLE SSL FOR THE WEBADMIN LISTENER 6
2.5 REVIEW THE CONFIGURATION 7
2.6 LOGIN WITH A WEB BROWSER 8

3. PREPARE THE APPLICATIONS 9

3.1 OVERRIDING BAR FILE 14

4. USING WEB BROWSER INTERFACE WITHOUT ADMIN SECURITY 17

4.1 ADMINISTRATION 17
4.2 INTEGRATION NODE AND SERVER PROPERTIES 20

5. AUTHENTICATION AND AUTHORISATION FOR REMOTE USERS 22

5.1 START THE LDAP SERVER 23
5.2 CONFIGURE AUTHENTICATION WITH LDAP FOR TESTNODE 25
5.3 ACTIVATE ADMINISTRATION SECURITY FOR TESTNODE_IIBUSER 26
5.4 DEFINE ADMINISTRATION ROLES AND SET FILE-BASED PERMISSIONS 28
5.5 DEFINE THE WEB USERS FOR TESTNODE_IIBUSER 31

6. USING THE WEB BROWSER INTERFACE WITH ADMIN SECURITY 33

6.1 USER WITH READ-ONLY ACCESS 33
6.2 USER WITH WRITE ACCESS 36
6.3 THE WEB ADMIN INTERFACE FOR A USER WITH 'ALL' ACCESS 41
6.4 ADMINISTRATION OF OPERATIONAL POLICY 45

7. INTEGRATION TOOLKIT AUTHORIZATION 47

END OF LAB GUIDE 52

1. Introduction

IBM Integration Bus V10 has enhanced the web browser user Interface allowing administration of Integration Nodes. This has replaced the majority of the admin function that was previously available in the IB Explorer (IBX). A small amount of function was moved from the IBX to the Integration Toolkit – Configurable Services, Policy sets.

The Web Browser User Interface (known in this document as the Web UI) provides the following capabilities:

1. Ability to allow defined web users to perform administration at defined level
2. Ability to perform update actions against deployed resources (start, stop, etc.)
3. Ability to view trace and log files through the Web UI
4. Ability to view and update Node policy documents
5. Ability to deploy application BAR files
6. Resource statistics

This lab will demonstrate some of these points, while others are the subject of separate labs.

1.1 Reset TESTNODE_iibuser security

The first part of this lab assumes that security has not been activated for TESTNODE_iibuser. If security has been activated on your system, deactivate it now. To check, run the command below in an Integration Console:

```
mqsireportbroker TESTNODE_iibuser
```

Ensure that you see the line

```
Administration security = 'inactive'
```

To set security off, use the following commands in an Integration Command Console:

```
mqsistop TESTNODE_iibuser

mqsichangebroker TESTNODE_iibuser -s inactive

mqsistart TESTNODE_iibuser
```

Finally, change directory to c:\student10\webadmin\install. Run the command

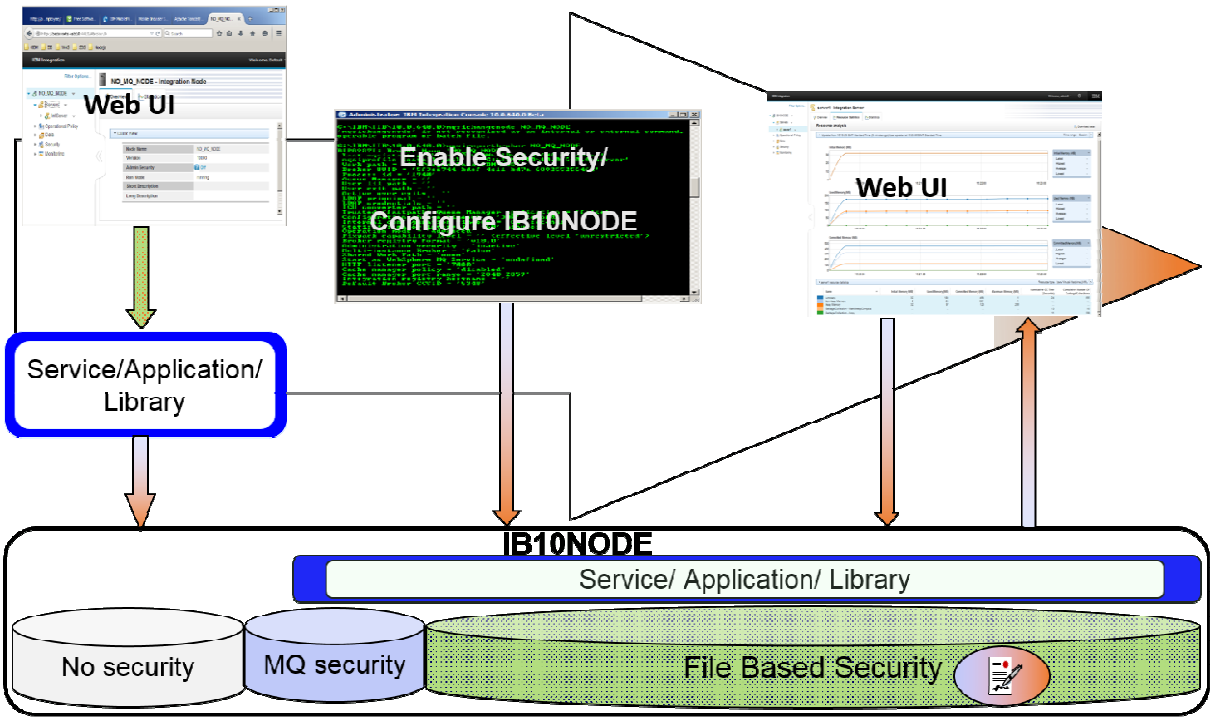
```
reset_admin_users_ACL_using_FileAuth.cmd
```

Ignore any messages concerning userids that are not defined.

1.2 Outline of Lab

This lab will show you the following functions:

- Configuration of HTTPS connections
- Deployment of applications and libraries from the Web UI
- Explore Web UI without role-based security
- Configure Web UI with role-based security
    - Activate file-based security
    - Set administration roles
    - Define web users
- Explore Web UI with role-based security:
    - Web UI for ‘Read-only’ user
    - Web UI for ‘Read/ Write’ user
    - Web UI for ‘Read/ Write/ Execute’ user



2. Configure remote administration using HTTPS

The default connection protocol for admin users for an IIB node is HTTP. All remote administration clients connect using HTTP, and these requests are handled by the webadmin listener in the IIB node. The webadmin listener is defined and started by default, so it will normally be a requirement to provide additional levels of security for administration of an IIB node.

The complete set of commands that are required for this are shown in full. Note that the command options are shown on separate lines for readability. However, when you execute them in an IIB Command Console, simply append each parameter to the command.

This first section will show you how to change the webadmin listener to use HTTPS. This lab assumes that the appropriate keystores and truststores are already available. These will typically be defined and created by the security department of an enterprise. This lab scenario is illustrated with the use of self-signed certificates. The IIB WS-Security lab document shows you how we generated these certificates and related key/truststores.

2.1 Set the port for the webadmin listener for HTTPS

The following command sets the webadmin listener HTTPSConnector object, and specifies the precise port that will be used for this. For this lab, use port 4421 (to avoid conflicts elsewhere).

```
mqsichangeproperties TESTNODE_iibuser
      -b webadmin
      -o HTTPSConnector
      -n port
      -v 4421
```

2.2 Set the keystore filename for the webadmin listener

The following command sets the filename of the keystore for the webadmin listener for the IIB node. IIB requires the format of the keystore to be a JKS file (java keystore).

```
mqsichangeproperties TESTNODE_iibuser
      -b webadmin
      -o HTTPSConnector
      -n keystoreFile
      -v c:\student10\webadmin\keystore\IIB.jks
```

2.3 Set the keystore password for the webadmin listener

The following command sets the password of the keystore file so that the IIB node can open the keystore file.

```
mqsichangeproperties TESTNODE_iibuser
      -b webadmin
      -o HTTPSConnector
      -n keystorePass
      -v passw0rd
```

2.4 Enable SSL for the webadmin listener

The following command enables SSL for the webadmin listener. Note that from IIB v9.0.0.3, and in IIB v10, the default SSL protocol is TLS. Use of SSLv3 is not recommended.

```
mqsichangeproperties TESTNODE_iibuser
      -b webadmin
      -o server
      -n enableSSL
      -v true
```

2.5 Review the configuration

To enable these changes to be effective, stop and restart the IIB node:

```
mqsistop TESTNODE_iibuser
mqsistart TESTNODE_iibuser
```

The IIB Event Log will show the webadmin listener starting on the specified port, using https.

```
BIP3132I: ( TESTNODE_iibuser ) The HTTP Listener has started
listening on port ''4421'' for ''WebAdmin https'' connections.
[23/12/2015 16:52:31]
```

To review the HTTPS setting, in an IIB Command Console, run the command

```
mqsireportproperties TESTNODE_iibuser
      -b webadmin
      -o HTTPSConnector
      -a
```

Output similar to this will be produced. Note that sslProtocol is set to "Platform Default", which for IIB v10 will be TLS.

```
HTTPSConnector
  uuid='HTTPSConnector'
  algorithm='Platform Default'
  clientAuth='false'
  keystoreFile='c:\student10\webadmin\keystore\IIB.jks'
  keystorePass='********'
  keystoreType='Platform Default'
  truststoreFile=''
  truststorePass=''
  truststoreType='Platform Default'
  sslProtocol='Platform Default'
  ciphers='Platform Default'
  address=''
  port='4421'
  maxPostSize=''
  acceptCount=''
      compressableMimeTypes='text/html,text/css,
      application/javascript,image/gif,image/png,application/json'
  compression='on'
  connectionLinger=''
  connectionTimeout=''
  maxHttpHeaderSize=''
  maxKeepAliveRequests=''
  maxThreads=''
  minSpareThreads=''
  noCompressionUserAgents=''
  restrictedUserAgents=''
  socketBuffer=''
  tcpNoDelay=''
  enableLookups='false'
  serverName=''

BIP8071I: Successful command completion.
```

2.6 Login with a web browser

Open a web browser (on the workshop VM, use a Firefox browser).

On the workshop VM, use the following URL:

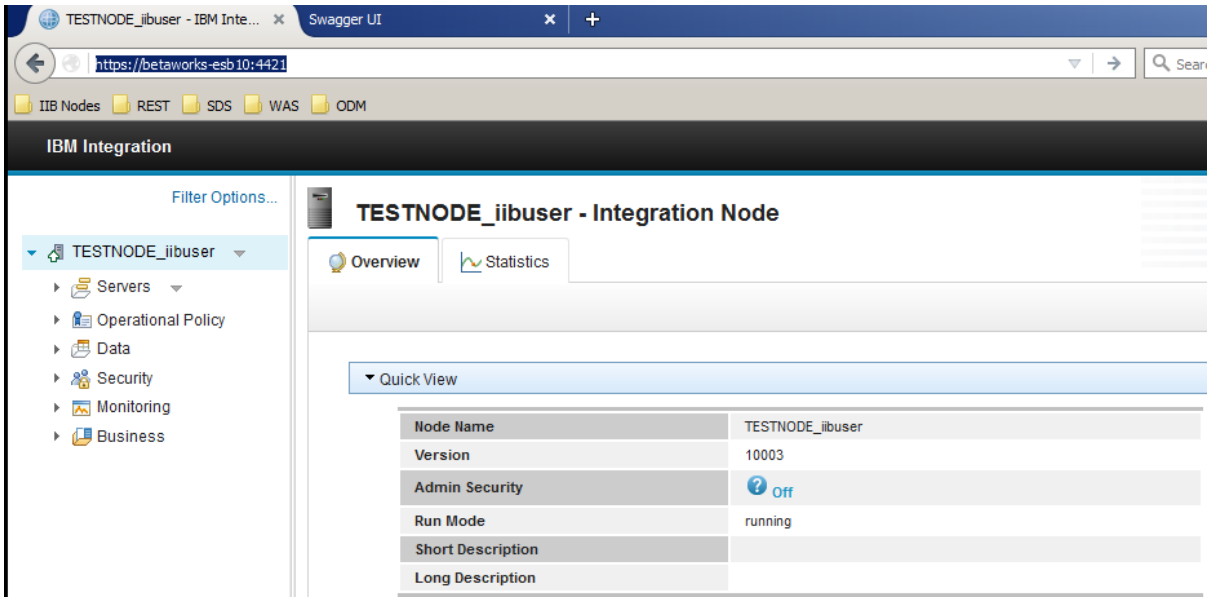
`https://betaworks-esb10:4421`

If you have not previously connected the browser to the target IIB node, you may received a certificate challenge. If you trust the IIB server, then accept the certificate challenge.

Click on '**I understand the Risks**', then 'Add Exception'.

'**Confirm Security Exception**' on the following dialog.

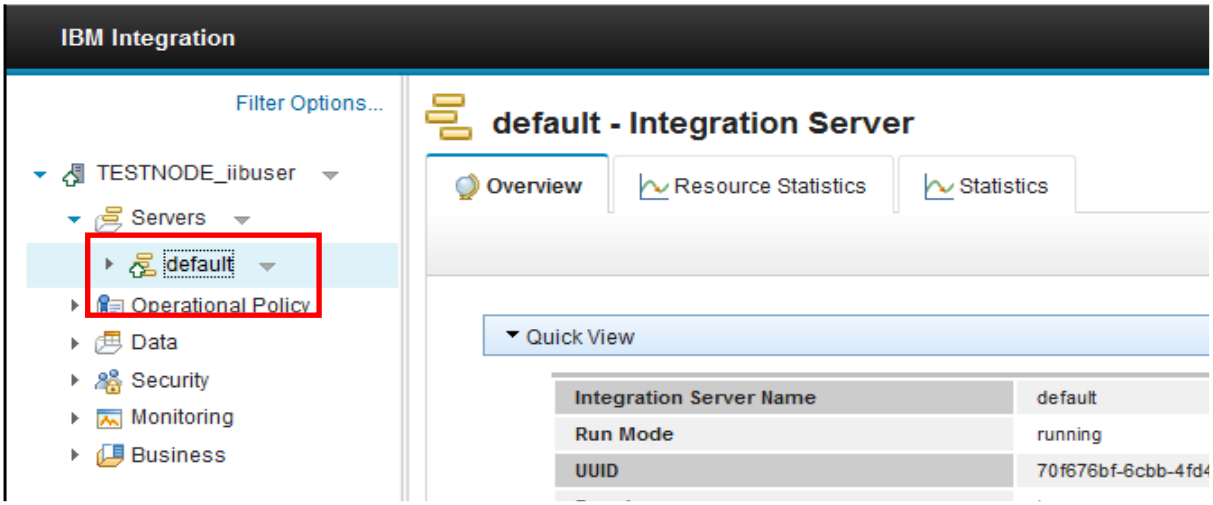
The browser will be connected to the IIB node using HTTPS.



3. Prepare the applications

1. This lab will be use the integration server ‘default'.

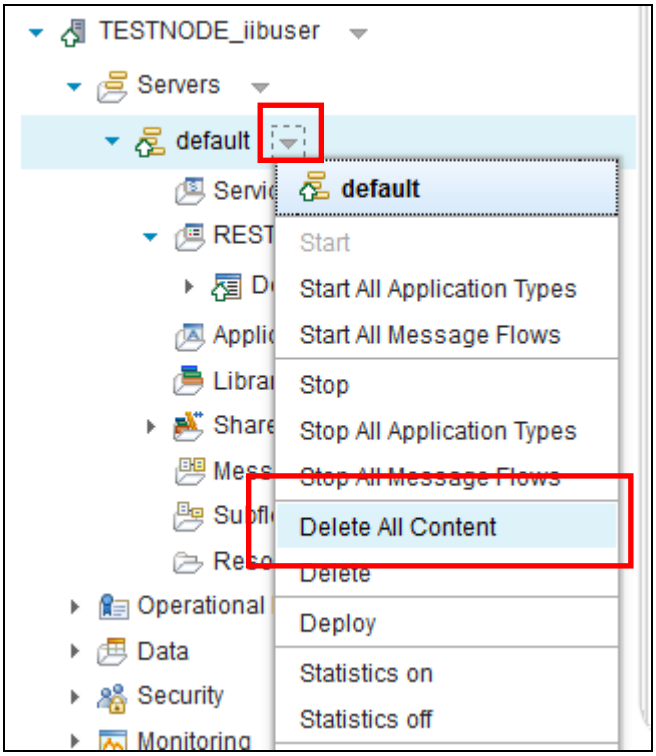
   Expand ‘Servers’ and click **default** to see its details.



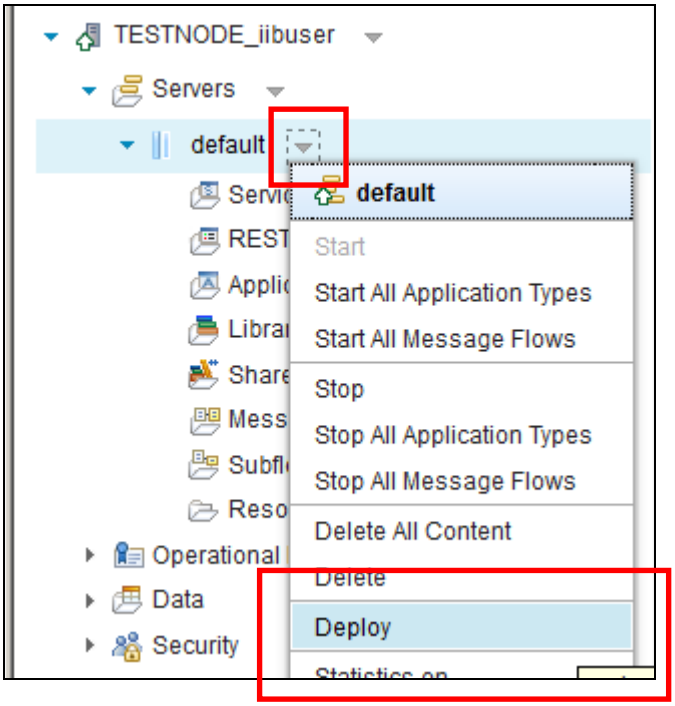
2. If you have done previous labs, you may already have resources deployed in this server. We want to deploy fresh copies of the applications to the integration server, so delete any current resources.

   On the **default** Integration Server, click to arrow to open the context menu available for the application and select "Delete All Content".

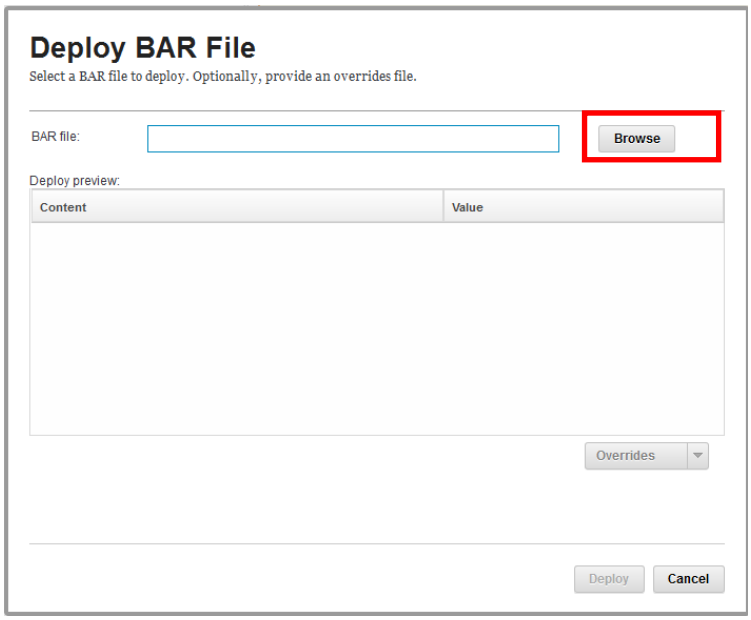
   Respond ‘Yes’ when prompted to confirm the delete of the resource.



3. Click open the context menu next to **default**. Click 'Deploy'.

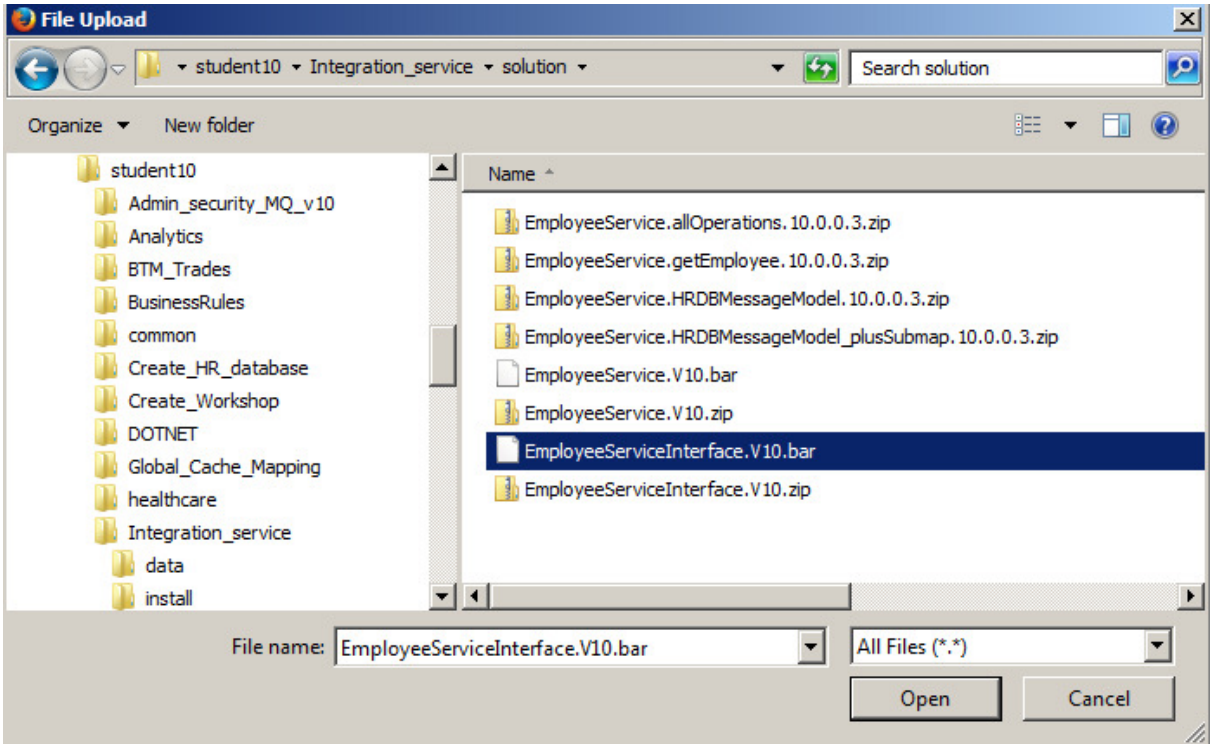


4. This will open the dialog for BAR deployment.

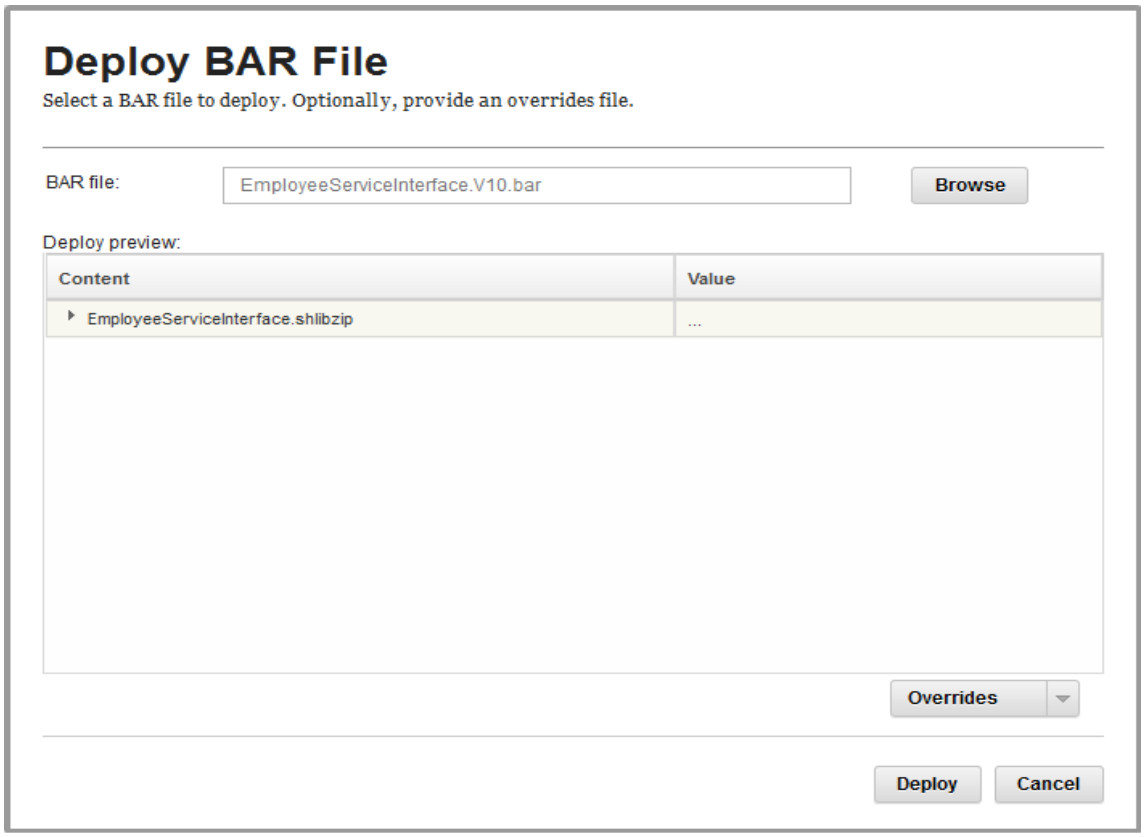


Click 'Browse' as shown and navigate to **C:\student10\Integration_service\solution**

5. Select 'EmployeeServiceInterface.V10.bar' and click Open.



6. Click 'Deploy' in the 'Deploy BAR File' window:

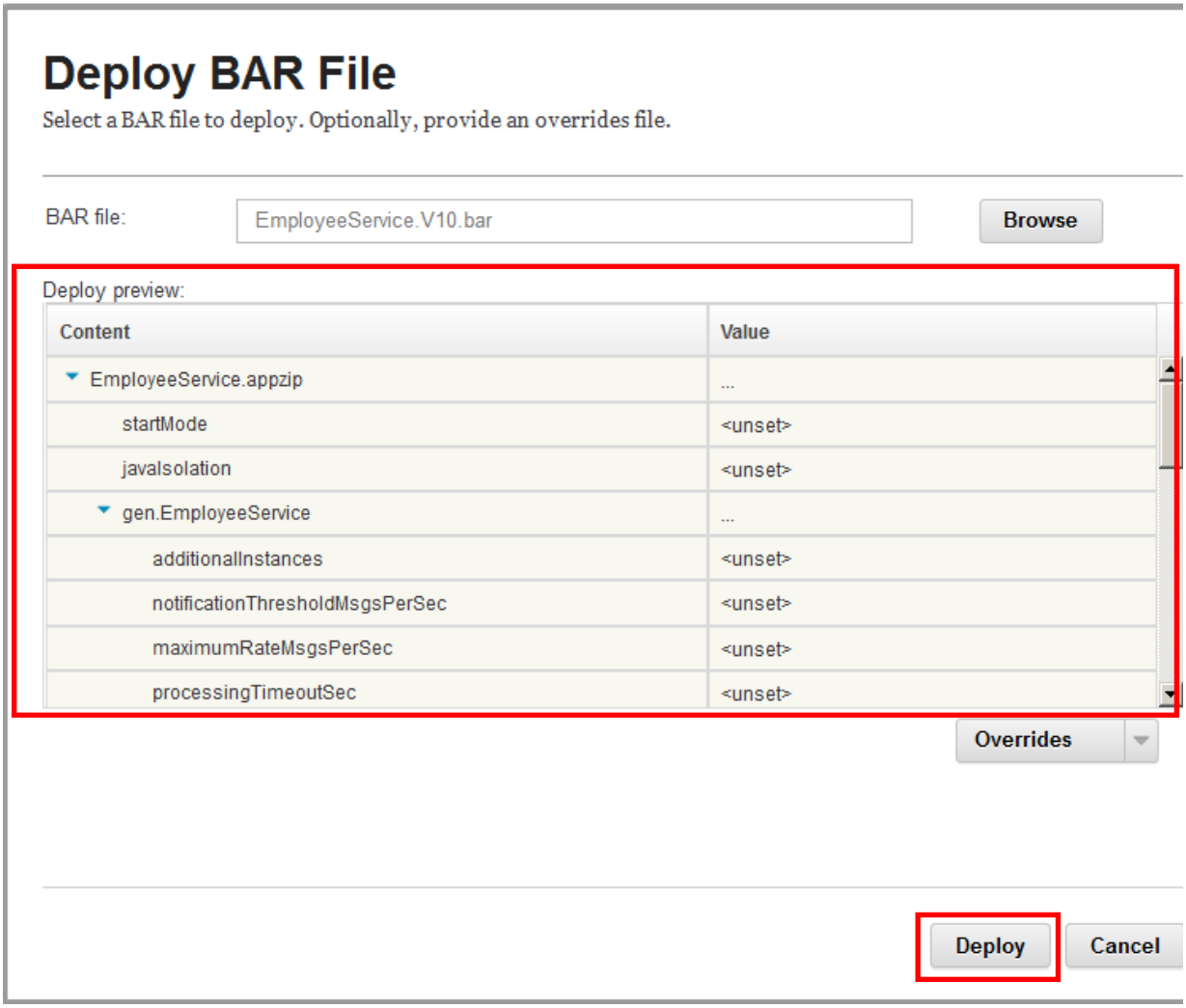


7. Repeat steps 5 and 6. This time select **EmployeeService.V10.bar** file and click 'Open'.

   In the 'Deploy BAR File' preview window you will see that the 'Deploy Preview' table has been populated with properties existing in the BAR file. Note that you need to expand 'EmployeeService.appzip' to be able to see them.

   In the 'Values' column some of the properties show **<unset>**. This means that these properties could be configured (this will be shown later in the lab).

   Once finished reviewing, click the 'Deploy' button.

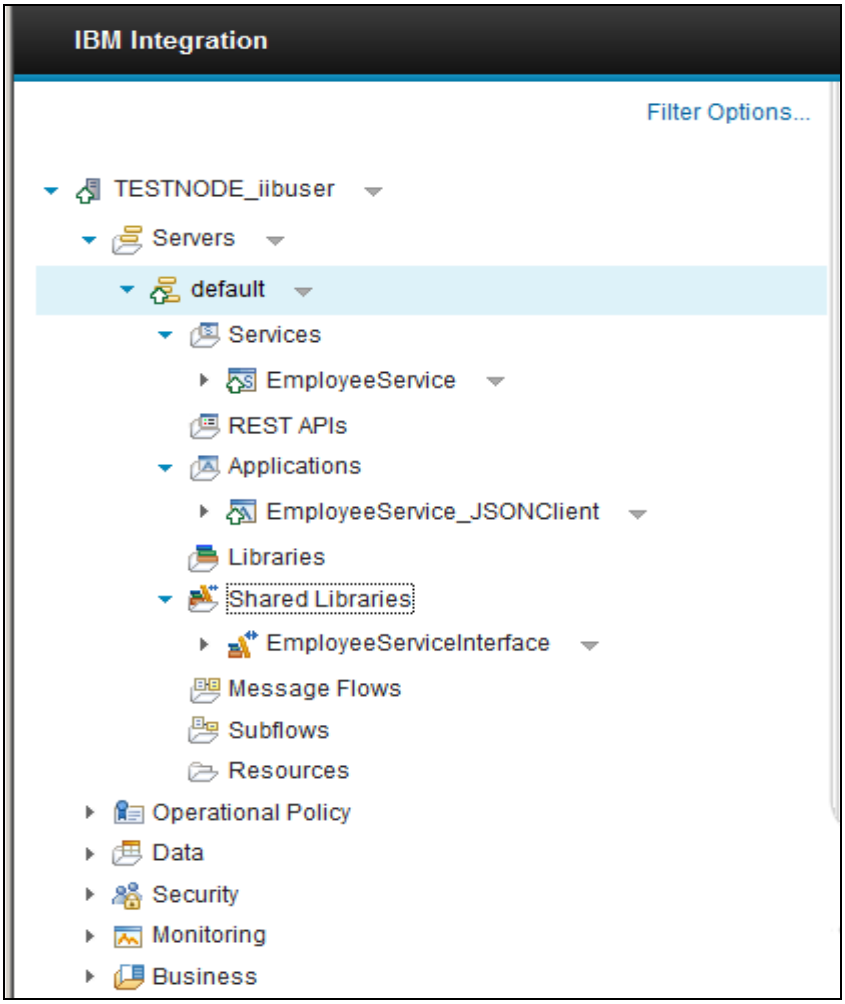


8. Repeat the above step and deploy 'EmployeeService_JSONClient.V10.bar' only this time import from **C:\student10\Integration_service_JSONClient\solution** folder.

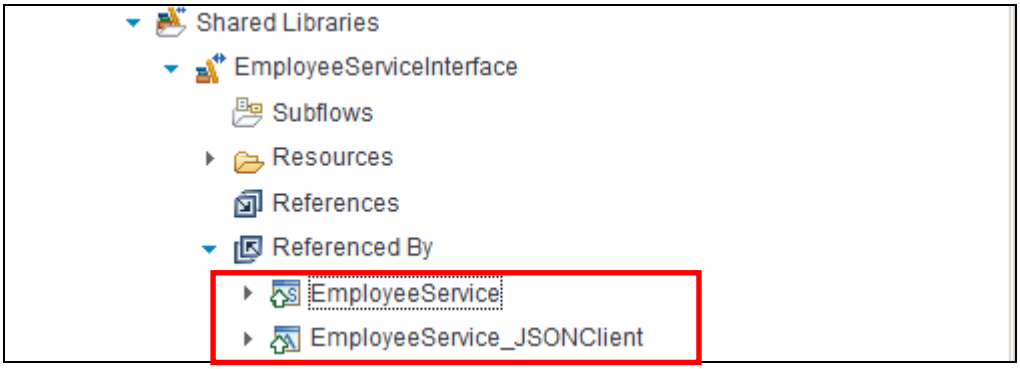
   Again, feel free to explore the properties and their values in the deployment table.

   Please note that it may take a few seconds to see the updated view while the resources are being deployed.

   You should now have resources on the default server under 'Services', 'Applications' and 'Shared Libraries' categories:



9. To see which applications or services reference the shared library, expand 'EmployeeServiceInterface' then 'Referenced by'. You will see the service and application that you imported in the previous steps.



3.1 Overriding BAR file

The Web UI in Integration Bus V10 allows you to override BAR files during the process of deployment. The 'override' uses a file with an extension **'.properties'.**, In the file, new values are specified for the properties that need to be set.

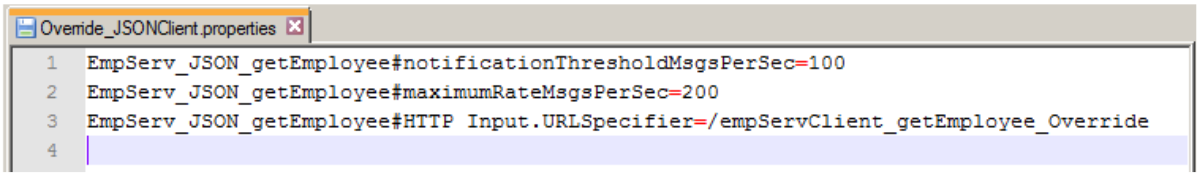
1. The override file for this lab has been created and is provided for you.

   Using Windows Explorer, go to **C:\student10\Integration_service_JSONClient\solution** folder and right-click **Override_JSONClient.properties.**

   From the menu select 'Edit with Notepad++'.

2. In the override file, the properties that will be overridden are specified following the message flow name that they refer to.

   For the URL property in addition to the message flow name, the HTTP Node name is specified (HTTP Input).

   The property names can be seen from the 'Deploy BAR File' window (step 2.1.9)

   No changes will be made here so once finished viewing the file close it.

3. In the Web UI click again on the context menu next to **default** and select Deploy.

   Navigate to **C:\student10\Integration_service_JSONClient\solution** folder and select 'EmployeeService_JSONClient.V10.bar.

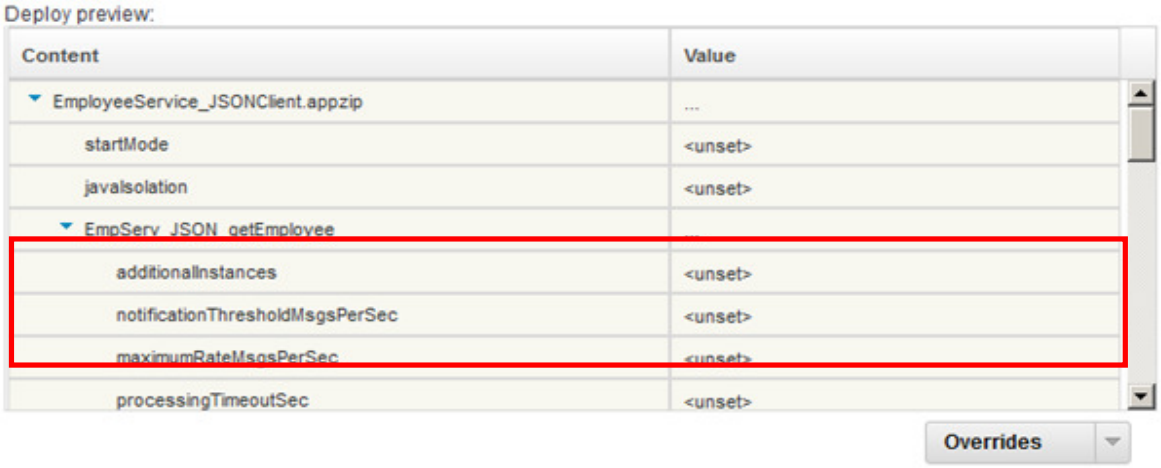
   Click Open.

4. In the ‘Deploy BAR File’ window you can see again the content of the BAR file expand EmpServ_JSON_getEmployee.

   In the ‘Value’ column you are presented with the properties that can be configured.

   <unset> refers to a property that has not been set. You can set the property using an override file.

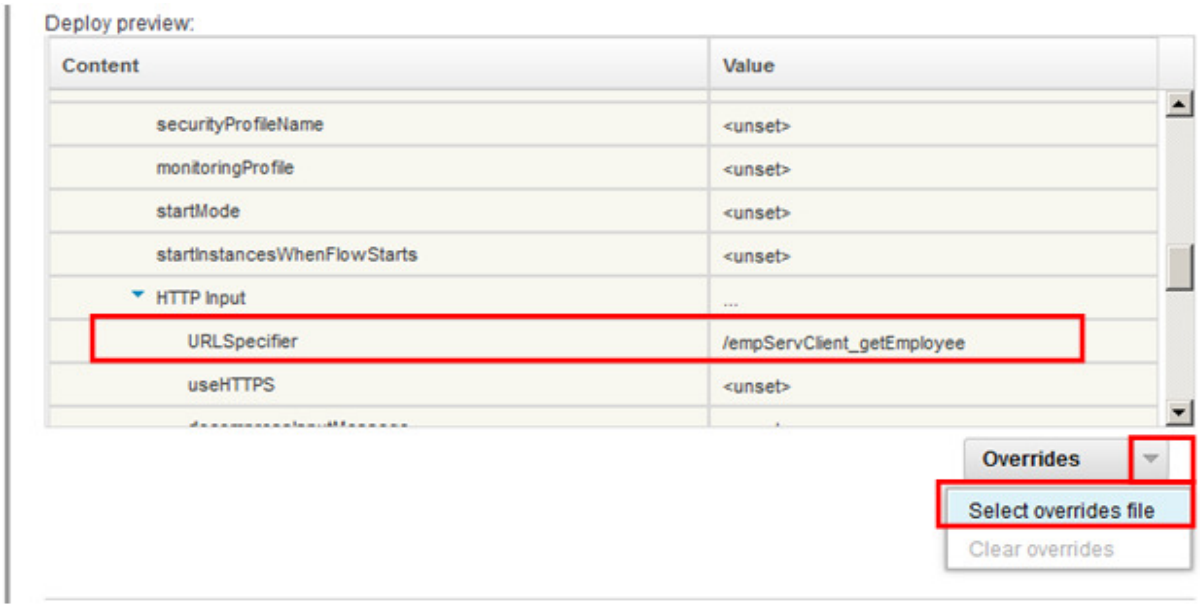
   You will see two of the properties which were present in the override file.



5. Scroll down and expand the HTTP Input node properties.

   The URLSpecifier is the third property that you will override in the next step.

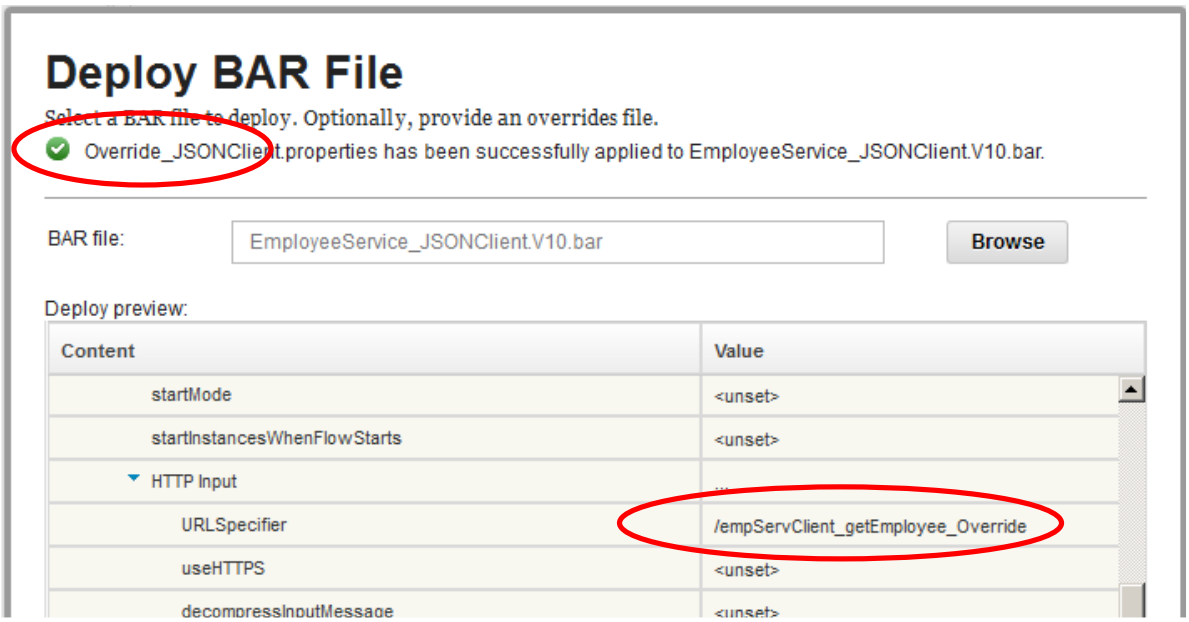
   From the context menu next to ‘Overrides’ click on ‘Select overrides file’.



6. Select 'Override_JSONClient.properties' from **C:\student10\Integration_service_JSONClient\solution** directory and click Open.

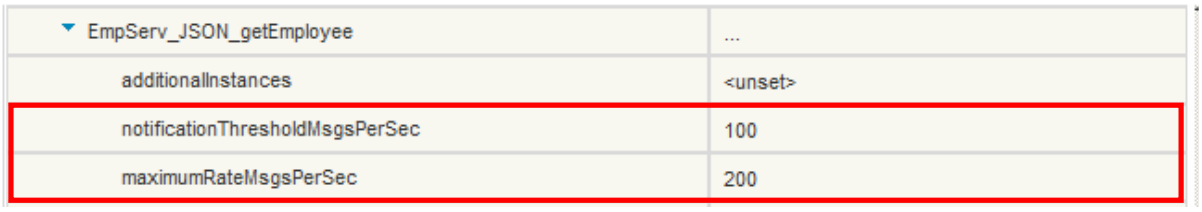
   The override file will be applied, and a message will be shown to indicate this.

   The URLSpecifier value has changed to /empServClient_getEmployee_Override

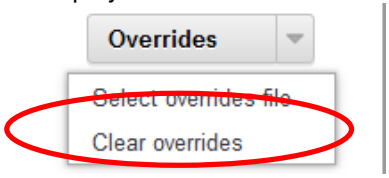


7. Scroll up to see the other two properties that were overridden.

   Now they their values have been set accordingly based on the values in the override file.



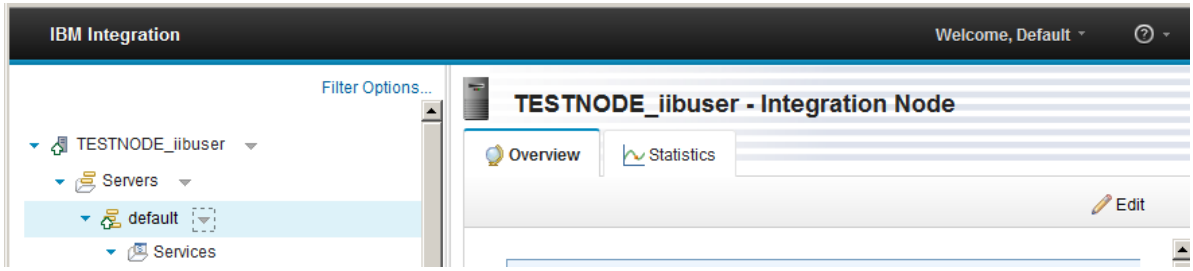
8. To restore the original property values from the BAR file, click the down arrow beside Overrides and click Clear overrides file and Cancel the deploy.



4. Using Web Browser interface without admin security

4.1 Administration

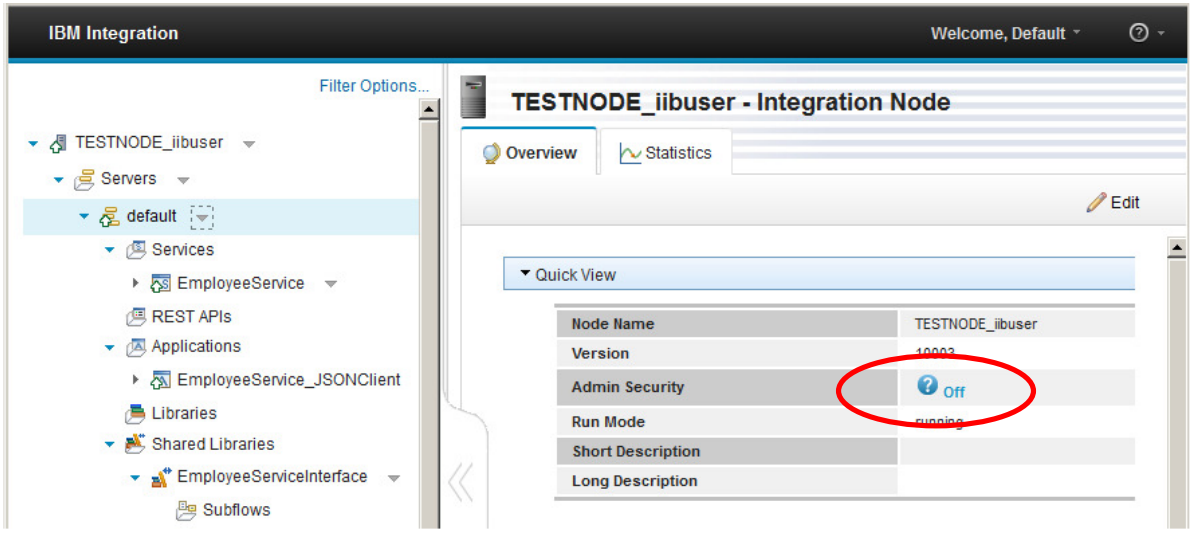
1. You may have noticed in the top right corner of your web browser that you are logged in as a 'Default' user.



2. Click TESTNODE_iibuser. In 'Quick View', you will see that 'Admin Security' is 'Off'.

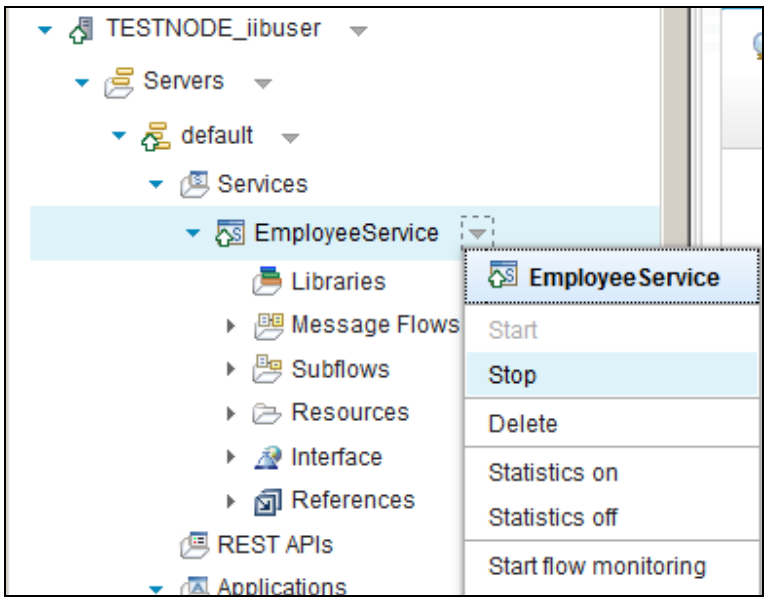
   In this scenario, the default user has full update access to all deployed broker resources.

   If you have enabled the webadmin http or https listener, and admin security is not active for the node, then any user will be able to access the Web UI facilities.



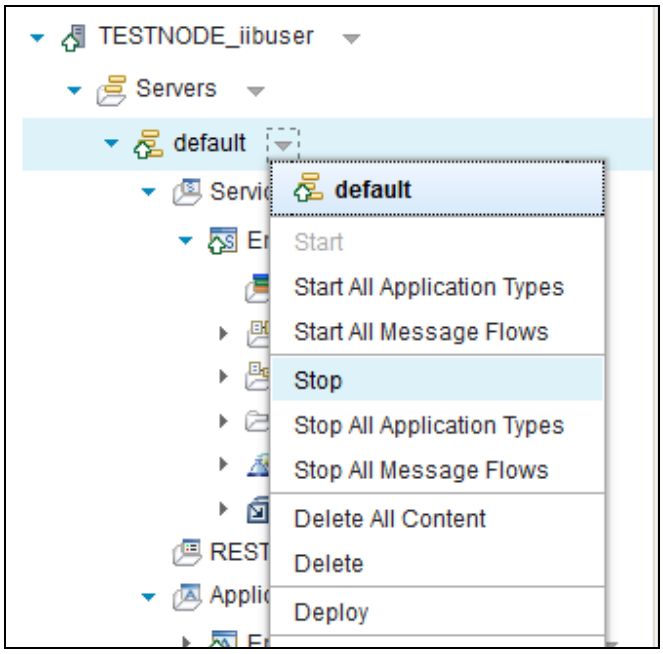
3. In the Servers section, expand **default**. Expand the deployed Services and Applications. You should see the applications you just deployed in the previous step.

   You can Start and Stop these services and applications using the context menu.



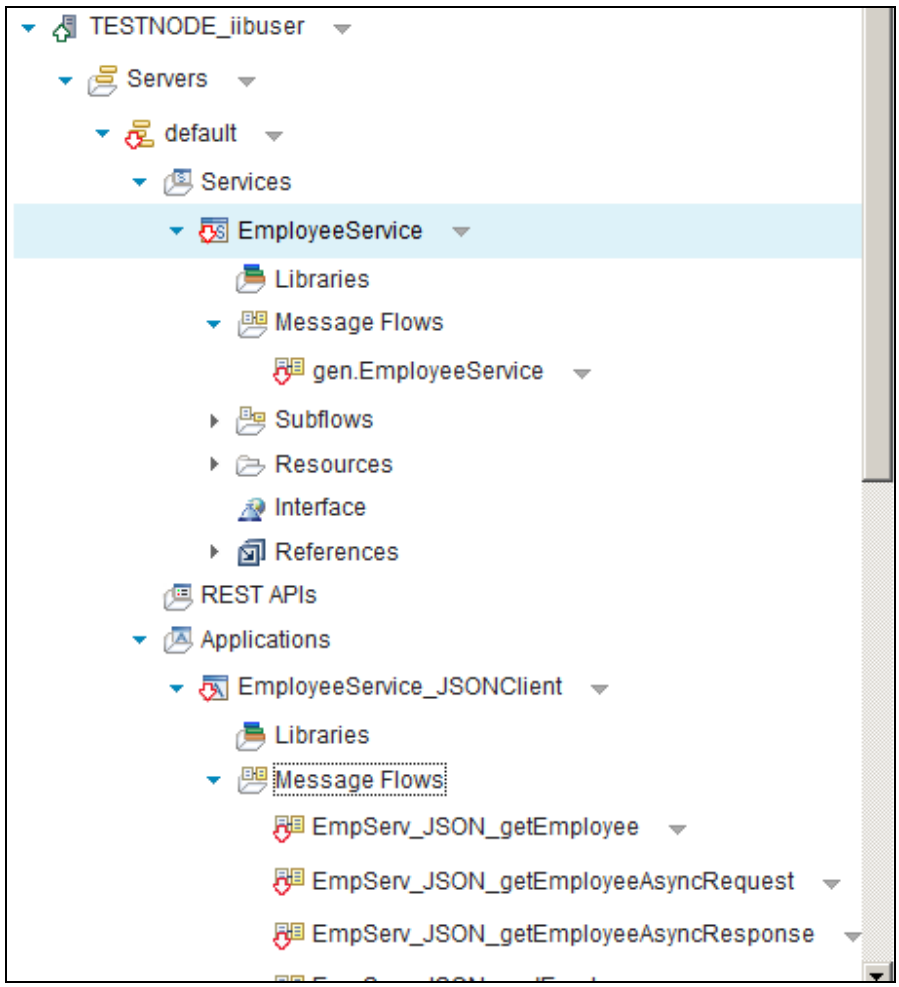
4. You can Start and Stop the server (eg. default) by using the context menu on the server. You can also Start/Stop all the Applications contained within a server, without stopping the server itself (Stop all Application Types).

   Click Stop.



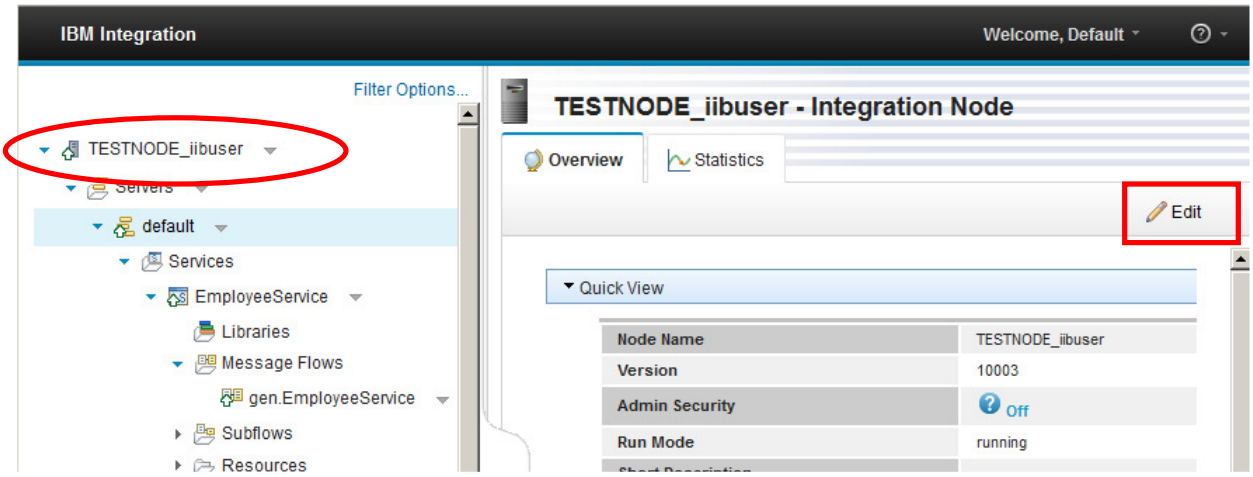
5. Wait for it to show stopped (red down arrow). Notice that the server will show as stopped, with the red "down arrow" showing against the server. Expanding each service and application will show that the individual message flows are also stopped.

   Select the context menu again and select Start. The server will start and show a green “up arrow”.

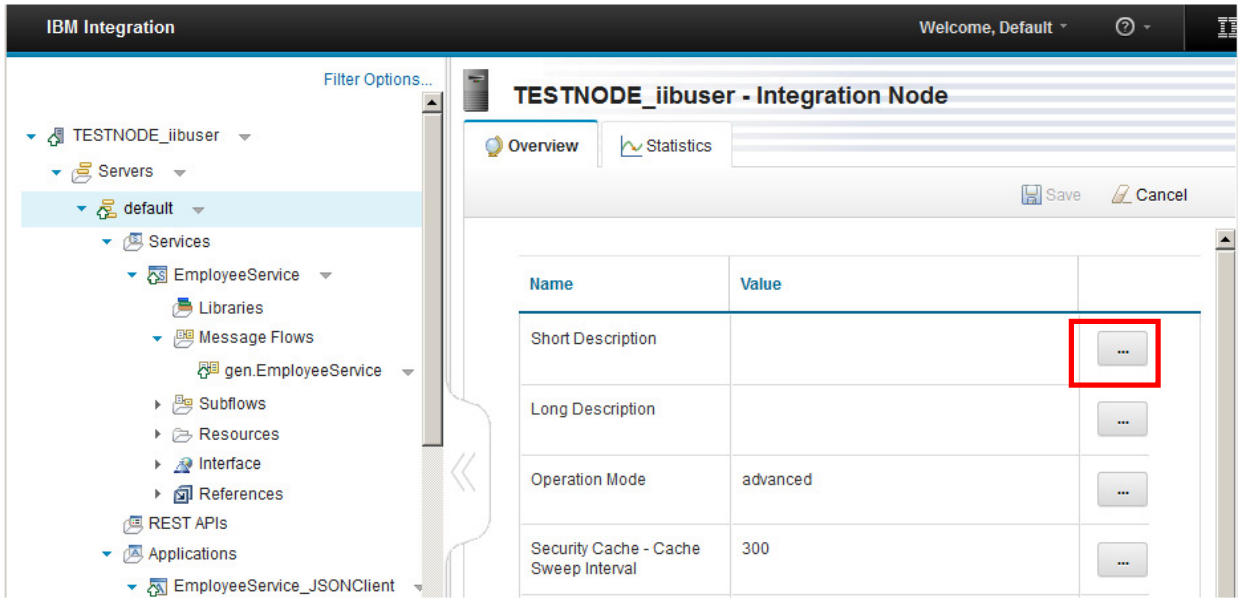


4.2 Integration Node and Server properties

1. Click TESTNODE_iibuser and then 'Edit':

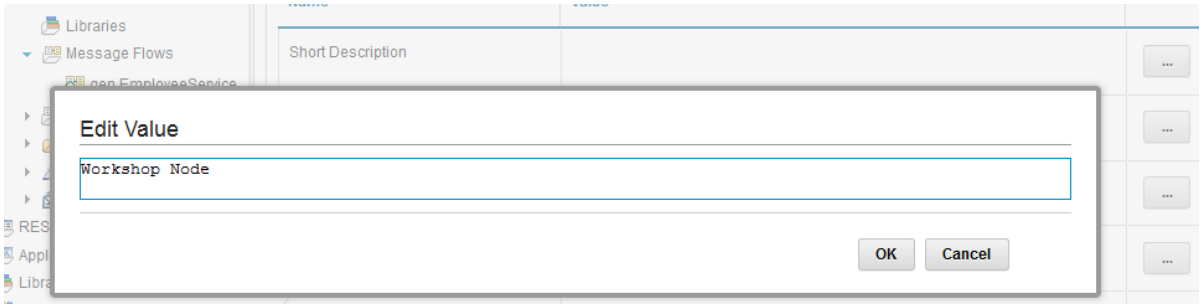


2. This will open a table with various Integration Node properties. You can change any of the properties by clicking the button in the far right column against each property, or by double-clicking the Value field.

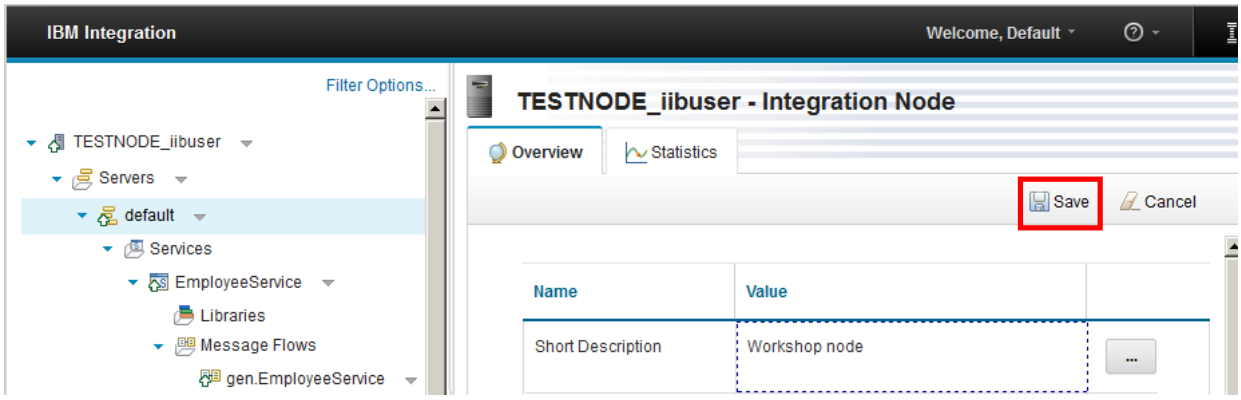


Click to change the 'Short Description' of the Integration Node.

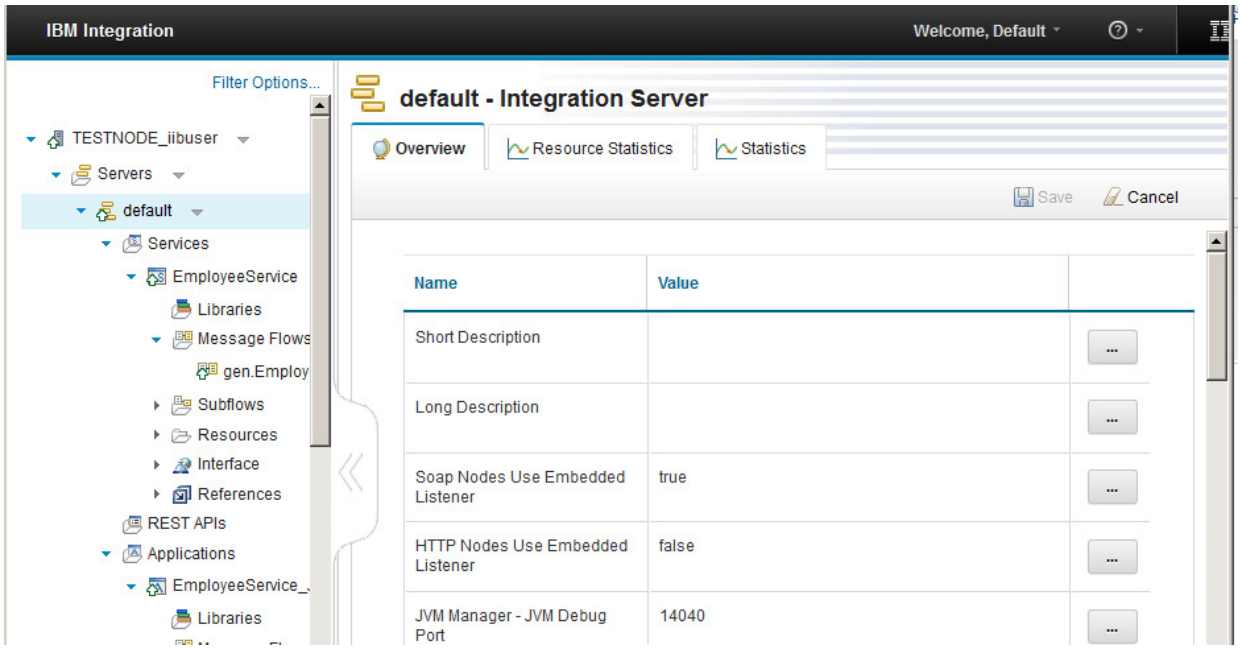
3. In the Value field type some text and click OK.



4. To save any changes that you have made to the Integration Node's properties, click the 'Save' button:



5. Similarly, you can edit the properties of an Integration Server. Click the required server (eg. **default**), and click Edit. Make required changes and click Save.



5. Authentication and authorisation for remote users

Administration security controls the rights of users to complete administrative tasks for an integration node and its resources.

IBM Integration Bus V10 provides the ability to control access to Integration node resources through the Web user interface and REST application programming interface (API). All the Nodes administration functions can be viewed and controlled, as well as all the functionality that in previous IIB releases was available through IBM Integration Explorer. Different web users can have different access rights across these functions, and access can be granted, denied or revoked quickly.

IBM Integration Bus V10 allows role-based security to be achieved by using one of two options:

- Security functions based on Integration Bus and WebSphere MQ;
- File-based security in IBM Integration Bus.

While the user has the option to choose how to implement the role-based security, based on their infrastructure, this guide will show the new functionality for IBM Integration Bus V10 - enabling file-based security.

The access authorities are defined against a set of user definitions which represent the available security roles. A role is a set of security permissions that control access to an integration node and its resources, and each web user account is associated with a particular role. The permissions are checked to determine a web user's authorization to perform tasks in the web user interface or the REST application programming interface (API). Each web user is then defined to use one or more of these security roles.

For the purposes of this lab, you will create several user roles. Then you will create web users and assign them to one of the defined roles:

1. iibAdmin1 IIB node administration access, **read-only**
2. iibAdmin2 IIB node administration access, **read, write** functions
3. iibAdmin3 IIB node administration access, **all** functionality
4. iibDev full access for developer - selected IIB nodes
5. iibQA full access for selected person - selected IIB nodes

5.1 Start the LDAP Server

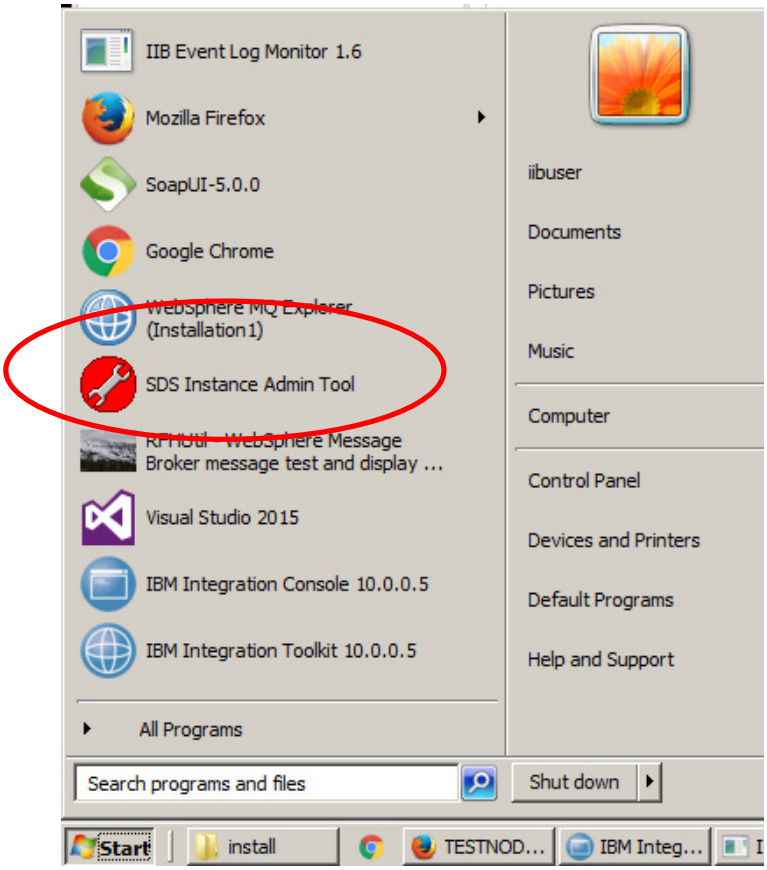
In this lab document, the LDAP is provided by IBM Security Directory Server (SDS). On the workshop system, IBM SDS is already installed and configured.

The LDAP database has already been populated with three admin users, admin1, admin2, admin3, and two development users, dev1, and dev2.

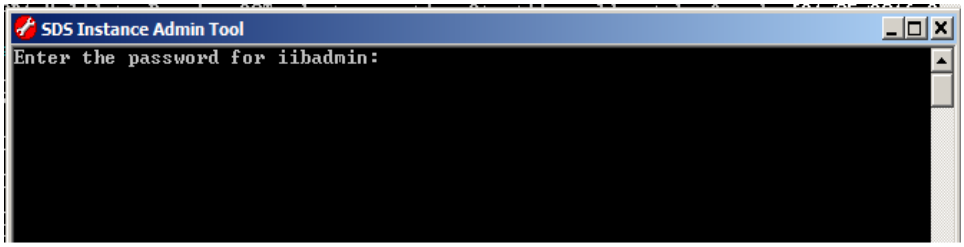
Start the SDS by performing the following steps.

1. SDS must be started by a Windows user with administrator access. **iibuser** does not have this access, but the icon on the Windows Start menu for **iibuser** is defined to "runas" **iibadmin**.

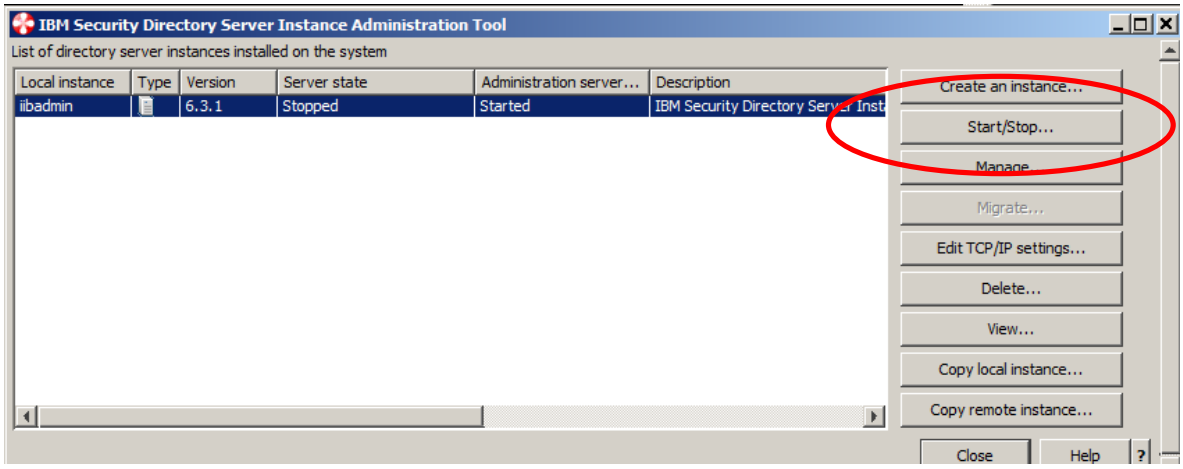
   From the Windows Start menu, click SDS Instance Admin Tool.



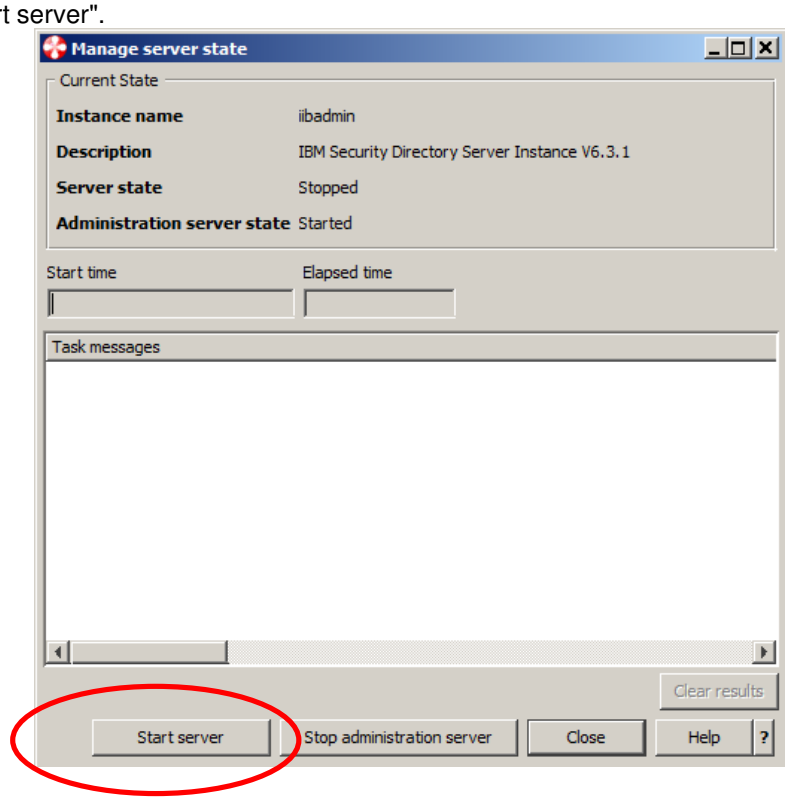
2. When requested to provide the password for **iibadmin**, enter "passw0rd".



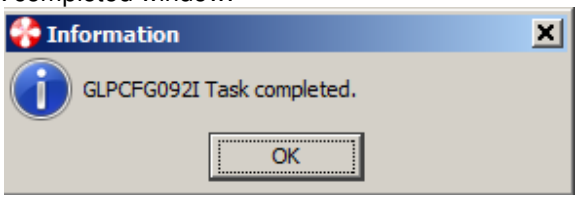
3. After a short time, the Instance Admin Tool will open. The "Server state" will be Stopped. Click Start/Stop to start the server.



4. Then click "Start server".



5. Click OK to close the Task completed window.



The SDS windows can be left in their current state for the rest of the lab.

5.2 Configure authentication with LDAP for TESTNODE

IIB v10 fixpack 4 provides the option to authenticate users in an LDAP database. This applies both to users of the web browser administration function, and to users of the IIB Toolkit, when connecting to a remote IIB node.

In this section, you will see how to configure the IIB node TESTNODE_iibuser to authenticate users with an LDAP database hosted by IBM Security Directory Server (SDS). On the workshop system, IBM SDS is already installed and configured.

The LDAP database has already been populated with three admin users, admin1, admin2, admin3, and two development users, dev1, and dev2.

6. Open an IIB Integration Console from the Windows Start menu.

7. The integration node has to be configured to enable it to connect to the LDAP server.

   In the IIB workshop system, this connection will be defined with username/password security. The connection will be configured for "plain text", so will use the default LDAP port of 389.

   Run the command:

```
mqsisetdbparms TESTNODE_iibuser
      -n ldap::LDAP
      -u cn=root
      -p passw0rd
```

   This command saves the LDAP connection credentials within the IIB node registry.

   Normally, this connection would be defined with SSL (the default port for LDAP with SSL is 636). However, this requires the definition of key and truststores - see Lab 9, Message Flow Security for more details.

8. Now define the precise LDAP connection string details that this IIB node will use for user authentication.

   Run the following command:

```
mqsichangeproperties TESTNODE_iibuser
-b webadmin
-o server
-n ldapAuthenticationUri
-v \"ldap://localhost:389/ou=users,ou=iib,o=ibm?uid\"
```

   Notes:
   1. The -n ldapAuthenticationUri parameter is new in IIB v10.0.0.4.
   2. The -v parameter is the value of the LDAP database connection, and the LDAP string that represents the location of the users for authentication checks.
   3. Because the LDAP string contains embedded commas, the string must be contained within quotes. The quotes must be preceded by the escape character \, hence the string is contained within the characters \".

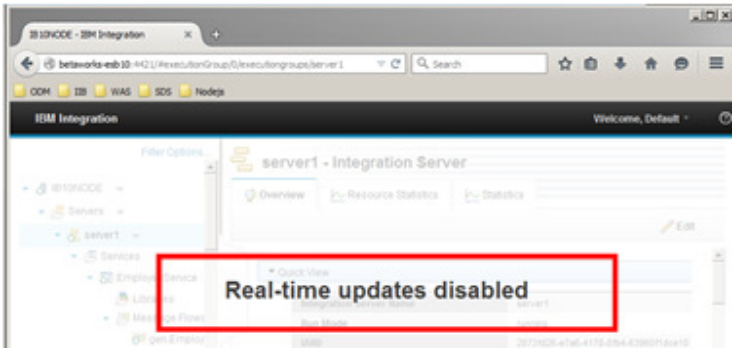
5.3 Activate administration security for TESTNODE_iibuser

1. Open an IIB Integration Console from the Windows start menu.

   Security can only be activated whilst the integration node is shutdown, so issue the following commands

   ```
   mqsistop TESTNODE_iibuser
   ```

   When you issue the **mqsistop,** you will see that the Browser page is greyed with a message indicating that “Real-time updates are disabled”.

2. In the Integration Console, confirm that administration security is inactive with the following command:

   ```
   mqsireportauthmode TESTNODE_iibuser
   ```

   You will receive the response:

   ```
   BIP8930I: Integration node name 'TESTNODE_iibuser'
    Administration security = 'inactive'
    Authorization mode = 'file'
   ```

   As expected the administration security is returned as "**inactive**".

   Since our Integration Node has been created automatically by the IIB Toolkit, without an associated Queue Manager, its default authorization mode is "**file**".

3. Turn on the administration security and change the authorization mode with the command:

```
mqsichangeauthmode TESTNODE_iibuser -s active -m file
```

(In this case, the "-m file" option is not required, since the authorization mode is already set to "file"; we have shown it here for completeness).

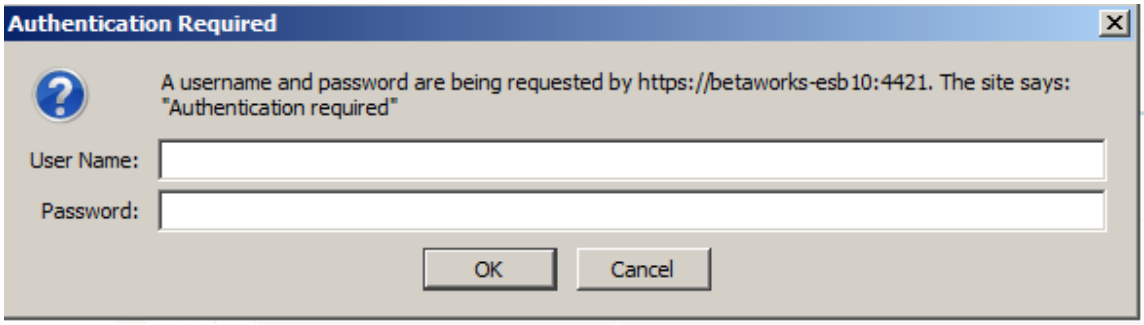
You should see the response:

```
BIP8071I: Successful command completion.
```

Restart the Node with the following command:

```
mqsistart TESTNODE_iibuser
```

4. After a short while, on the web browser, you will now see a user/password challenge.



Before you can continue, we must configure security so you are able to sign in, so continue with the next step.

5.4 Define administration roles and set file-based permissions

You will now create administration roles and grant administration authorities to these roles. These will be referenced by the Integration Bus web user definitions.

Three levels of authorization are supported for IBM Integration Bus administration security: **read**, **write**, and **execute**. You can assign permissions to a role (user) by specifying the type of permission followed by a plus (+) to grant permissions, or a minus (-) to revoke permissions.

User authorisations
The following table shows the file-based authorities that are required for different types of users in this lab guide:

Role	Authority
iibAdmin1	read
iibAdmin2	read, write
iibAdmin3	read, write, execute (all)
iibDev	all
iibQA	all

These authorities are related to actions as follows:

- **read** – view resources
- **write** – view resources, create Integration Servers and modify their settings
- **execute** – start, stop, deploy and modifying resources
- **all** - all authorities

LDAP user definitions
In the pre-built workshop system, the following users have been defined in the LDAP server (IBM Security Directory Server). The passwords have been set to unique values, to ensure that authentication with different users/passwords do not get confused.

User	Password (in LDAP)
admin1	admin1
admin2	admin2
admin3	admin3
dev1	dev1
dev2	dev2

1. Check if there are any roles defined on the Integration Node by running the command:

```
mqsireportfileauth TESTNODE_iibuser -l
```

   You should see the response:

```
BIP8071I: Successful command completion.
```

   No defined roles have been returned.

2. Create the role **iibAdmin1** by running the command:

```
mqsichangefileauth TESTNODE_iibuser
                        -r iibAdmin1
                        -p read+
```

   Response:

```
BIP8071I: Successful command completion.
```

3. Create **iibAdmin2** by running the command:

```
mqsichangefileauth TESTNODE_iibuser
                        -r iibAdmin2
                        -p read+,write+
```

   Response: `BIP8071I: Successful command completion.`

4. Create **iibAdmin3** by running the command:

```
mqsichangefileauth TESTNODE_iibuser
                        -r iibAdmin3
                        -p all+
```

   Response: `BIP8071I: Successful command completion.`

5. Re-run the command for displaying any defined roles:

```
mqsireportfileauth TESTNODE_iibuser -l
```

The returned response should be as below:

```
BIP8931I: Role = 'iibAdmin1', Resource = '', Permissions = 'read+,write-,execute-'
BIP8931I: Role = 'iibAdmin2', Resource = '', Permissions = 'read+,write+,execute-'
BIP8931I: Role = 'iibAdmin3', Resource = '', Permissions = 'read+,write+,execute+'

BIP8071I: Successful command completion.
```

5.5 Define the web users for TESTNODE_iibuser

1. Define an Integration Bus web user for read-only access. This user will be able to see what applications are deployed, but will not be able to control the status of these applications.

   In an Integration Bus Command Console, run the command

   ```
   mqsiwebuseradmin TESTNODE_iibuser
                            -c
                            -u admin1
                            -r iibAdmin1
                            -x
   ```

   This command will define a new web user, admin1. The user will have the security profile defined by the associated role, which in this case will mean that the user can only view the broker and any deployed applications.

   The "-x" parameter means that no password will be stored locally (in the IIB node); all requests for authentication of this user will be sent to the connected LDAP system.

2. Define an Integration Bus web user for read/write access. This user will be able to see what applications are deployed, and will have administration privileges to change properties on the Integration Node and Integration Server.

   ```
   mqsiwebuseradmin TESTNODE_iibuser
                            -c
                            -u admin2
                            -r iibAdmin2
                            -x
   ```

   This command will define a new web user, admin2. The user will have the security profile defined by the associated role, which in this case will mean that the user will be able to view the broker and execution groups, and edit their properties. Also, the user will be able to view the deployed resources.

3. Define an Integration Bus web user for all access. This user will be able to see what applications are deployed, and will be able to control completely the resources (start/stop, etc).

   ```
   mqsiwebuseradmin TESTNODE_iibuser
                            -c
                            -u admin3
                            -r iibAdmin3
                            -x
   ```

   This command will define a new web user, admin3. The user will have the security profile defined by the associated role, which in this case will mean that the user will be able to view the broker and execution groups, and any deployed applications, as well as control their status.

4. Display the newly-defined web users by running the command

```
mqsiwebuseradmin TESTNODE_iibuser -l
```

The response received should be as below:

```
BIP2837I: Web user 'admin1' is defined as having a role of 'iibAdmin1'.
          This user has no local password.
BIP2837I: Web user 'admin2' is defined as having a role of 'iibAdmin2'.
          This user has no local password.
BIP2837I: Web user 'admin3' is defined as having a role of 'iibAdmin3'.
          This user has no local password.

BIP8071I: Successful command completion.
```

6. Using the Web Browser interface with admin security

6.1 User with read-only access

Now that security has been configured for Web Admin users, we will login as the admin1 user, which has read-only access, and see what the browser interface offers for someone who can only view the Integration Node.

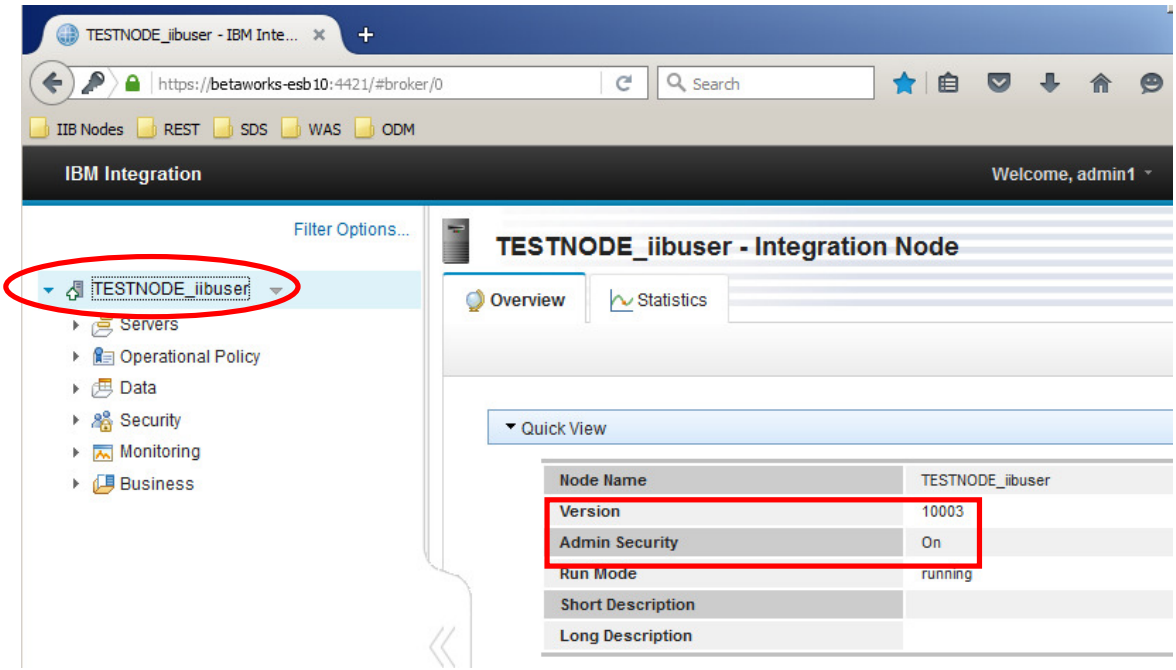
1. You should have your Firefox web browser window still open with the 'IBM Integration' log in page.

   Login with the userid '**admin1**' (the password in the LDAP database is **admin1**).

2. This user has read-only access to the node.

   Click TESTNODE_iibuser in the navigator.

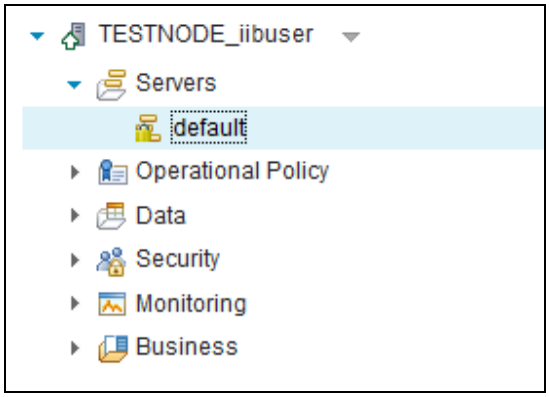
   Note that the Quick View will now show you that Admin Security is active.



.

3. Expand the 'Servers' category, by clicking the twisty.

   You are presented with the available servers but you are not able to view any of the resources. Yes, you have guessed correctly – although you gave the role **`iibAdmin1`** 'read' authorities, you specified this at a Node level. This allows more 'granular' approach by authorization for individual servers completed with a separate command (shown in the next step).



4. The IIB file-based authorization allows you to change the roles' permissions without a restart of the Integration Node – the changes are picked up dynamically

   In the Integration Console, run the command:

```
mqsichangefileauth TESTNODE_iibuser
                               -e default
                               -r iibAdmin1
                               -p read+
```

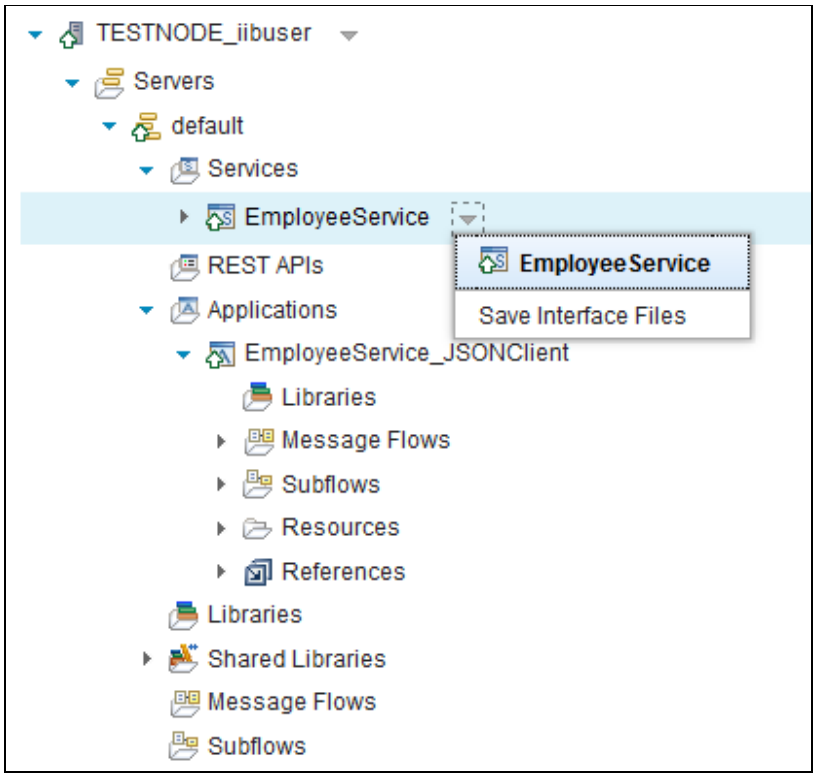
   Response: **`BIP8071I: Successful command completion.`**

   The Integration Server has been specified with the parameter **`-e default,`** which means that now you are applying the permissions for the integration server named **default**.

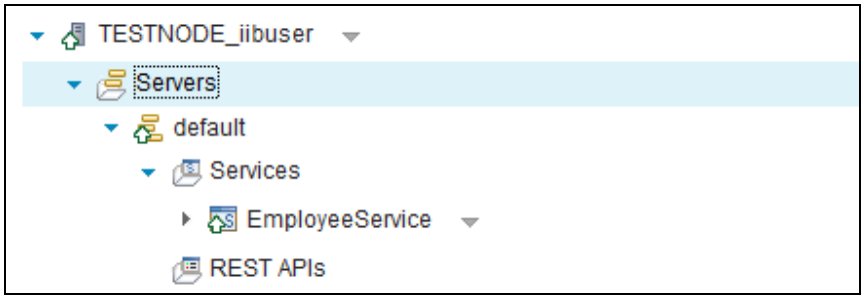
5. Back in the Browser, refresh the page and log in again as user ‘admin1’.

   Expand the node and server.

   You can now view the resources on the Integration Server. Notice that although you are able to expand the default server resources folders you do not have permission to start, stop or other actions to the deployed artefacts. (For services, you are permitted to save the interface files).



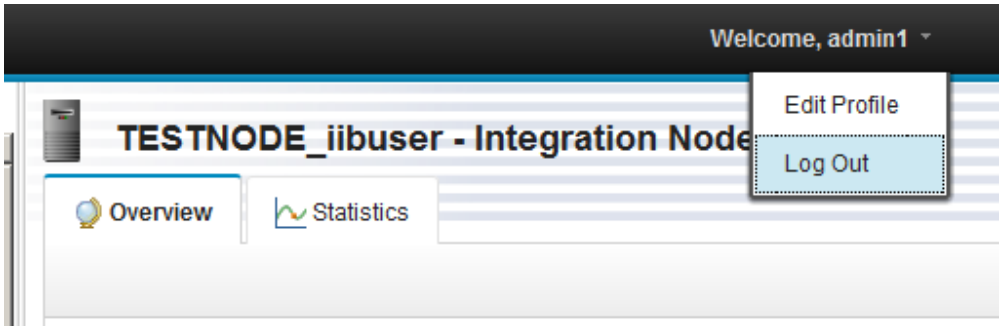
6. The Web UI provides facilities to create and delete IIB servers. However, for admin1, with only read access, the context menu arrow on Servers is not shown, and no actions are permitted (it is permitted to save the service interface files, hence the arrow on EmployeeService).



6.2 User with write access

Now login as the admin2 user which is the 'read/write' user. This user has functionality for viewing resources, editing properties for Integration Node and Integration Server and for creating Integration servers.

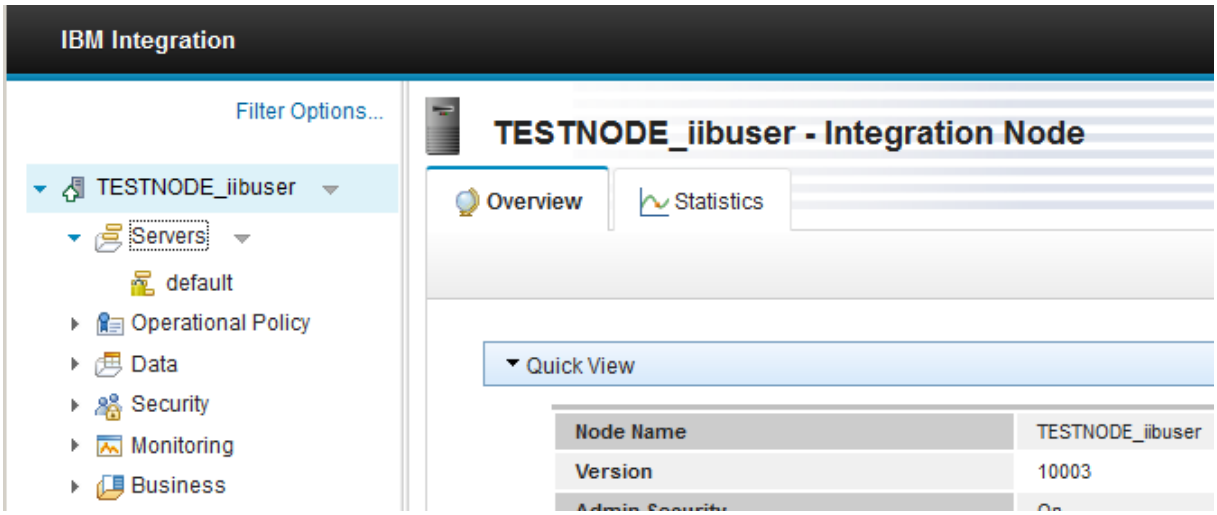
1. Click the pull-down beside the Welcome, admin1 banner and select Log Out.



2. Log in as **admin2** (password is admin2).

3. Expand 'Servers' and you will see that although you can view the available servers, no deployed resources can be seen.

   This is because admin2 has read/write access for TESTNODE_iibuser, but not for any individual servers.



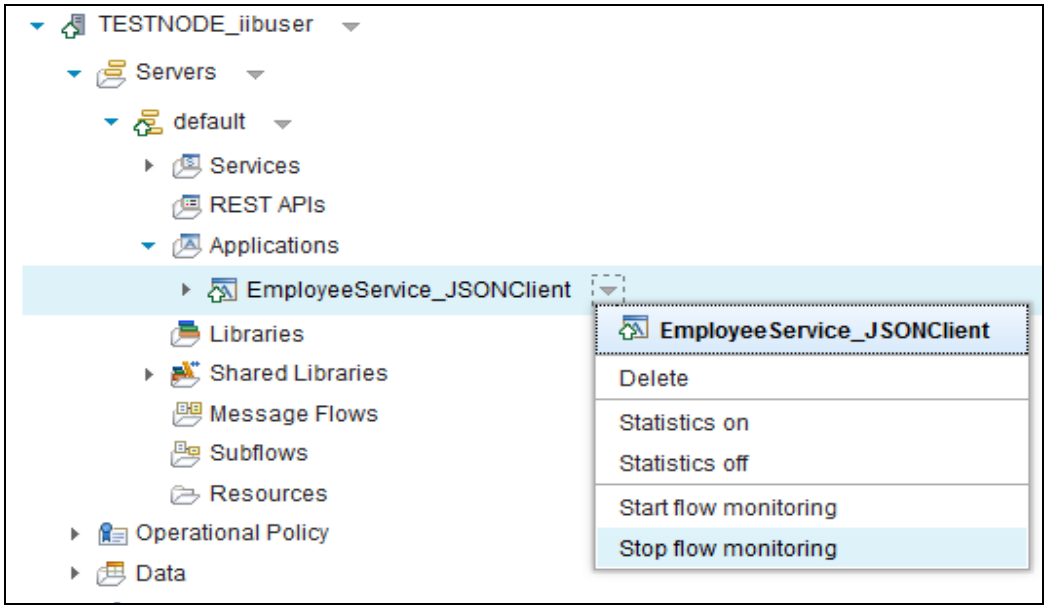
4. In the Integration Console, run the command:

```
mqsichangefileauth TESTNODE_iibuser
                   -e default
                   -r iibAdmin2
                   -p read+,write+
```

This will allow this role to view the resources on **default**, and create and modify the integration server properties.

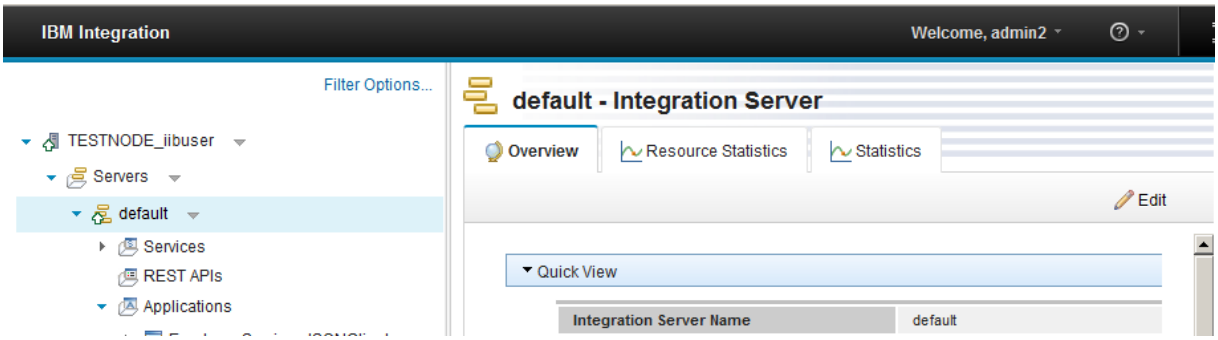
5. In the Web UI, refresh the login for **admin2** (F5 or refresh button).

admin2 is now able to perform certain actions on deployed resources. "write+" access permits the control of statistics and flow monitoring. It also permits the user to delete the deployed resource, so care should be taken when setting this permission.

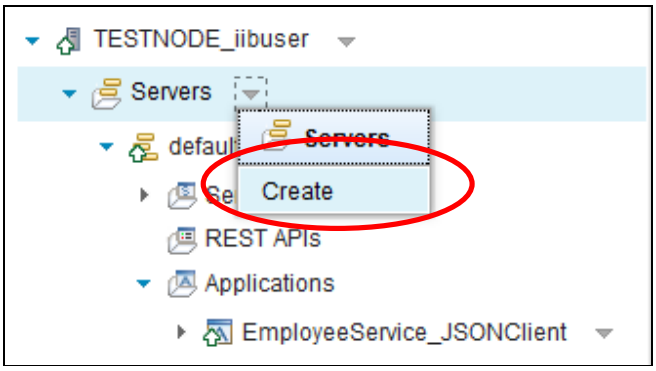


6. Highlight the **default** server. You will see there is an 'Edit' button (which was not available for 'admin1').

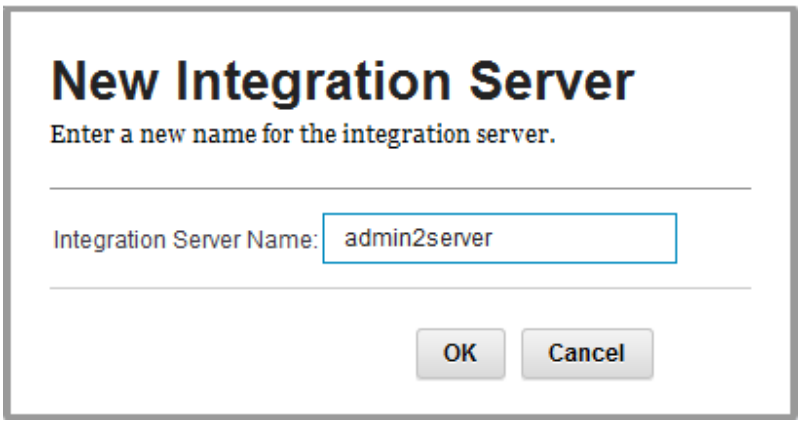
The table with the server's properties is opened and the user 'admin2' has authorities to change its settings. You will not make any changes here, so when finished reviewing, click 'Cancel'.



7. The admin2 user has Write permission on the IIB node. This permits the user to create a new server. Click the arrow on the right of ‘Servers’ and then ‘Create’:

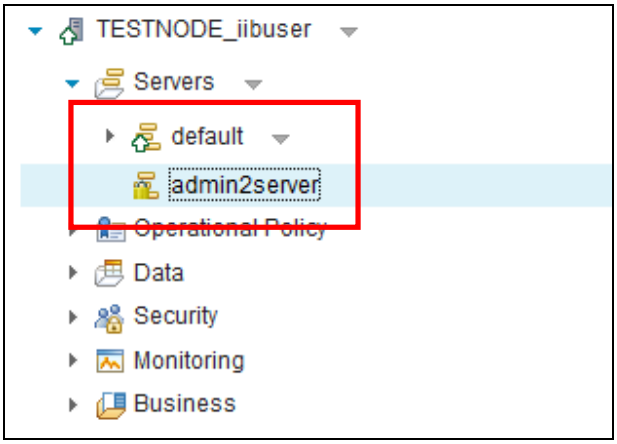


8. In the dialog enter ‘admin2server’ and click ‘OK’:



9. Shortly, you will see that a new Integration server has been created. The user **admin2** has 'Write' authorities on TESTNODE_iibuser, which allows the creation of new servers on the Integration Node. However, even though this user has created the Integration Server, the administrator has to provide additional authorization to permit admin2 to perform any actions on the new server such us start/stop.

   You may need to refresh the web UI (F5) to see the new server.



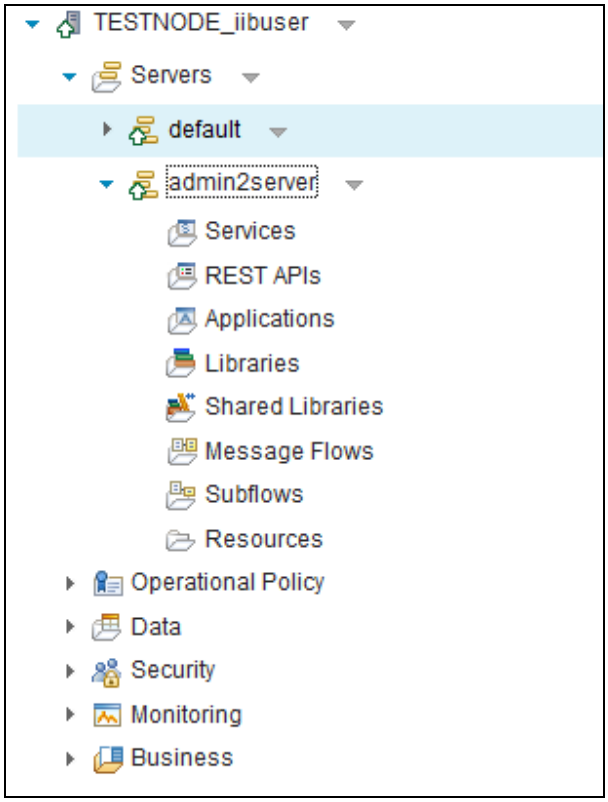
10. In the Integration Console run the command;

```
mqsichangefileauth TESTNODE_iibuser
                   -e admin2server
                   -r iibAdmin2
                   -p read+,write+
```

   This sets the permissions for this role on the 'admin2server' server as read and write.

11. Refresh the Browser window again, and log in as ‘admin2’ again (you may be automatically logged in after the refresh).

    Now, the web user is able to view the newly created server and its resources (although no resources have been deployed at this point).



12. Log out the user **admin2.**

6.3 The Web Admin interface for a user with ‘all’ access

You will now login as the **admin3** web user which has full authorities. This user has full functionality for the resources including stop/ start deployment of applications and start/stop statistics.

1. Although admin3 has full authorities for the resources on the node, the administrator still has to give permissions to the role, to which the web user is aligned for a particular Integration Server.

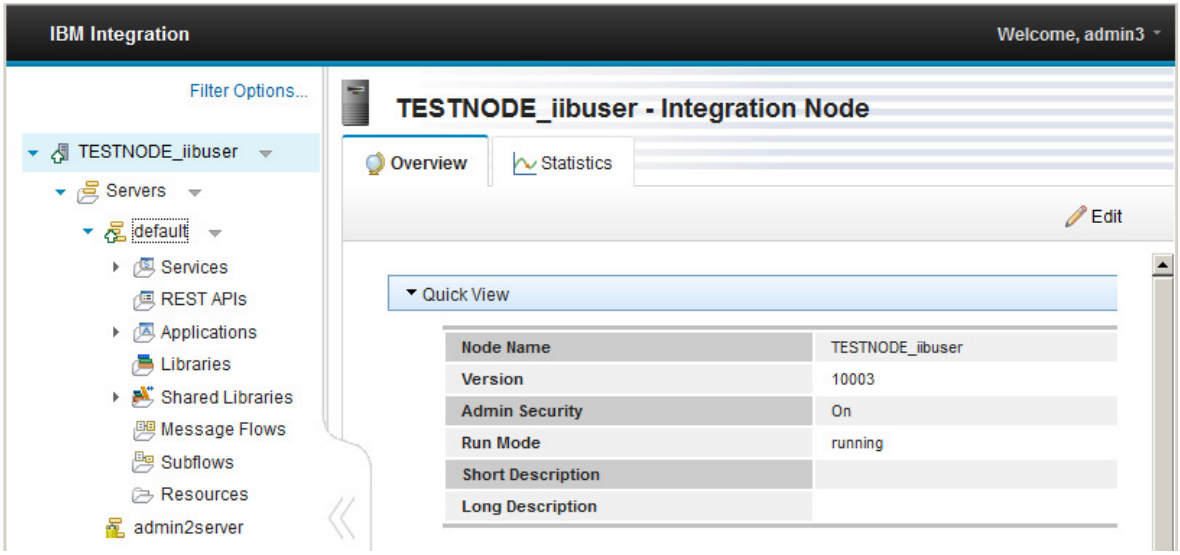
   Run the command:

   ```
   mqsichangefileauth TESTNODE_iibuser
                              -e default
                              -r iibAdmin3
                              -p all+
   ```

   Response: `BIP8071I: Successful command completion.`

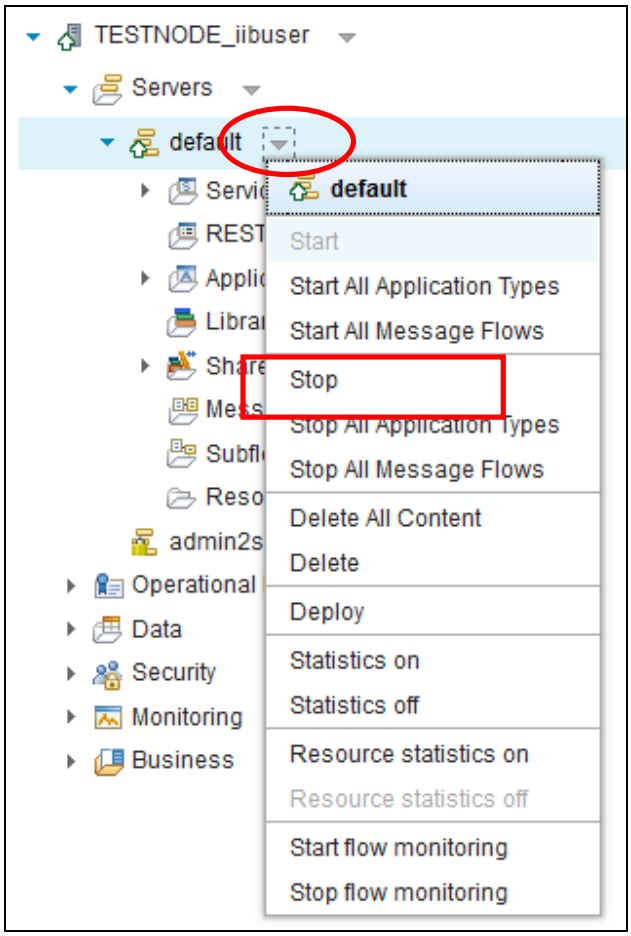
2. IN the web browser, login to the IIB node as ‘**admin3**’ (password is **admin3**):

3. Expand the ‘Servers’ group, then **default**.

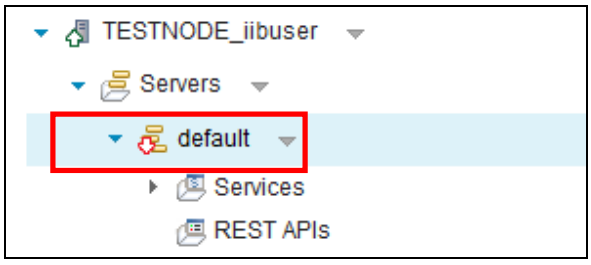


4. Click the drop-down menu next to the **default** server. You will see that the full list of available actions is available to this user.

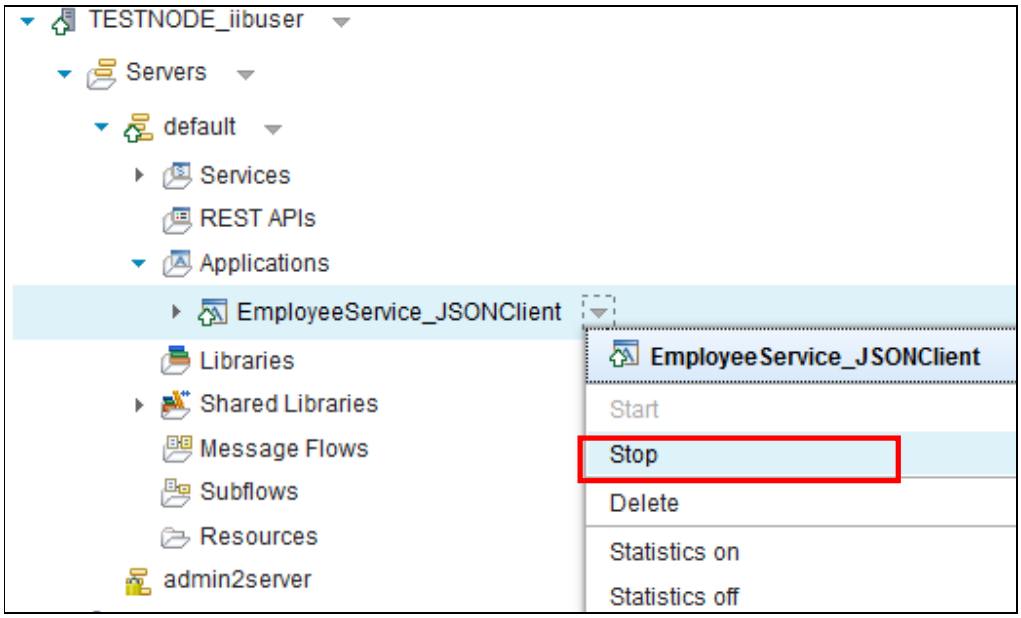
   Click 'Stop' to stop the Integration Server.



5. After a few seconds you will see that the integration server has stopped, shown with the red arrow pointing down. Start the server again.

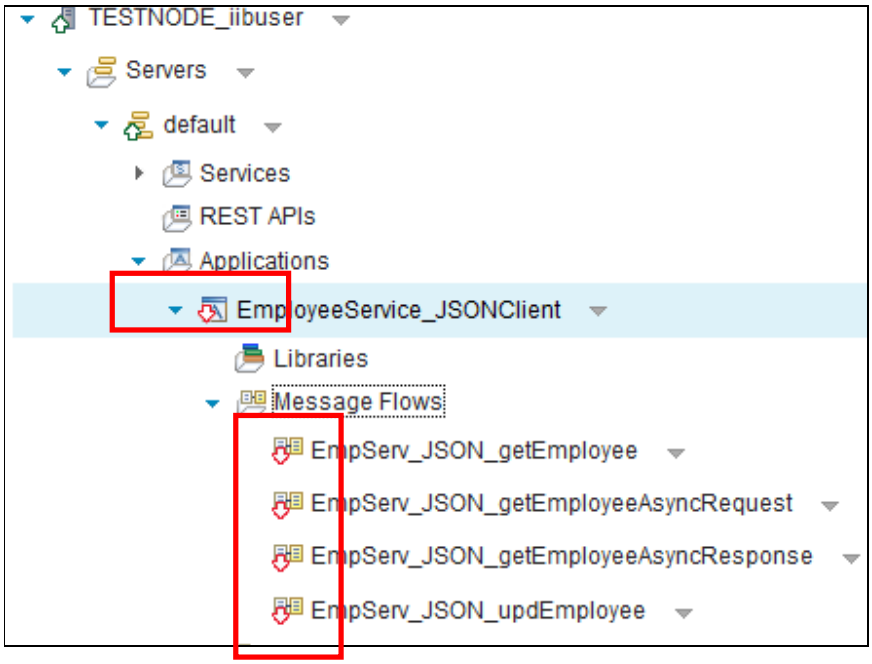


6. When the Integration Server starts, expand the 'Applications' group and then click on the twisty next to the 'EmployeeService_JSONClient' application. From the context menu, click Stop.

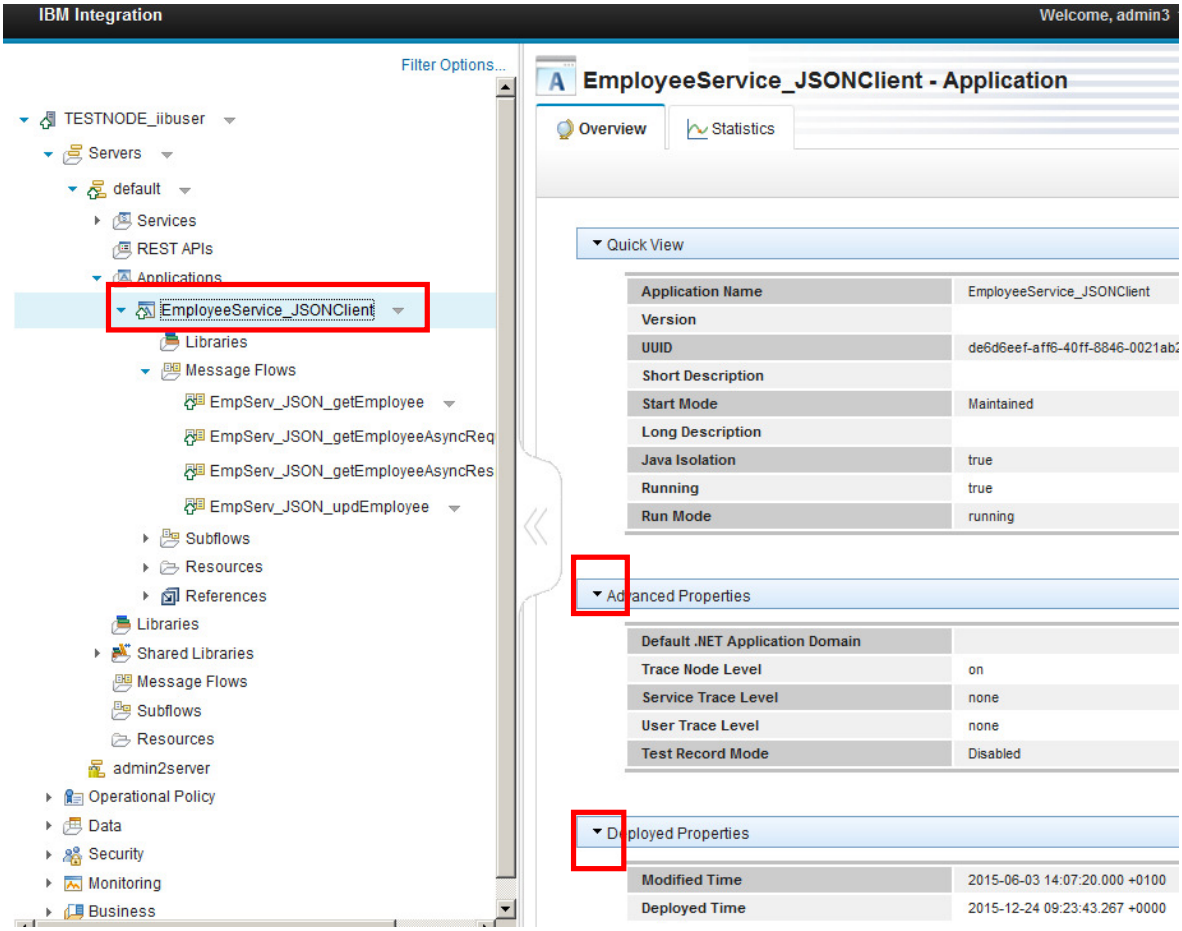


7. The application and all its flows have been stopped, shown with red arrows pointing down.

   Restart the application when ready.



8. When the application has been started, click on ‘EmployeeService_JSONClient’. On the right, expand ‘Advanced Properties’ and ‘Deployed Properties’



The Quick View panel displays important information about the application such as its name, UUID, Run Mode.

Advanced Properties and Deployed Properties show more detailed information.

9. Log out user **admin3.**

6.4 Administration of Operational Policy

In this part of the lab you will explore the administration of MQEndpoint Policy from users with different permissions based on the roles they have been assigned to.

1. In the Integration Console, navigate to **c:\student10\webadmin\Install** and type the following command:

```
mqsicreatepolicy TESTNODE_iibuser
                 -t MQEndpoint
                 -l WebAdminPolicy
                 -f MQEndpointSample.xml
```

The command references a provided MQEndpoint configuration policy file MQEndpointSample.xml, which has been provided for you.

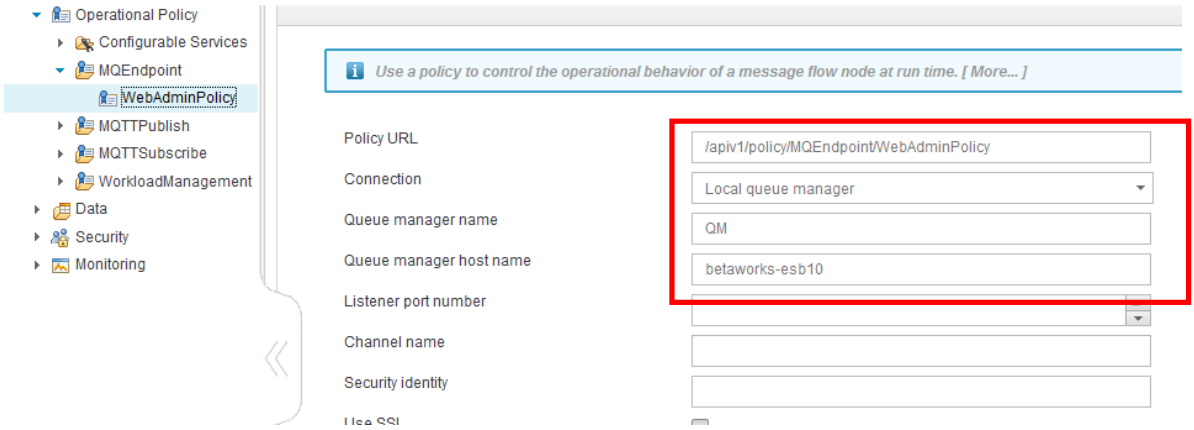
The result you see should be as below:

Response: **`BIP8071I: Successful command completion.`**

2. Log in to the Web UI as **admin1** (password=admin1).

Under TESTNODE_iibuser, expand Operational Policy → MQEndpoint and click on the newly created policy.

This user has 'read-only' permission and is only able to view the defined values.

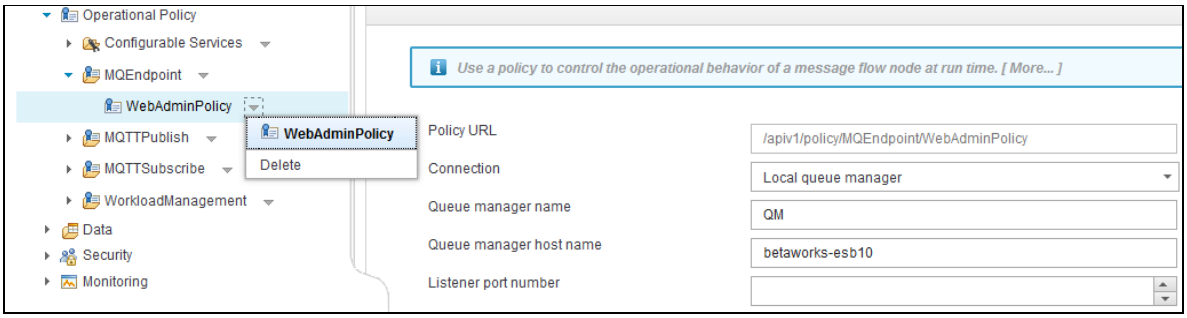


The values specified have been taken from the policy configuration file.

3. Log in to the Web UI as admin3 (password=admin3).

   Navigate to the newly created policy and click on it..

   This user has 'all' permissions and is able to re-configure the policy details and to delete it.



4. In the Integration Console run the command from step 1, but this time use the flag –i to specify a user, their password and hostname.

```
mqsicreatepolicy TESTNODE_iibuser
                 -t MQEndpoint
                 -i tcp:\\admin3:admin3@betaworks-esb10
                 -l admin3_policy
                 -f MQEndpointSample.xml
```

5. In the command console a message will be returned (`BIP1921S`) notifying you that the node cannot be reached (if you are using the workshop VMware image supplied).

```
C:\student10\webadmin\install>mqsicreatepolicy TESTNODE_iibuser -t
MQEndpoint -i tcp:\\admin3:admin3@betaworks-esb10 -l WebAdminPolicy -f
MQEndpointSample.xml

BIP1921S: The integration node cannot be reached. Check that the
integration node is running. Check that the TCP/IP address
'tcp:\\admin3:admin3@betaworks-esb10' matches the address of the machine
where the integration node is running, and that port '4414' matches the web
administration port that is configured for the integration node. The
specific error text is The IP address tcp:\\admin3:admin3@betaworks-esb10
or port number 4414 have invalid syntax.  Check that the values supplied
are correct.
```

   The reason this message is returned is because on the TESTNODE the SSL has been enabled. If you would like multiple users to connect to a remote integration node that has SSL enabled on its web administration port to execute a command, you will need to use **`.broker`** file.

   This is not subject of this lab. Please refer to IBM Knowledge Center for more information.

7. Integration Toolkit Authorization

The defined Web users' authorities also apply for a remote connection of an Integration Toolkit to existing Integration Node.

1. First, create a new role for developer access to an IIB node. In this lab, developers will be given full access to all resources in specified node, but will only be granted access to specific nodes for development and unit testing.

   In a Command Console, run the commands:

```
mqsichangefileauth TESTNODE_iibuser
      -r iibDev
      -p all+

mqsichangefileauth TESTNODE_iibuser
      -e default
      -r iibDev
      -p all+
```

2. Validate the new role.

   Run the command:

```
mqsireportfileauth TESTNODE_iibuser -l
```

   which should show:

```
BIP8931I: Role = 'iibDev', Resource = 'default', Permissions =
                  'read+,write+,execute+'
```

   Run the command:

```
mqsireportfileauth TESTNODE_iibuser -e default -l
```

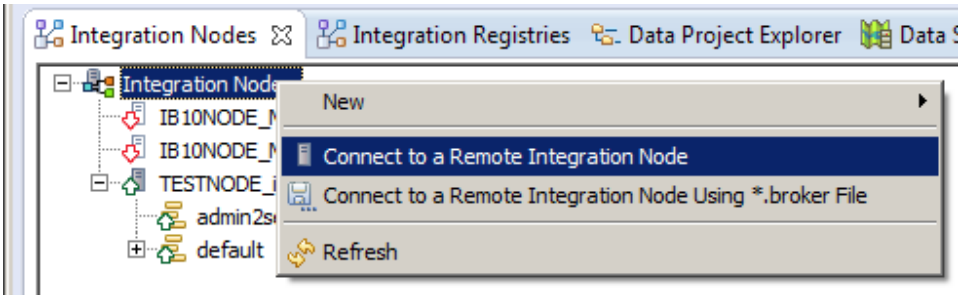
   which should show:

```
BIP8931I: Role = 'iibDev', Resource = '', Permissions =
                  'read+,write+,execute+'
```

3. Create a new IIB user, dev1, with role iibDev.

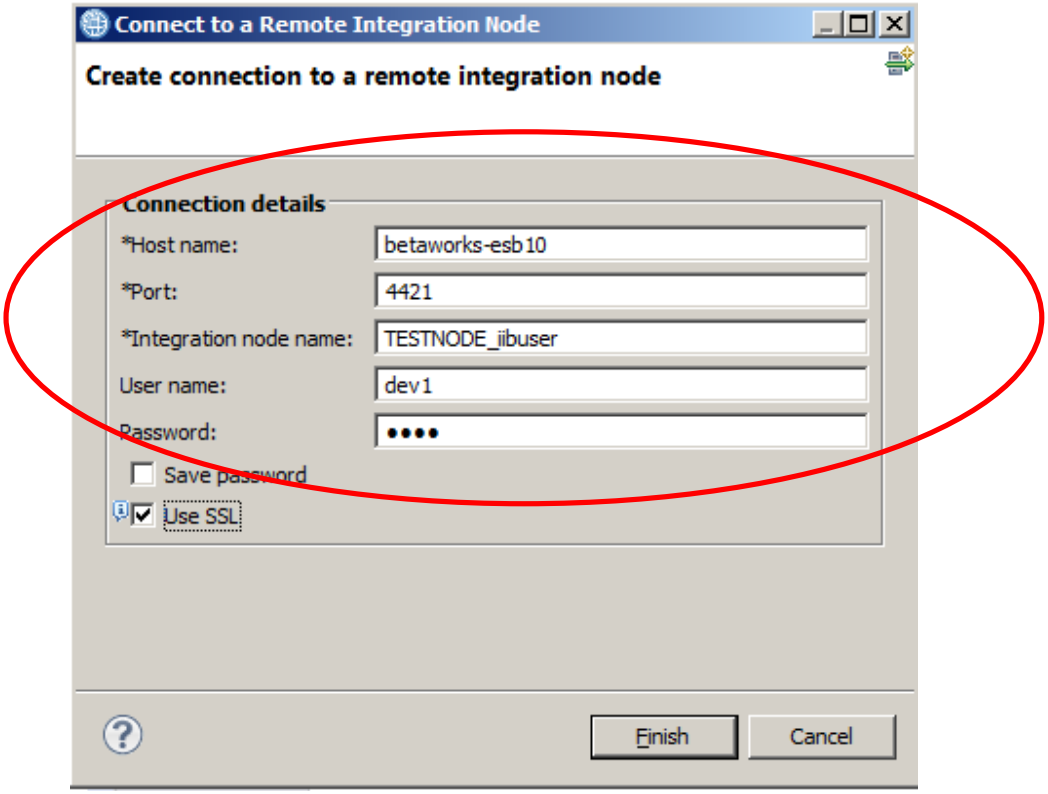
```
mqsiwebuseradmin TESTNODE_iibuser
      -c
      -u dev1
      -r iibDev
      -x
```

4. Create a remote connection to TESTNODE_iibuser from the Integration Toolkit.



5. Connect as ‘dev1’, providing the connection details as below:

Host: betaworks-esb10 (if using the workshop VM system)
Port: 4421
Integration Node: TESTNODE_iibuser
Use SSL: ticked

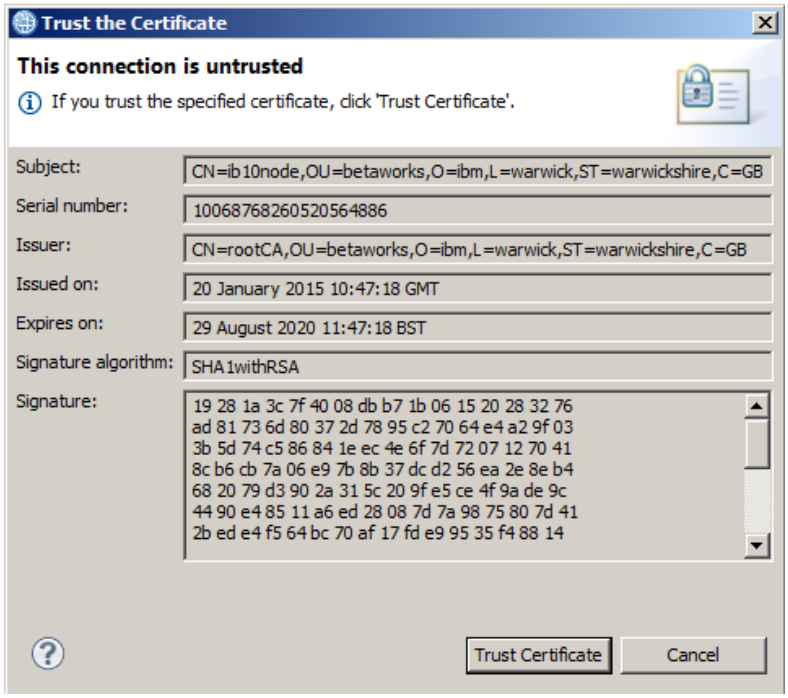


6. Click ‘Finish’.

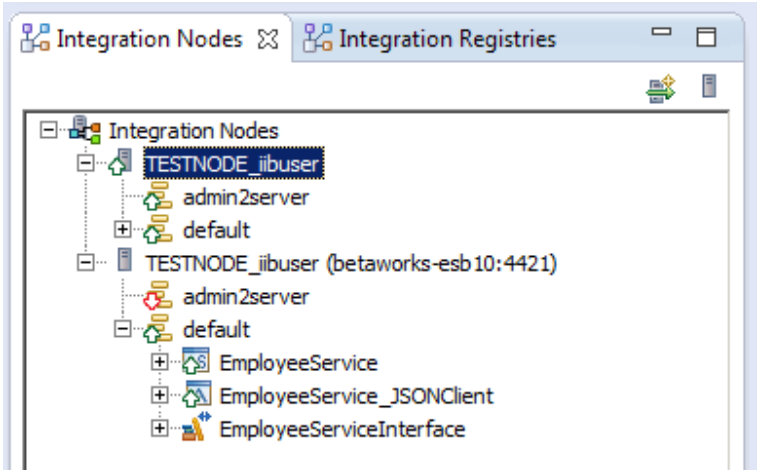
You may see a progress information window to which you may have to respond before continuing.

7. You will be presented with a dialog, where you will need to confirm the Trust certificate.

   Click on 'Trust Certificate' to continue.

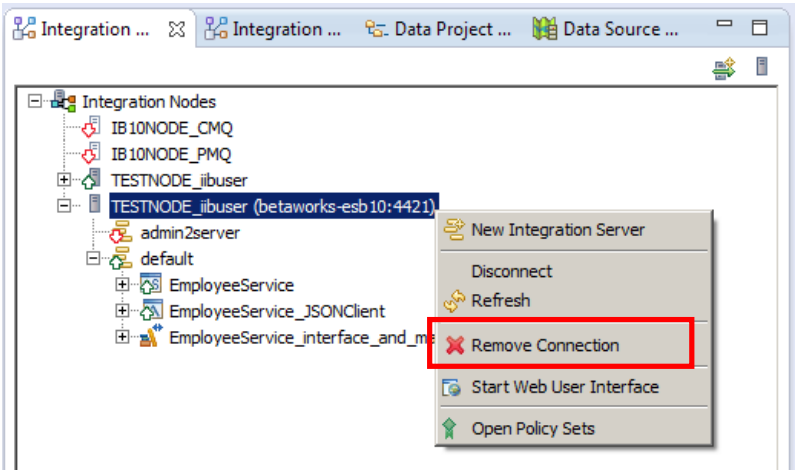


8. Expand **default** under the 'remote' connection and you will see all the deployed resources.

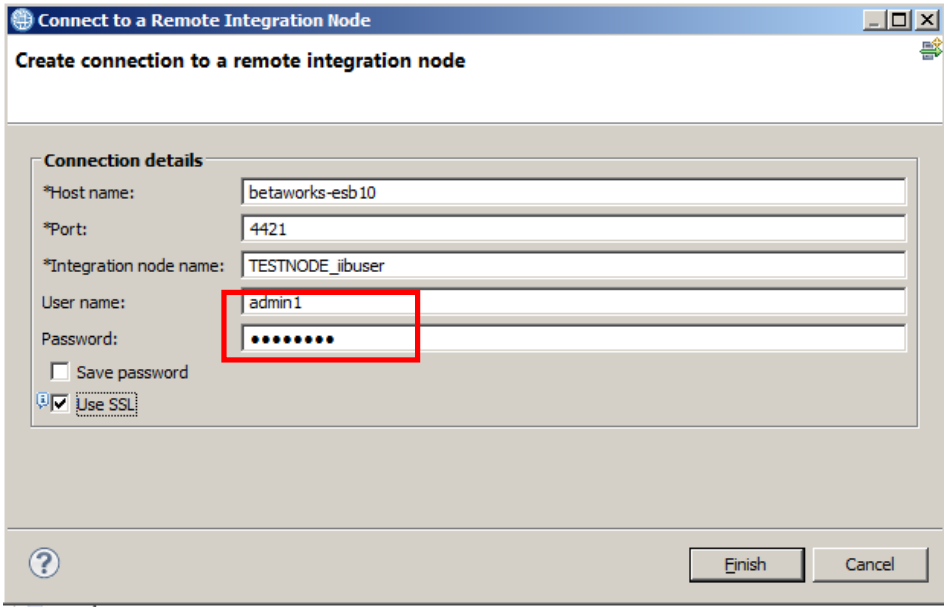


Note that this user, dev1, can see the admin2server through the local connection, but not through the remote connection. This is because that users connecting to a local node (ie. on the same system as the Toolkit) are deemed to have unrestricted access. Connecting remotely will use the access controls appropriate for the user.

9. Click on the newly created connection and then ‘Remove Connection’.



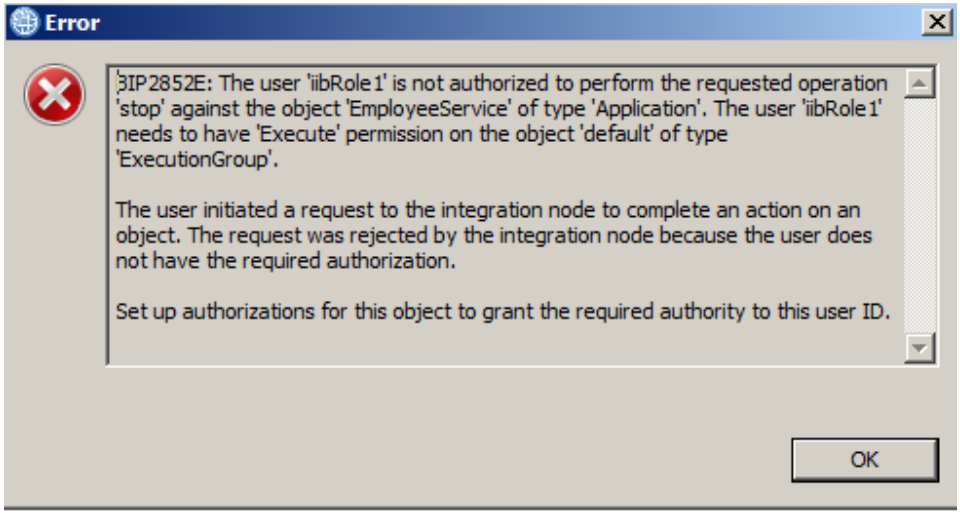
10. Once the previous step has been completed, create a new connection repeating step 6.1 and 6.2, only this time user will be admin1 (password=admin1). Click ‘Finish’.



As above, the admin1 user will be able to view all deployed artefacts in the default server.

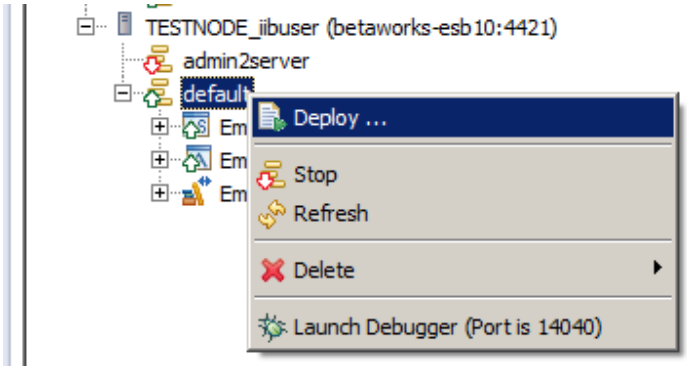
11. Right-click the EmployeeService service, and select Stop. The access controls for admin1 (iibRole1) do not permit this action, and an error window will be seen.

    Click OK.

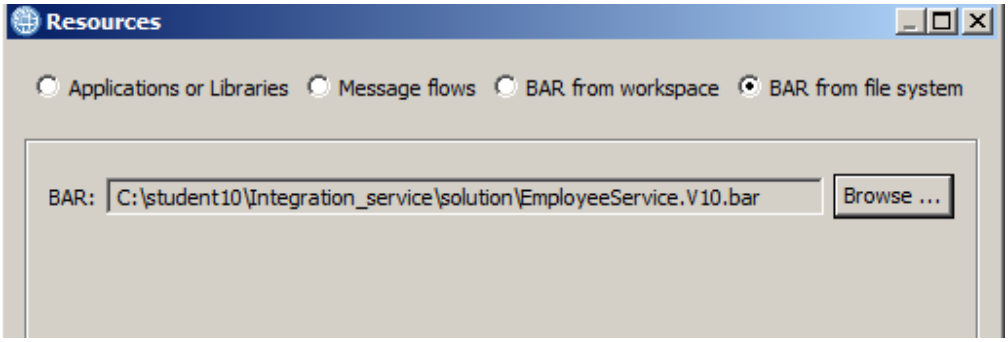


12. Now attempt to deploy a new resource to the default server.

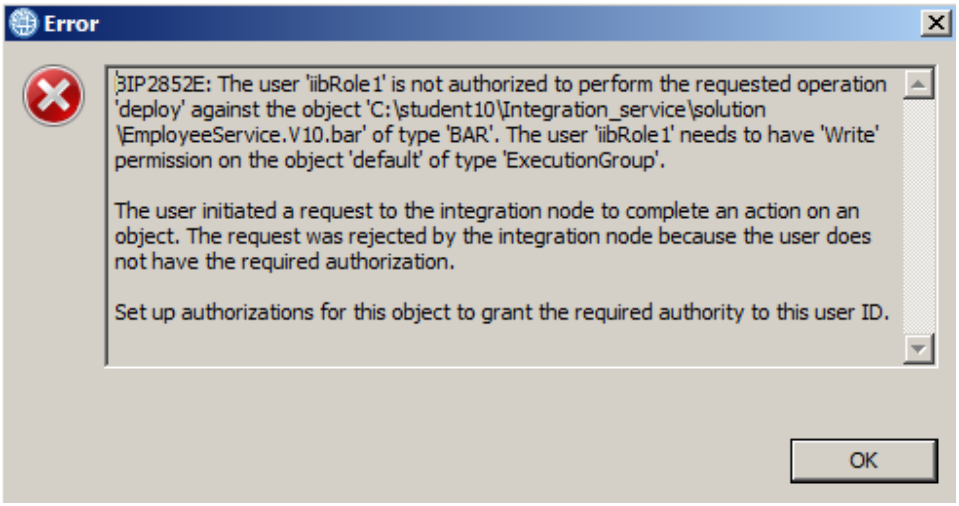
    Right-click the default server and select Deploy.



13. Select "BAR from file system", and use the Browse button to navigate to c:\student10\integration_service\solution\EmployeeService.V10.bar.



14. Click OK to deploy the bar file. A security failure window will describe why this user is not authorized to perform this action.



As a final step stop the integration node and turn off the administration security:

```
mqsistop TESTNODE_iibuser
```

```
mqsichangebroker TESTNODE_iibuser -s inactive
```

This concludes the Web Admin lab. Web users with the required authorization can administer services, applications and other resources on the Integration Node.

END OF LAB GUIDE